Curso 2011/12
CIENCIAS Y TECNOLOGÍAS/45
I.S.B.N.: 978-84-15910-49-7

CÁNDIDO CABALLERO GIL

Soluciones para la autenticación y gestión de subredes en MANETs y VANETs

Directora
PINO CABALLERO GIL

SERVICIO DE PUBLICACIONES
UNIVERSIDAD DE LA LAGUNA

SOPORTES AUDIOVISUALES E INFORMÁTICOS
Serie Tesis Doctorales

Esta tesis doctoral se la dedico a mi madre J. Teresa,
a mi esposa Jezabel, a mis hermanas Pino y Lidia
y a la memoria de mi padre Cándido y de mi hermana Gloria.

Agradecimientos

Esta tesis no habría sido posible sin la valiosa orientación y ayuda de varias personas que, de una manera u otra, han contribuido a su realización.

En primer lugar me gustaría dar las gracias a mi directora, Pino Caballero, quien, con su inestimable y generoso apoyo e inacabable estímulo, me ha ayudado a superar todos los problemas que han surgido en el camino, trabajando incansablemente y guiándome desde el principio hasta lograr el objetivo propuesto.

También quiero agradecer a Candelaria Hernández, por sus sugerencias y amabilidad en todo momento, lo que ha facilitado el desarrollo de este trabajo.

Alexis Quesada merece igualmente mi agradecimiento por haberme ayudado en los inicios de mi investigación.

Además, doy las gracias a Wladimir Bodrow y Otokar Grosek por la gran oportunidad de estar con ellos en Berlín y Bratislava, donde su hospitalidad me permitió encontrar un ambiente propicio y agradable de trabajo.

No me quiero olvidar de agradecer a Miguel Soriano, quien durante su visita me dio muy buenos consejos y recomendaciones que han aportado más valor a esta tesis.

Un agradecimiento especial va para todos mis amigos del departamento, en especial para Pepe Moreno, Belén Melián y Marcos Moreno, por compartir buenos momentos.

También agradezco a mi madre, familia y amigos, porque han estado ahí para apoyarme y subirme el ánimo cuando lo necesitaba. En particular, gracias a mi hermana Lidia por revisar el inglés de algunas partes de esta tesis.

Por último, pero no por ello menos importante, estoy agradecido a mi esposa y compañera de viaje, Jezabel Molina, por todo.

Sin ustedes, nunca habría llegado a donde estoy ahora mismo.

Prólogo

En los últimos años las redes inalámbricas están ganando cada vez más popularidad conforme sus prestaciones aumentan y se descubren nuevas aplicaciones. Dichas redes permiten a sus usuarios acceder a información y recursos en tiempo real sin necesidad de estar físicamente conectados. Además, ofrecen una gran flexibilidad a un bajo coste ya que en general no hay necesidad de usar instalaciones cableadas, lo que implica que sean fácilmente desplegables. Es por eso que resultan muy útiles en entornos donde es muy costoso instalar infraestructuras fijas, como son entornos militares, agrícolas, situaciones de emergencia, etc.

Las redes móviles ad-hoc o MANETs (Mobile Ad-hoc NETworks) son un tipo de red inalámbrica, distribuida y sin autoridad central en la que los nodos son móviles. El comportamiento de una MANET es en muchos aspectos similar al de una red Peer-TO-Peer (P2P) pues en ambos casos los nodos de la red reciben y envían información de forma descentralizada. La gestión de las MANETs conlleva muchas dificultades ya que por ejemplo su topología cambia constantemente debido a la movilidad de los nodos y a la inexistencia de una infraestructura fija.

Las redes ad-hoc vehiculares o VANETs (Vehicular Ad-hoc NETworks) pueden considerarse un subconjunto de las MANETs en las que los nodos móviles son vehículos. En su definición clásica, las VANETs permiten comunicar información no solo entre las unidades a bordo u OBUs (On Board Units) situadas en los vehículos, sino también con la infraestructura de la carretera o RSU (Road Side Unit). El objetivo principal de estos sistemas es proporcionar un mejor conocimiento de las condiciones de las carreteras a los conductores para reducir el número de accidentes y lograr que la conducción sea más cómoda y fluida, reduciendo con ello la cantidad de CO_2 que los vehículos expulsan a la atmósfera.

Las redes ad-hoc son especialmente vulnerables a varios tipos de ataques, tanto activos como pasivos. Por ejemplo, un atacante puede intentar emular a un nodo legítimo y capturar paquetes de datos y de control, destruir tablas de encaminamiento, etc. En particular, los efectos de los ataques a las VANETs pueden ser muy destructivos, ya que pueden llegar incluso a causar muertes. Por este motivo, el propósito fundamental de la presente Tesis es la propuesta de nuevas herramientas que permitan proteger las redes móviles ad-hoc contra diferentes ataques, asegurando en la medida de lo posible que la generación de información, así como su retransmisión se realizan correctamente. Para ello,

se proponen y analizan aquí nuevos esquemas de autenticación y gestión de subredes en MANETs y VANETs.

Hay que destacar que las simulaciones juegan un papel fundamental en este trabajo ya que permiten analizar y evaluar el comportamiento de las propuestas realizadas a gran escala y en diversas condiciones. En particular, gran parte de los algoritmos diseñados en esta Tesis han sido simulados con el simulador de redes NS-2 y el simulador de tráfico SUMO. También son de gran interés en esta Tesis las implementaciones de algunas de las propuestas en dispositivos reales, ya que no sólo permiten evaluar su comportamiento en entornos reales, sino también descubrir problemas que las simulaciones no detectan, y obtener datos reales para alimentar simulaciones a gran escala. Las implementaciones en dispositivos reales se han llevado a cabo en particular en la plataforma Windows Mobile usando Visual Studio 2008.

Como resultado práctico de este trabajo, y en colaboración con otras investigaciones, surge VAiPho (VANET in Phones), que es una herramienta para la asistencia a la conducción. VAiPho permite crear una red vehicular real utilizando únicamente teléfonos móviles inteligentes, sin necesidad de instalar ningún tipo de infraestructura ni en los vehículos ni en la carretera. VAiPho cuenta ya con varias aplicaciones en entornos urbanos, tales como la detección de atascos, plazas de aparcamiento libres y vehículo aparcado. Dicha herramienta es el producto de la implementación de una patente presentada por la Universidad de la Laguna como resultado de la investigación teórica descrita en esta Tesis.

Esta memoria se divide en cuatro capítulos. En el primero principalmente se presentan los conceptos previos que se utilizarán en el resto del documento. En el segundo capítulo dedicado a las MANETs se describen nuevos esquemas de gestión del ciclo de vida, de claves públicas, y de la topología de este tipo de redes. El tercer capítulo, orientado a las VANETs, recoge varias propuestas de autenticación y gestión de clústers. Finalmente, el cuarto capítulo presenta algunas cuestiones del diseño, implementación y resultados de la herramienta VAiPho, así como varias soluciones propuestas a un problema de fusión de subredes detectado tras la implementación de VAiPho.

La presente investigación sigue en curso y el próximo objetivo es desarrollar una implementación completa y funcional de VAiPho para las principales plataformas móviles iOS y Android con el objetivo de analizar su comportamiento a gran escala en entornos reales.

Contenido

Capítulo 1

Introducción

En este primer capítulo se describe brevemente el ámbito de trabajo en el que se desarrolla esta tesis doctoral, así como su motivación y los objetivos que se pretenden conseguir con ella.

Incluye también una descripción de la estructura de la memoria para facilitar su comprensión.

1.1. Acerca de la Tesis

Esta tesis doctoral ha sido desarrollada en el Departamento de Estadística, Investigación Operativa y Computación (DEIOC) de la Universidad de la Laguna (ULL). Los desarrollos a los que esta Tesis ha contribuido se enmarcan dentro de los siguientes proyectos y becas de investigación:

- Proyecto MUOVE: Mejora de la segUridad vial mediante la planificación, diseño e integración de servicios criptOgráficos en VanEts (TIN2008-02236/TSI), financiado por el Ministerio de Ciencia e Innovación.
- Beca FPI (BES-2009-016774), asociada a dicho proyecto y financiada por el Ministerio de Ciencia e Innovación.
- Proyecto titulado Diseño de un Esquema Global de Seguridad para Redes Móviles Ad-Hoc (PI2007/005), financiado por la Agencia Canaria de Investigación, Innovación y

Sociedad de la Información del Gobierno de Canarias.

1.2. Motivación y Objetivos

Esta tesis doctoral viene motivada por el propósito de encontrar soluciones, o al menos, dar nuevas ideas para el diseño de esquemas robustos de autenticación, gestión de claves y subredes en redes inalámbricas, teniendo en cuenta que estas redes se encuentran en expansión por constituir un mecanismo de comunicación flexible que permite su ampliación sin coste, y que aún no se dispone de mecanismos suficientemente rápidos y seguros para la comunicación entre los distintos dispositivos que la forman.

En particular, son objeto específico de estudio de este trabajo las redes inalámbricas de tipo MANET y VANET, para las que hemos detectado un hueco en la investigación realizada hasta el momento, que se pretende cubrir con esta Tesis.

Además, una necesidad adicional descubierta durante la realización de esta Tesis es el desarrollo de soluciones a los problemas mencionados pero sin utilizar grandes recursos de forma que se garantice el nivel de seguridad más adecuado para cada red de comunicaciones. La mejor forma de conseguir este proposito es crear esquemas auto-organizados ya que se evitaría la necesidad de una autoridad centralizada que controlase la red, y también la necesidad de instalar infraestructuras. No obstante, estos esquemas auto-organizados son mucho más vulnerables y requieren especial atención al definir los esquemas de seguridad necesarios.

Por estas razones, y a grandes rasgos, los objetivos principales que se pretenden alcanzar en este trabajo son:

- Diseño de nuevos mecanismos y protocolos de seguridad para una arquitectura auto-organizada de comunicaciones en MANETs. Las principales tareas realizadas en MANETs pueden subdividirse en cinco grandes grupos:
  - Gestión del Ciclo de Vida: Diseño de los estados y procesos que deben seguir cada nodo para formar parte de una red auto-organizada de forma que garantizen la seguridad de las comunicaciones y excluyan a los nodos maliciosos que intentan atacar la red.

  - Autenticación de Nodos: Desarrollo de un protocolo de autenticación ligero, energéticamente eficiente y capaz de ejecutarse sobre dispositivos de baja capacidad de cómputo, y sin autoridad centralizada.
  - Gestión de Claves Públicas: Desarrollo de un protocolo para intercambiar claves entre los nodos de la red de forma segura y distribuida, teniendo en cuenta la limitación de espacio de los dispositivos.
  - Gestión de Topología Mediante RFID: Se desarrolla una solución muy ligera y centralizada utilizando la tecnología RFID para autenticar etiquetas mediante lectores conectados a un servidor.
  - Simulación y Análisis: Los diferentes esquemas propuestos son objeto de análisis a través de numerosas simulaciones realizadas con el simulador de redes NS-2 (Network Simulator).

- Diseño de nuevos mecanismos y protocolos de seguridad para una arquitectura auto-organizada de comunicaciones en VANETs, que pueden subdividirse en tres grandes grupos:

  - Autenticación de Nodos: Se describen los diferentes estados en los que puede estar un nodo y se proponen clústers para organizarlos y evitar la saturación de las comunicaciones. Esta organización es especialmente importante en VANETs puesto que la topología de las carreteras hace que los vehículos tiendan a agruparse cuando se producen atascos.
  - Clústers Auto-Organizados: Desarrollo de una arquitectura distribuida y auto-organizada basada en la agrupación de nodos para reducir la sobrecarga de paquetes de control en condiciones de tráfico denso.
  - Simulación y Análisis: Los esquemas propuestos son evaluados usando los datos obtenidos de las simulaciones realizadas con el simulador de tráfico SUMO y el simulador de redes NS-2.

- Implementación de algunos de los protocolos propuestos en teléfonos móviles, a través de una herramienta diseñada llamada VAiPho (VANET in Phones). A raíz de dicha

implementación surgen numerosos problemas objeto de estudio de este trabajo, entre los que destacan:

- Interfaz de Usuario: Diseño que evite la distracción del conductor siendo totalmente automatizada, con una funcionalidad muy fácil y avisos visuales y por voz.
- Combinación de Subredes: La implementación de redes inalámbricas usando dispositivos reales mediante el protocolo IEEE 802.11n genera un problema de creación de subredes en distintos canales. En este trabajo se propone e implementa una solución a este problema.

1.3. Estructura de la Tesis

Al igual que las nuevas tecnologías, esta Tesis ha evolucionado adaptándose a las nuevas tendencias en redes inalámbricas móviles. La investigación inicialmente estaba planteada basada únicamente con el objetivo de buscar mejoras y soluciones de seguridad para MANETs. Sin embargo, una vez diseñados e implementados distintos esquemas de seguridad para MANETs, y en el marco del proyecto de investigación MUOVE y la beca asociada a él, se avanzó hacia el siguiente paso en la investigación, centrando entonces los estudios y desarrollos principalmente en las VANETs. Dado que las VANETs tienen una serie de características que las distinguen, los esquemas diseñados para MANETs no son siempre válidos para VANETs y en muchos casos deben modificarse o incluso rehacerse completamente. Finalmente, tras el diseño y simulación de los esquemas propuestos en VANETs, el último paso de esta investigación ha sido su puesta en marcha en dispositivos reales. Para ello se implementaron los diferentes esquemas en una plataforma móvil, y al conjunto de todos los esquemas se le dio el nombre de VAiPho (abreviatura de VANET in Phones).

Resultado de la evolución descrita de la investigación es la estructura de esta memoria, que se organiza como se indica a continuación.

- En el Capítulo 2 dedicado a las MANETs, tras las definiciones básicas necesarias

de las características específicas de las MANETs, se describe cómo se ha utilizado la herramienta de simulación NS-2, ofreciendo como muestra un ejemplo sencillo de simulación básica de red inalámbrica sin ningún tipo de seguridad. A continuación, se propone un nuevo protocolo denominado SLCM (Self-organizing Life Cycle Management) de gestión auto-organizada del ciclo de vida de una MANET. Después se proporciona una definición completa de un nuevo protocolo propuesto para la autenticación de nodos en MANETs y se muestra su simulación usando la herramienta NS-2. A continuación se especifica un nuevo esquema de gestión de claves públicas basado en grafos certificados y dos novedosos algoritmos de actualización de repositorios. Todo esto acompañado con la simulación realizada en NS-2 tanto del esquema propuesto como de su antecesor, permite proporcionar datos comparativos concretos. Se termina el capítulo describiendo una propuesta de gestión de la topología mediante RFID basada en que cada cada nodo lleve adherida una etiqueta para localizarlo, en la que la principal aportación es un esquema ligero de autenticación segura diseñado para funcionar con las restricciones de las etiquetas pasivas.

- En el Capítulo 3, dedicado a las *VANETs*, se describe un esquema de autenticación de nodos distribuido, especialmente diseñado para el funcionamiento en VANETs sin la necesidad de soportes externos tales como autoridad centralizada o unidades de carretera. En esta misma sección se describe cómo se generan los certificados entre nodos y cómo se actualizan los repositorios locales para que cada nodo guarde la información justa y necesaria para poder comunicarse con cualquier otro nodo de la red. Además, en este capítulo también se describe el ciclo de vida completo de un esquema de agrupación de nodos también conocido como clúster, que tiene el objetivo de reducir el número de comunicaciones realizadas en situaciones de conglomeraciones de vehículos, lo que produciría gran cantidad de paquetes en un área muy pequeña degradando la calidad de las comunicaciones. De todos estos esquemas para VANETs se realizan simulaciones mediante NS-2 y el simulador de tráfico, SUMO.

- En el Capítulo 4 se presenta la herramienta *VAiPho*, que surge de la implementación de algunos de los esquemas diseñados en esta Tesis. Las principales aportaciones

implementadas en VAiPho son el esquema distribuido de autenticación de nodos y gestión de claves, la formación del grafo certificado, y el diseño de la estructura y de las interfaces de usuario. Por otro lado, una vez diseñada e implementada la primera versión de VAiPho, nuevos problemas detectados, como es el caso de la existencia de subredes que deben fusionarse, requieren soluciones por lo que se proponen dos, una determinística y otra que intenta mejorar la primera usando lógica difusa. Las implementaciones llevadas a cabo en la versión beta a modo de prueba de VAiPho han sido desarrolladas para la plataforma Windows Mobile.

- En el Capítulo 5 se destacan algunas conclusiones y trabajos futuros.

Al comienzo de cada uno de los Capítulos 2 y 3 se incluye una sección de Estado del Arte, en la que se proporciona un breve repaso por algunas de las referencias bibliográficas más importantes de autenticación, gestión de claves y subredes en MANETs y VANETs. Asimismo el Capítulo 4 comienza con un recorrido por algunas de las soluciones alternativas existentes para los problemas de la detección de atascos y de aparcamientos.

1.4. Conceptos Básicos Generales

1.4.1. Redes Inalámbricas

El término red inalámbrica se utiliza para designar la conexión de dispositivos mediante ondas de radio o electromagnéticas en lugar de enlaces físicos (cables). La transmisión y la recepción de la información se realiza a través de puertos en los dispositivos, y su implementación se suele llevar a cabo en la capa física de red. Una de sus principales ventajas es que evita los costos de instalación de cableado, ya que son innecesarios tanto el cableado para redes de área local como el resto de conexiones físicas entre los nodos. Por ello en la actualidad las redes inalámbricas son una de las tecnologías más prometedoras. Sin embargo, también tienen una desventaja importante y es que para este tipo de redes se requiere una seguridad mucho más exigente y robusta para evitar a los intrusos ya que las transmisiones son fácilmente interceptadas. En particular, autenticación, confidencialidad, integridad, no-repudio y control de accesos son algunos de los aspectos de seguridad que

tienen que ser considerados al diseñar una red inalámbrica. Entre estos servicios, la autenticación de nodos, que garantiza la verdadera identidad de los nodos, es sin duda la más importante porque el resto de características de seguridad dependen completamente de la correcta autenticación de los dispositivos. Además de este servicio, la gestión de las claves puede considerarse otro prerrequisito indispensable para garantizar la interacción segura en cualquier red de comunicaciones, y en particular, en las redes inalámbricas.

Una red móvil ad-hoc, también conocida como MANET (Mobile Ad-hoc NETwork), consiste en un conjunto de dispositivos que forman espontáneamente una red inalámbrica y pueden moverse libremente. En dichas redes autónomas de nodos móviles se dan una serie de condiciones tales como la desprotección física del medio de difusión, los cambios frecuentes de ruta causados por la movilidad de los nodos, la carencia de una jerarquía estructurada, etc., que dificultan la protección de su seguridad. En particular, una característica que merece ser destacada por estar presente en la mayoría de MANETs es que no existe ningún tipo de infraestructura central. Otra cualidad de las MANETs que las hace más difíciles de tratar en cuestiones de seguridad es que, en la mayoría de casos, los nodos son dispositivos con poca capacidad de cómputo y una batería que se agota rápidamente, razón por la cual cualquier protocolo de seguridad adecuado para estas redes debería ser poco costoso computacionalmente, y no incrementar demasiado el tráfico de la red.

El conjunto de aplicaciones de las MANETs es muy diverso, extendiéndose desde pequeñas redes bastante estáticas con recursos energéticos limitados hasta redes a gran escala, y altamente dinámicas. Estas redes están teniendo un auge cada vez mayor en las comunicaciones móviles, lo que se ha visto reflejado en un número creciente de publicaciones en este campo durante los últimos años.

Una red ad-hoc vehicular o VANET (Vehicular Ad-Hoc NETwork), se puede ver como una forma de MANET cuyos nodos móviles son vehículos. En su definición tradicional cada unidad a bordo u OBU (On Board Unit) de cada vehículo que participa es a la vez nodo origen y/o destino de la información y router capaz de enviar información a sus vehículos vecinos y a los sistemas implantados en las carreteras o RSUs (Road Side Units) (ver Fig. 1.1). Al ser una extensión de la MANET, se parece mucho a ella pero a la vez tiene importantes características que la diferencia.

Figura 1.1: Representación de una VANET

Los nodos de una VANET tienden a moverse de forma organizada y no aleatoriamente, como se presupone en una MANET, debido a que son vehículos que siguen el trazado marcado por las carreteras. Para que los vehículos puedan formar parte de la VANET según su definición clásica, deben tener hardware añadido para generar y gestionar las comunicaciones. Además, estas redes tal como han sido tradicionalmente definidas se caracterizan por tener, en su mayoría, equipamiento adicional situado en las carreteras para comunicarse con los vehículos e informarles de los eventos que ocurren, lo que implica también tener que adecuar las infraestructuras de las carreteras.

1.4.2. Conceptos Criptográficos

Para la transmisión de información entre los nodos de la red inalámbrica, se pueden usar técnicas de protección de su confidencialidad conocidas como cifrados. El cifrado constituye una modificación del mensaje original denominado texto claro en un mensaje inteligible, denominado texto cifrado o criptograma. La aplicación concreta del algoritmo de cifrado se basa en la existencia de una clave de cifrado que define la modificación de cada mensaje original en un criptograma.

El descifrado es el proceso inverso que permite recuperar el texto claro a partir del

criptograma y la clave de descifrado. Al conjunto de algoritmos de cifrado y descifrado se le denomina criptosistema.

Existen dos grandes grupos de cifrados, los algoritmos que usan una única clave tanto en el proceso de cifrado como en el de descifrado, (ver Fig. 1.2) y los que emplean una clave para cifrar y una clave distinta para descifrar (ver Fig. 1.3). Los primeros se denominan cifrados simétricos o de clave secreta, mientras que los segundos se denominan cifrados asimétricos o de clave pública.

La llamada criptografía simétrica o de clave secreta es la ciencia que estudia y utiliza los cifrados simétricos o de clave secreta. Obsérvese que en estos esquemas las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar, por ejemplo mediante su transmisión por un canal seguro. Una vez ambas partes tienen acceso a la clave secreta, el remitente cifra el mensaje original usándola, y envía el mensaje cifrado al destinatario, que éste descifra usando la misma clave. Un buen sistema de cifrado de clave secreta basa toda su seguridad en el secreto de la clave compartida ya que los algoritmos de cifrado y descifrado normalmente son públicos. En otras palabras, no debe ser de ninguna ayuda para un atacante conocer los algoritmos que se están usando, y sólo si el atacante obtuviera la clave, le serviría de algo conocer dichos algoritmos. Entre los cifrados simétricos más conocidos destacan el cifrado en flujo RC4 y los cifrados en bloque Rijndael y Triple DES, también llamados AES y TDES respectivamente.

Por otra parte, la llamada criptografía asimétrica es la ciencia que estudia los cifrados asimétricos o de clave pública. Obsérvese que en dichos cifrados las dos claves utilizadas en las operaciones de cifrado E_B, y descifrado D_B pertenecen al mismo usuario B (Bob) al que la usuaria A (Alice) ha enviado el mensaje original M. La clave usada para cifrar es pública y se puede entregar a cualquier usuario, mientras que la otra clave, usada para descifrar, es privada y su propietario debe guardarla de modo que nadie más tenga acceso a ella. Si A usa la clave pública de B para cifrar el mensaje, una vez cifrado, sólo la clave privada de B sirve para descifrar este mensaje. Por tanto se logra la confidencialidad total del mensaje enviado, ya que nadie más, salvo B puede descifrarlo. Un buen ejemplo de cifrado asimétrico es el algoritmo RSA, que constituye a fecha de hoy el cifrado de clave pública más utilizado en el mundo. Es destacable también que los cifrados asimétricos son

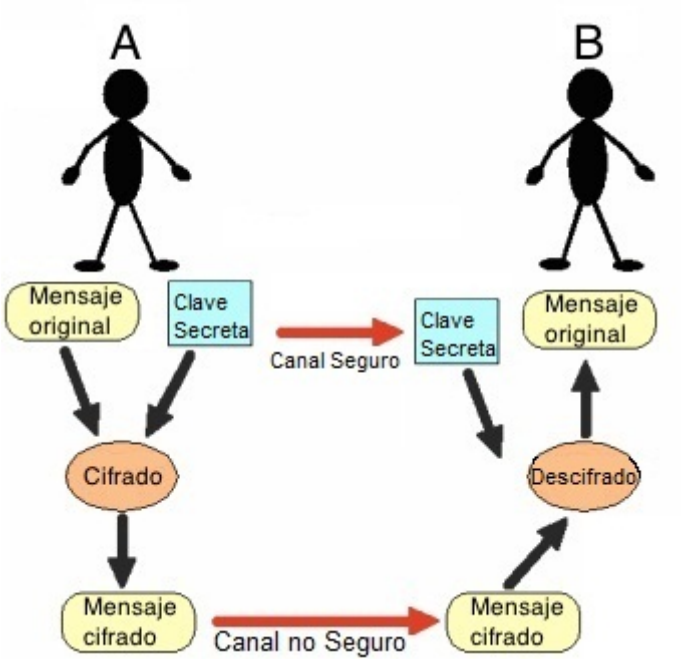


Figura 1.2: Cifrado Simétrico

el fundamento básico para la firma digital, que se menciona a continuación.

La firma digital es un esquema criptográfico que sirve para demostrar la autenticidad de un mensaje o documento electrónico así como de su emisor o autor (ver Fig. 1.4). Si la propietaria A del par de claves usa su propia clave privada para cifrar un mensaje a enviar, cualquier receptor B podría descifrarlo utilizando la clave pública de A pero en este caso lo que se consigue es la autenticación de la remitente A, ya que se sabe que sólo pudo haber sido ella quien empleó su clave privada para cifrar el mensaje. Por tanto, la firma digital proporciona al destinatario seguridad de que el mensaje fue creado por la remitente, y que no fue alterado durante la transmisión. Las firmas digitales se utilizan normalmente en numerosas aplicaciones, tales como la distribución de software, transacciones financieras y todas aquellas donde es importante detectar posibles falsificaciones y manipulaciones de información enviada o almacenada.

Una función hash o función resumen es un método para generar, resumir o identificar probabilísticamente un gran conjunto de información dando como resultado un conjunto imagen finito generalmente menor. Una de sus principales aplicaciones es producir como sali-

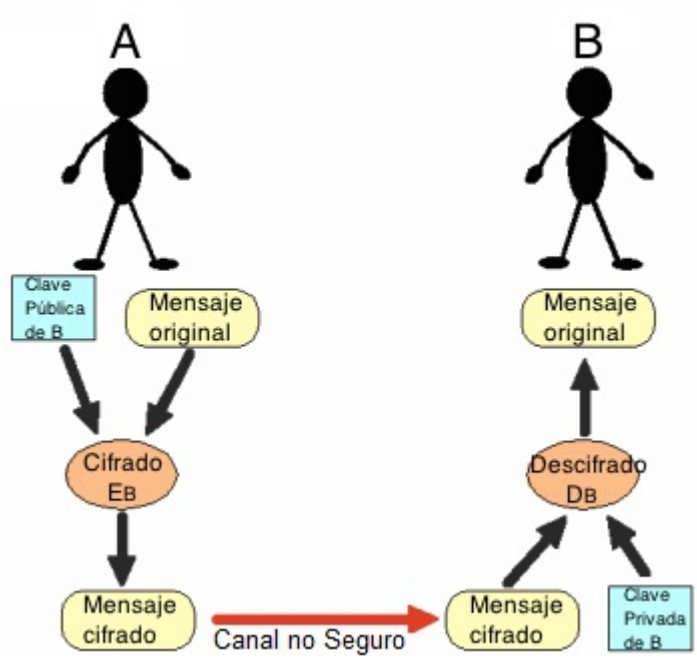


Figura 1.3: Cifrado Asimétrico

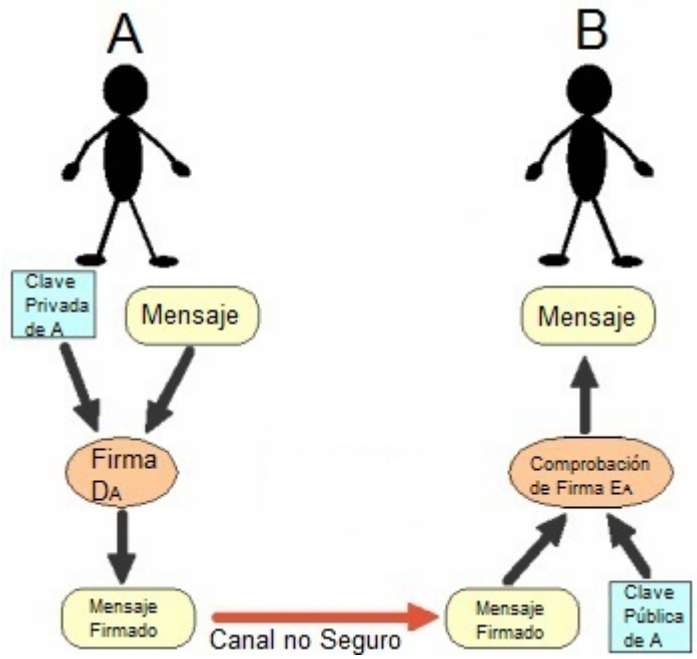


Figura 1.4: Firma Digital

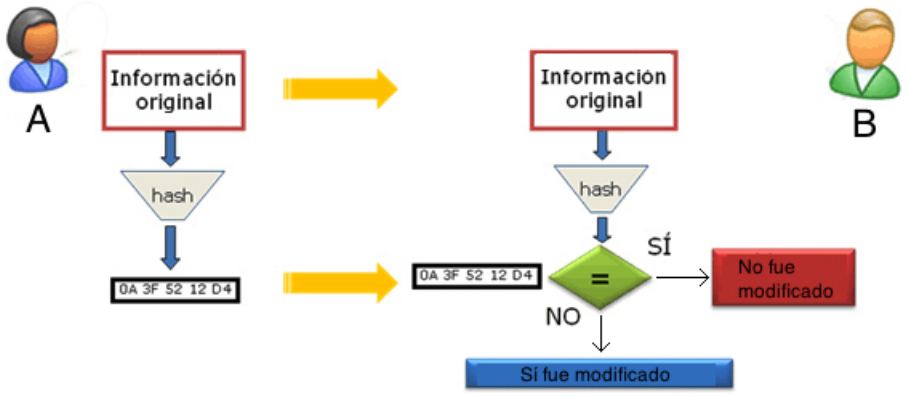


Figura 1.5: Función Hash

da elementos que representen de manera casi unívoca documentos, mensajes, etc. de entrada (ver Fig. 1.5). Sin embargo, es posible que existan salidas resultantes iguales para entradas diferentes, ya que el rango de posibles salidas es mucho menor que el de posibles entradas. Las funciones hash se usan en múltiples aplicaciones, y en el caso de la criptografía su principal uso es en firmas digitales ya que permiten producir para cualquier mensaje o documento de cualquier longitud salidas del tamaño fijo adecuado para que dicho mensaje o documento remitido sea descifrado con la clave privada de la emisora A. Una buena función hash debe producir pocas colisiones en el conjunto esperado de entrada, es decir, debe permitir que se identifiquen unívocamente las entradas con una alta probabilidad, sin provocar ambigüedad. Las propiedades que debe cumplir una buena función hash criptográfica h son:

1. Sea cual sea la longitud de la entrada M, la longitud de su hash $h(M)$ siempre debe ser la misma. Por ejemplo, si la longitud de la salida es 128 bits, si aplicamos una función hash a un mensaje M de 5 bits producirá un hash de 128 bits, y si se la aplicamos a un mensaje M de 380 millones de bits, dará un hash de 128 bits igualmente (ver Fig. 1.6).

2. Para cada entrada, la función *hash* generará una salida única.

3. Dado una entrada, debe ser fácil y rápido calcular su resumen.

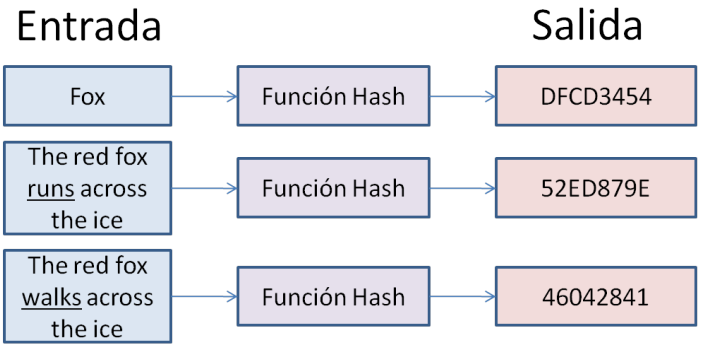


Figura 1.6: Ejemplos de Uso de una Función Hash

4. Debe ser imposible reconstruir una entrada a partir de su resumen.

5. Debe ser difícil encontrar dos entradas distintas que produzcan una colisión, es decir, que tengan el mismo resumen.

Uno de los algoritmos más conocidos de una función hash es el denominado MD5, que produce salidas de 128 bits.

La autenticación de usuarios, que permite comprobar su identidad y se usa normalmente en el control de accesos, es el aspecto más importante de la seguridad en cualquier sistema. Los esquemas de autenticación más habituales son débiles ya que se basan en la aportación de un secreto invariable llamado contraseña, que puede ser capturada y posteriormente repetida. Una solución a este problema, que se usa para definir esquemas de autenticación fuerte, es la llamada demostración de conocimiento nulo, concepto criptográfico introducido por primera vez en 1986 por Goldreich, Micali y Wigderson [95].

Una demostración de conocimiento nulo o ZKP (Zero-Knowledge Proof) se puede definir como un protocolo criptográfico bipartito que permite que una probadora A con gran potencia de cálculo convenza a un verificador B en tiempo probabilístico polinomial de que una afirmación es verdadera, sin revelar ninguna información salvo la veracidad de la afirmación.

A pesar de su peculiar definición, existe actualmente mucha investigación en ZKPs motivada principalmente por su utilidad en la descripción de sistemas de autenticación donde la probadora A desea demostrar su identidad al verificador B mediante cierta información

secreta comprobable, que generalmente es la solución a un problema difícil, pero sin que el verificador pueda aprender nada sobre dicho secreto.

Se pueden encontrar en la bibliografía varias ZKPs como parte de esquemas del control de acceso para las redes cableadas, basadas en diversos problemas tales como el problema del logaritmo discreto [70] [181], el problema de los residuos cuadráticos [88] o problemas difíciles de grafos [43] [118]. Por otra parte, existe una propuesta reciente basada en ZKPs para MANETs donde se definen dos niveles de seguridad diferentes mediante el uso de un problema NP-completo y de un problema difícil en media, [44].

Con respecto a la definición formal de ZKP, hay tres características principales que toda ZKP debe satisfacer:

1. Verificación: Si la afirmación es verdadera, el verificador quedará convencido de ello mediante una ejecución correcta del protocolo.

2. Validez: Si la afirmación es falsa, ninguna probadora deshonesta puede convencer a un verificador honesto de que sea verdad, excepto con una pequeña probabilidad.

3. Conocimiento Nulo: Si la afirmación es verdad, ningún verificador falso puede aprender nada de ello, aparte del hecho de que la afirmación es verdadera.

Por otra parte, con respecto al diseño práctico de ZKPs, las técnicas de "corte-y-elección" y de "reto-respuesta" son esquemas básicos típicamente usados. Para concretar, normalmente el diseño de una ZKP implica interacción entre A y B de forma que algunas de las posibles respuestas de A a un reto aleatorio lanzado por B demuestran su conocimiento de la solución secreta, mientras que las otras sirven de garantía contra el posible fraude de un probador deshonesto. También en general, las ZKPs son protocolos probabilísticos basados en la existencia de una probabilidad concreta de fraude. Además, las ZKPs generalmente consisten en varias iteraciones de un subprograma, de manera que repitiéndolo un número suficiente de veces, la confianza del verificador en la honradez del probador aumenta ya que la probabilidad de fraude se hace más pequeña a medida que el número de iteraciones crece.

1.4.3. Problemas de Grafos

Los esquemas de control de acceso incluidos en las propuestas de autenticación de nodos que se describen más adelante se basan en el esquema general de demostración de conocimiento nulo introducido en [43].

Dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ que tienen el mismo conjunto de vértices $V = V_1 = V_2$ se dice que son isomorfos si existe una permutación π de V tal que $(a, b) \in E_1 \leftrightarrow (\pi(a), \pi(b)) \in E_2$ (véase la Fig. 1.7). El problema computacional de determinar si dos grafos finitos son isomorfos se llama el problema de isomorfismo de grafos. Tal problema no es probable que sea NP-completo, pero es NP, debido a que no se conoce un algoritmo de tiempo polinomial que los resuelva. El isomorfismo de grafos es utilizado varias veces en esta investigación.

Un circuito hamiltoniano de un grafo es un circuito que visita cada nodo exactamente una vez y vuelve al nodo en el que comienza. La determinación de si tales ciclos existen en un grafo o no es lo que se conoce como problema del circuito hamiltoniano (HCP, Hamiltonian Cycle Problem), que es un problema NP-completo.

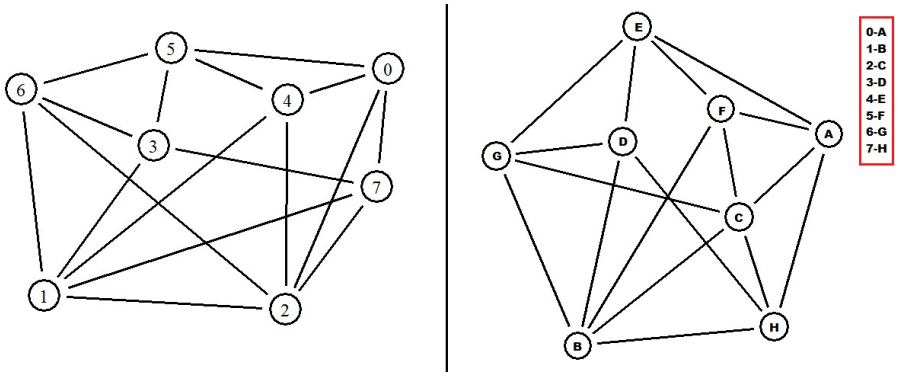


Figura 1.7: Grafo Isomorfo

Un conjunto independiente en un grafo de $G = (V, E)$ es un subconjunto de vértices $I \in V$ tal que no hay ningún par de nodos en I unidos por una arista en E. El problema del conjunto independiente máximo consiste en encontrar el mayor conjunto independiente de un grafo, mientras que el problema del conjunto independiente maximal consiste en encontrar un conjunto independiente que no esté contenido en otro conjunto independiente

mayor. El problema del conjunto independiente de máximo peso se define para grafos en los que se asocia cada nodo con un peso, y en este caso el objetivo es encontrar un conjunto independiente cuyos nodos maximizan el peso total. Este problema ha sido escogido como base para la propuesta de gestión de clústers en VANETs.

1.4.4. Seguridad en Wi-Fi

En la actualidad la mayoría de las redes inalámbricas se basan en tecnología Wi-Fi (Wireless Fidelity), que es una certificación de que los dispositivos cumplen los estándares 802.11 relacionados con las redes inalámbricas de área local. Los dispositivos habilitados con Wi-Fi, tales como: ordenadores personales, consolas de videojuegos, teléfonos inteligentes o reproductores de audio digital, pueden conectarse a Internet a través de puntos de acceso de red inalámbrica. Dichos puntos de acceso (o hotspots) tiene un alcance de unos 20 metros en interiores y al aire libre alcanzan una distancia mayor que puede cubrir grandes áreas, usando además la superposición de múltiples puntos de acceso.

Entre los protocolos de seguridad básica habituales para redes inalámbricas tipo Wi-Fi en general destacan:

- *WEP (Wired Equivalent Privacy)* es uno de los protocolos de seguridad utilizado en las redes inalámbricas que cumplen el estándar IEEE 802.11, también conocidas como redes Wi-Fi [189]. Se diseñó para garantizar la confidencialidad e integridad de los datos en un medio inseguro como es el inalámbrico. Ofrece servicios de confidencialidad a la capa de enlace, cifrando los datos que viajan por el aire y que son susceptibles de ser interceptados. WEP utiliza el algoritmo RC4 [175] para el cifrado en flujo, con claves de 40 ó 104 bits según la versión, y un vector de inicialización de 24 bits en ambos casos. Debido a defectos estructurales de diseño de WEP, hoy se considera totalmente inseguro, por lo que sólo es aceptable en redes inalámbricas donde los requerimientos de seguridad son muy bajos. En particular, se han encontrado multitud de vulnerabilidades concretas en WEP, tales como:

  - es posible acceder a la red y conseguir autorización para unirse a ella, por lo que no ofrece servicio específico de autenticación segura,
  - una vez conocida la contraseña se puede descifrar el tráfico dirigido a otros nodos, por

lo que no ofrece servicio de confidencialidad,

- es habitual reutilizar el vector de inicialización, lo que facilita los ataques para recuperar la clave,
- el chequeo de redundancia cíclica usado para proteger la integridad de los datos transmitidos es inseguro ya que es posible alterar la información y actualizar el código de chequeo del mensaje sin que el receptor detecte la modificación.

- *WPA (Wi-fi Protected Access)* es un sistema para proteger las redes inalámbricas Wi-Fi creado para corregir las deficiencias del sistema previo WEP. WPA soluciona dos problemas vitales de la seguridad de WEP [189]: resuelve faltas en la autenticación de los dispositivos frente a la red, y repara el débil cifrado de la información que viaja por la red. WPA fue diseñado para utilizar un servidor de autenticación que distribuye claves diferentes a cada usuario, aunque también se puede utilizar en un modo menos seguro con clave compartida. La información se cifra utilizando el mismo algoritmo RC4 que WEP, aunque con una clave de 128 bits y un vector de inicialización de 48 bits. Una de las mejoras sobre WEP es la implementación del llamado protocolo de integridad de clave temporal, que cambia las claves de forma dinámica. Además WPA también mejora la integridad de la información cifrada ya que incluye un nuevo código de integridad del mensaje conocido como *Michael*. De todos modos el protocolo WPA también es vulnerable ante ciertas amenazas, tal y como la que se muestra en [143].

- *WPA2* está basado en el nuevo estándar 802.11i [189]. Este protocolo apuesta por el mecanismo de cifrado en bloque AES, que implica a la vez mayor seguridad, mayor complejidad, y mayor potencia o capacidad de procesamiento para cifrar y descifrar la información, lo que hace que los dispositivos inalámbricos más antiguos sean absolutamente incompatibles con WPA2.

1.4.5. Encaminamiento en Redes Ad-hoc

Dada la ausencia de infraestructura centralizada, en las redes ad-hoc el encaminamiento requiere la colaboración de los nodos de la red ya que se realiza de forma que cada

dispositivo hace la función de router de los paquetes ajenos, es decir, el encaminamiento es multi-hop o a múltiples saltos. Por otra parte, dada la movilidad de sus nodos, en las MANETs se dan una serie de condiciones restrictivas que dificultan más aún el encaminamiento [73].

En este trabajo hemos elegido basarnos en una clasificación según el alcance de los protocolos para determinar cuál es el mejor para un escenario concreto, independientemente de su algoritmo o de su esquema de descubrimiento de rutas. En concreto los protocolos que usamos en este trabajo son:

- **DSDV**.

  Destination-Sequenced Distance-Vector (DSDV) [165] es un protocolo unicast proactivo de encaminamiento adaptado del tradicional RIP (Routing Information Protocol). Su principal objetivo es evitar los problemas de bucles en la actualización de las tablas de encaminamiento por lo cual añade un nuevo campo a las tablas RIP, el número de secuencia que permite distinguir entre una tabla antigua y una más reciente.

  Como su nombre indica, DSDV implementa un algoritmo basado en el vector de distancias. Eso significa que mantiene tablas con todos sus destinos accesibles junto con el siguiente salto, la métrica, y un número de secuencia de la entrada en la tabla generado por el nodo destino. Las tablas se mandan en modo broadcast de forma periódica o cuando ocurre un cambio significativo de la topología de red. Una ruta es considerada mejor que otra si tiene un número de secuencia mayor o, en caso de empate, si la distancia al destino es menor.

  Cuando un nodo B detecta que la ruta hacia cierto destino D se ha roto, inunda la red con una actualización de esa entrada en la que se ha incrementado el número de secuencia en uno y la distancia se marca como infinita. Cuando A recibe este mensaje incorpora a su tabla la actualización de la entrada hacia D a través de B siempre que no tuviera una entrada mejor para alcanzar D.

  Para conseguir una cierta consistencia en las tablas de encaminamiento de cada nodo al cambiar la topología de la red, las actualizaciones deben ser frecuentes y suficientemente rápidas como para que cada nodo pueda tener una visión realista de la red

en un momento dado. El problema fundamental de DSDV es la elevada sobrecarga de control que genera. Al no haber una especificación estándar, no hay productos comerciales basados en este protocolo. Sin embargo, es la base sobre la cual se han desarrollado otros protocolos como por ejemplo AODV.

- **DSR**

  Dynamic Source Routing (DSR) es un protocolo reactivo unicast. El protocolo se compone de dos mecanismos: el descubrimiento y el mantenimiento de rutas que permiten a un nodo origen descubrir y mantener las rutas hacia un nodo destino cuando se necesita mandar tráfico en la red ad-hoc. Se basa en una técnica de "Source Routing". La idea de esta técnica es determinar la mejor ruta completa hacia un destino. El nodo origen inunda la red con una trama de exploración. Al recibir una réplica de la trama exploradora, cada nodo se agrega explícitamente en la cabecera de la trama, y actualiza sus tablas con la información contenida en la cabecera de dicha trama.

  El descubrimiento de rutas es el mecanismo por el cual un nodo origen S que desea mandar tráfico a un nodo destino D, obtiene la ruta hacia D. Si S no dispone de ninguna ruta hacia D, empieza un proceso de descubrimiento de rutas mediante un broadcast del paquete Route REQuest (RREQ). Este paquete contiene la dirección de destino, la dirección del nodo fuente y un número único de identificación. Cada nodo que recibe un paquete RREQ, revisa si conoce la ruta hacia el destino. Si no la conoce, se reenvía el paquete. Si la tiene, contesta en sentido inverso con un paquete Route REPly Packet (RREP). Todos los nodos que participan en el reenvío del RREP añaden su dirección en la cabecera del paquete, creando de ese modo la ruta completa hasta el destino.

  En cambio, el mantenimiento de rutas consiste en la capacidad de detectar que una ruta almacenada en una tabla ya no se puede usar debido a un cambio de topología. El mantenimiento de rutas detecta que un enlace en la ruta hacia D ha desaparecido. Se producen paquetes de error en un nodo, cuando la capa de enlace encuentra un problema grave de transmisión. Este paquete de error contiene las direcciones de los dos nodos que están unidos por el enlace que falló. En este caso, si S conoce otra ruta

hacia D se puede usar, o bien se vuelve a invocar el mecanismo de descubrimiento de rutas para remplazar la ruta caída hacia D. El mantenimiento de rutas sólo tiene cabida cuando S está mandando tráfico a D.

DSR es un protocolo totalmente reactivo, lo que implica que no existe ningún tipo de mensaje periódico, y por tanto reduce de forma significativa el tráfico de control en la red y aprovecha más los recursos de red para paquetes útiles. Además cada vez que se lleva a cabo el descubrimiento de rutas, los nodos implicados pueden extraer y almacenar información sobre la topología de red, lo que ahorra muchos mensajes de control.

Para evitar que se produzca el problema de múltiples respuestas simultáneas y optimizar la ruta final, cuando un nodo recibe un RREQ, se introduce un pequeño retardo variable en la respuesta de cada nodo con una ruta en su caché.

- **AODV**

  Ad-hoc On-demand Distance-Vector (AODV) routing [166] es un protocolo reactivo unicast de encaminamiento. Se construye sobre el protocolo DSDV. La idea es mejorar DSDV minimizando el número de paquetes broadcast requeridos para crear rutas, ya que al ser bajo demanda, los nodos que están en el camino no tienen que participar en el intercambio de tablas ni que mantener la ruta. A pesar de ser un protocolo reactivo, AODV tiene la peculiaridad de emitir mensajes alertando sobre su presencia de forma periódica mediante una técnica llamada Link Layer Feedback. Esa técnica permite que los nodos tengan conocimiento de sus vecinos más cercanos y mantegan sus tablas actualizadas reflejando los cambios en la topología cercana. Estas tablas se mantienen actualizadas a lo largo del tiempo, eliminando las entradas innecesarias.

  Cuando un nodo S quiere transmitir tráfico a un destino D y no tiene una ruta válida hacia D comienza el mecanismo de descubrimiento. Primero se manda en modo broadcast una petición de ruta, Route REQuest (RREQ) a todos sus vecinos. Este mensaje incluye su propia dirección, la del nodo destino D, y el último número de secuencia recibido de D, en el caso de que se hubiera recibido algún dato con anterioridad. Este mensaje inunda la red y los nodos que atreviesa guardan una ruta inversa

hacia S, lo que implica que AODV sólo soporta enlaces bidireccionales. Cuando llega a un nodo que dispone de la ruta hacia D, se comprueba el número de secuencia para el destino D. Si éste es mayor que el incluido en el mensaje, se ha encontrado una ruta válida hacia D. El nodo que dispone de la entrada hacia D manda un mensaje de respuesta o Route REPly (RREP) de vuelta hacia S siguiendo la ruta creada durante la inundación. En este mensaje se incluye el último número de secuencia recibido por el emisor del mensaje RREP. Los nodos que reciben el RREP guardan una entrada hacia D que apunta al nodo que les ha transmitido el mensaje, por lo cual sólo se guarda en la tabla el siguiente salto y no la ruta entera. Si pasado un cierto tiempo y que no se ha recibido ningún RREP, S considera que no hay ruta válida hacia D en ese momento. Las tablas se mantienen actualizadas mientras esté en uso el enlace. Si un nodo origen se mueve, él mismo reinicia el proceso de descubrimiento de rutas. Si un nodo intermediario se mueve, su vecino anterior (en el sentido directo origen-destino) propaga hasta S, un RREP no solicitado con un número de secuencia mayor y con valor de saltos al destino infinitos. De esa manera, S sabe que si quiere seguir usando el enlace tiene que reiniciar el proceso de descubrimiento de rutas.

- **OLSR**

  Optimized Link State Routing (OLSR) [62] es un protocolo proactivo de encaminamiento basado en el estado de enlace. OLSR es una optimización directa del algoritmo de estados de enlace adaptado a los requisitos específicos de una red inalámbrica de área local con alta movilidad. La optimización consiste principalmente en la reducción del tamaño de las tablas de enlaces intercambiadas así como del número de retransmisiones necesarias durante los periodos de inundación. La clave del algoritmo reside en el uso de una técnica de retransmisiones multipunto llamada MPR (Multi Point Relay).

  Según dicha técnica los nodos intercambian periódicamente mensajes HELLO con sus vecinos que permiten detectar la presencia de un nodo vecino así como recoger información relativa al estado del enlace con ese vecino. En los mensajes HELLO se puede incluir información indicando que el nodo es un nodo MPR. Usando esa

información cada nodo elige dentro de su conjunto de vecinos un subconjunto que declara subconjunto MPR. Así, cada nodo tiene conocimiento de un subconjunto de nodos MPR que le permite tener conectividad con todos los nodos distantes uno o dos saltos. De este modo, sólo los nodos MPR se encargarán de retransmitir los mensajes broadcast. Para descubrir la topología de la red, los nodos intercambian información acerca del estado de enlace que los conectan con los nodos MPR. Los intercambios son periódicos o generados por eventos relativos a ruptura de enlace. Incluir en las tablas sólo los enlaces a los nodos MPR reduce el tamaño de las mismas, lo que permite reducir el ancho de banda consumido durante su intercambio. Al mismo tiempo permite que las rutas que se vayan creando a posteriori sean óptimas en cuanto a números de saltos ya que sólo usan nodos MPR. OLSR se adapta bien a redes con gran número de nodos y alta movilidad.

1.4.6. Estándar IEEE 802.11p (WAVE)

Un grupo de investigación de IEEE ha diseñado un nuevo estándar para comunicaciones en VANETs denominado WAVE (Wireless Access in Vehicular Environments), el cual es referenciado también como IEEE 802.11p. WAVE es una evolución del estándar IEEE 802.11 (base de las redes Wi-Fi), con modificaciones a nivel físico y de MAC para mejorar su comportamiento en el entorno vehicular y servir de soporte a los sistemas inteligentes de transporte o ITS (Intelligent Transportation Systems). Su característica más importante es la utilización de un esquema de modulación que alcanza distancias hasta de un kilómetro y soporta velocidades de los nodos relativamente altas (200 kmph). Su frecuencia de operación es 5.9 Ghz, con seis canales de servicio y un canal de control.

Existe un estándar de nivel superior, llamado IEEE 1609, sobre el que IEEE 802.11p se basa. WAVE a su vez es la base del proyecto DSRC (Dedicated Short Range Communications) basado en la arquitectura CALM (Communications, Air-interface, Long and Medium range) para VANETs, especialmente para transacciones comerciales y seguridad vial.

Se esperaba que el estándar IEEE 802.11p entrase a funcionar en el mercado en el año 2011 pero la crisis mundial trastocó estos planes posponiéndolos sin fecha definida, de

manera que a fecha de hoy no existe ningún dispositivo comercializado que cumpla dicho estándar.

1.5. Contribuciones de la Tesis

Esta memoria incluye los resultados de diversos trabajos de investigación que han sido objeto de publicación en revistas y congresos relacionados con la seguridad y la gestión de MANETs y VANETs. En la mayor parte de los casos, las implementaciones tanto en simulaciones con NS-2 y/o con SUMO, como en dispositivos reales han jugado un papel fundamental.

Como primera contribución recogida en esta memoria, en el Capítulo 2, dedicado a las MANETs, se propone un nuevo sistema denominado SLCM (Self-organizing Life Cycle Management) de gestión auto-organizada del ciclo de vida de una MANET. Una parte importante de dicho sistema es un nuevo protocolo propuesto para la autenticación de nodos en MANETs. Ambos esquemas forman parte de un artículo aceptado para su publicación en una revista de impacto [17], y de otro recogido en un volumen LNCS [33]. Además, sucesivas versiones preliminares de dichas propuestas fueron expuestas en dos congresos indexados [38] [39] y en dos congresos no indexados [21] [26].

El Capítulo 2 también recoge un nuevo esquema de gestión de claves públicas en MANETs basado en grafos certificados, y dos novedosos algoritmos de actualización de repositorios. Dichos algoritmos están incluidos en dos publicaciones LNCS [40] [103].

Se culmina el Capítulo 2 describiendo una propuesta de gestión de la topología de las MANETs mediante RFID basada en que cada nodo lleve adherida una etiqueta para localizarlo, en la que la principal aportación es un esquema ligero de autenticación mutua entre lector y etiqueta diseñado para funcionar con las restricciones de las etiquetas pasivas. Esta propuesta ha sido aceptada para su publicación en una revista de impacto [34]. Versiones preliminares, así como ideas relacionadas con diversos aspectos del esquema han sido objeto de dos publicaciones LNCS [28] [155], y presentadas en un congreso no indexado [36].

En el Capítulo 3, dedicado a las VANETs, se describe un esquema de autentica-

ción de nodos distribuido, especialmente diseñado para su funcionamiento sin necesidad de soportes externos tales como autoridades centralizadas o unidades de carretera. La primera contribución al respecto es la descripción de esquemas para generar certificados entre nodos y actualizar repositorios locales para que cada nodo pueda almacenar la información mínima necesaria para poder comunicarse con cualquier otro nodo de la red. Estas propuestas han sido presentadas en un volumen LNCS [47] y en un congreso indexado [37], y una versión mejorada ha sido enviada a una revista de impacto [20].

Además, en el Capítulo 3 también se describe una arquitectura para gestionar grupos de nodos en VANETs, aquí denominados clústers, que tiene como objetivo reducir el número de comunicaciones realizadas en situaciones de tráfico denso, donde se produce una gran cantidad de paquetes de información en un área limitada, lo que degrada la calidad de las comunicaciones. Diferentes componentes de la arquitectura propuesta han sido publicados en dos volúmenes LNCS [22] [149], y presentados en dos congresos indexados [23] [150], y tres congresos no indexados [24] [27] [25], habiendo recibido en uno de ellos un premio al mejor trabajo. La visión global de la arquitectura está en proceso de revisión en una revista de impacto [19].

En el Capítulo 4 se presenta la herramienta VAiPho, que surge de la implementación de algunos de los esquemas diseñados en esta Tesis con objeto de desplegar la primera VANET real usando solo teléfonos móviles. Las principales aportaciones implementadas en VAiPho son los esquemas distribuidos propuestos para autenticación de nodos y gestión de claves públicas, y el diseño de la herramienta y de las interfaces de usuario para teléfonos móviles, así como diversas aplicaciones vehiculares. Se ha llevado a cabo la implementación completa de una versión beta a modo de prueba de VAiPho para la plataforma Windows Mobile, así como versiones parciales en Android y Symbian, y han sido desarrolladas numerosas demostraciones con móviles y vehículos en entornos reales frente a distintas empresas y entidades. VAiPho puede considerarse la principal contribución de esta Tesis, ya que se ha visto reflejado en la patente [35], cuya licencia de comercialización ha sido recientemente adquirida por una empresa nacional. Diversos resultados relacionados con VAiPho han sido presentados en tres comunicaciones en dos congresos no indexados [24] [27] [31]. La versión más actual del diseño de VAiPho ha sido enviada a una revista de impacto [30]. Además

es destacable que la idea de VAiPho ha sido merecedora del primer premio del concurso de emprendedores "Conocer es Valer" de la Universidad de la Laguna [29]. La implementación de VAiPho para Symbian fue objeto de un proyecto fin de carrera dirigido que recibió la máxima calificación [139].

Por último, para cerrar la memoria y el Capítulo 4 dedicado a VAiPho, se incluyen propuestas de solución a un problema detectado una vez diseñada e implementada la primera versión de VAiPho, que es la existencia de subredes que deben fusionarse. En concreto proponemos dos soluciones: una determinística, y otra que la mejora usando lógica difusa. Los resultados de esta última sección han sido enviados a una revista de impacto, que se encuentra actualmente en fase de revisión [18].

1.5.1. Revistas Indexadas y LNCS

- [17] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-Organized Life Cycle Management of MANETs. Aceptado en Security and Communication Networks, 2012.

- [18] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Merging Subnetworks in VANETs by Using the IEEE 802.11xx Protocol. Enviado a Eurasip Journal of Wireless Communications and Networking, 2011.

- [19] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-Organized Clustering Architecture for Vehicular Ad-hoc Networks. Enviado a Journal on Cluster Computing, 2011.

- [20] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Zero-Knowledge Authentication in Self-Organized VANETs. Enviado a IETE Journal of Research, 2011.

- [22] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Group Formation Through Cooperating Nodes in VANETs. Cooperative Design, Visualization and Engineering, Lecture Notes in Computer Science, Vol. 6240, 105-108, 2010.

- [28] Caballero-Gil, C., Caballero-Gil, P., Peinado-Dominguez, A., Molina-Gil, J. Lightweight Authentication for RFID used in VANETs. Lecture Notes in Computer Scien-

ce. 12th International Conference on Computer Aided Systems Theory EUROCAST 2011, Springer-Verlag, 2011.

- [30] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P. Design and Implementation of VAiPho, Tool for Deploying VANETs with Phones. Enviado a Computers & Electrical Engineering, 2011.

- [33] Caballero-Gil, P., Caballero-Gil, C. A Global Authentication Scheme for Mobile Ad-hoc Networks. Advances in Information and Computer Security, Lecture Notes in Computer Science, Vol. 4752, pp. 105-120, 2007.

- [34] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. RFID Authentication Protocol Based on a Novel EPC Gen2 PRNG. Aceptado en Information-An International Interdisciplinary Journal, 2012.

- [40] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Quesada-Arencibia, A. A Simulation Study of New Security Schemes in Mobile Ad-hoc Networks. Computer Aided Systems Theory EUROCAST 2007, Lecture Notes in Computer Science, Vol. 4739, 73-81, 2007.

- [45] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Data Aggregation Based on Fuzzy Logic for VANETs. Computational Intelligence for Security in Information Systems, Lecture Notes in Computer Science, Vol. 6694, 33-40, 2011.

- [47] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C., Hernández-Goya, C. Security in Commercial Applications of Vehicular Ad-Hoc Networks, Financial Cryptography and Data Security, Lecture Notes in Computer Science, Vol. 6052, 427, 2010.

- [103] Hernández-Goya, C., Caballero-Gil, P., Delgado-Mohatar, O., Molina-Gil, J., Caballero-Gil, C. Using New Tools for Certificate Repositories Generation in MANETs. Data and Applications Security XXII, Lecture Notes in Computer Science, Vol. 5094, 175-189, 2008.

- [104] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Cooperation Enforcement Schemes in Vehicular Ad-hoc Networks. Computer Aided Systems

Theory EUROCAST 2009, Lecture Notes in Computer Science Vol. 5717, 429-436, 2009.

- [106] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Extending OLSR Functionalities to PKI Management. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6928, 2011.

- [144] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Avoid Non-Cooperation in Fully Self-Organized VANETs. Enviado a IEICE Transactions on Communications, 2011.

- [145] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Prevent Misbehaviour in VANETs. En segunda ronda de Journal of Universal Computer Science, 2011.

- [146] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Probabilistic Aggregation for Data Authentication in VANETs. En tercera ronda de Transportation Research Part C, 2011.

- [149] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Enhancing Collaboration in Vehicular Networks. Cooperative Design, Visualization and Engineering, Lecture Notes in Computer Science, Vol. 6240, 77-80, 2010.

- [155] Molina-Gil, J., Caballero-Gil, P., Fuster-Sabater, A., Caballero-Gil, C. Pseudorandom Generator to Strengthen Cooperation in VANETs. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6927, 2011.

1.5.2. Congresos Indexados

- [23] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Knowledge Management Using Clusters in VANETs. Description, Simulation and Analysis. International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management IC3K-KMIS. 2010.

- [37] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Fúster-Sabater, A. On privacy and Integrity in Vehicular Ad-hoc Networks. International Conference on Wireless Networks ICWN. 2010.

- [38] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Hernández-Goya, C. Self-Organized Authentication Architecture for Mobile Ad-hoc Networks. 6th International. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. Wiopt 2008.

- [148] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Cooperative Approach to Self-Managed VANETs. International Conference on Wireless Information Networks and Systems WINSYS. 2010.

- [150] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Group Proposal to Secure Vehicular Ad-hoc Networks. International Conference on Security and Management SAM. 2010.

- [151] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. A Vision of Cooperation Tools for VANETs. IEEE International Workshop on Data Security and PrivAcy in wireless Networks DSPAN-IEEE WoWMoM. 2010.

- [152] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Enhancing Cooperation in Wireless Vehicular Networks. 8th International Workshop on Security in Information Systems WOSIS. 2011.

- [156] Molina-Gil, J., Caballero-Gil, P., Hernández-Goya, C., Caballero-Gil, C. Data Aggregation for Information Authentication in VANETs. Sixth International Conference on Information Assurance and Security IAS. 2010.

1.5.3. Otros Congresos

- [21] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Solución Global para la Autenticación de Nodos en MANETs. II Simposio sobre Seguridad Informática - Congreso Español de Informática CEDI. 2007.

- [24] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Tool to Simulate Groups in Vehicular Networks Using NS-2 and Tracegraph. 5th European Conference on Circuits and Systems for Communications ECCSC. 2010.

- [25] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Using Groups to Reduce Communication Overhead in VANETs. Second International Conference on Advances in P2P Systems - AP2PS. 2010.

- [26] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organizing Life Cycle Management of Mobile Ad hoc Networks, FTRA International Symposium on Advances in Cryptography, Security and Applications for Future Computing. ACSA. 2011.

- [27] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J., Hernández-Goya, C., A. Fúster-Sabater. Gestión de Grupos en VANETs: Descripción de fases. XI Reunión Española sobre Criptología y Seguridad de la Información RECSI. 2010.

- [31] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P., Martín-Fernández, F., Yánes-García, D. Introducing Secure and Self Organized Vehicular Ad-hoc Networks. International Conference on Computer Systems and Technologies. CompSysTech. 2011.

- [36] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil. J. An EPC Gen2 Compliant Authentication Scheme Based on a New Pseudorandom Number Generator. The 2011 FTRA International Workshop on Strategic Security Management for Industrial Technology. SSMIT, 2011.

- [39] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Hernández-Goya, C. Flexible Authentication in Vehicular Ad-hoc Networks. 15th IEEE Asia-Pacific Conference Communications APCC, 576-879. 2009.

- [41] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Yánes-García, D., Martín-Fernández, F. Detecta Atascos y Aparcamiento en tu Móvil. Salón atlántico de logística y transporte. SALT2011, Las Palmas de Gran Canaria, 2011.

- [42] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Yánes-García, D., Martín-Fernández, F. Vaipho: Una Herramienta para la Asistencia a la Conducción. En VIII

Foro de innovaciones tecnológicas para el transporte. TRANSNOVA2011, Las Palmas de Gran Canaria. 2011.

- [46] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C., Hernández-Goya, C. Stimulating Cooperation in Self-Organized Vehicular Networks. 15th IEEE Asia-Pacific Conference on Communication APCC, 346-349. 2009.

- [105] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Cooperation Requirements for Packet Forwarding in Vehicular Ad-hoc NETworks (VANETs). International Conference on Computer Systems and Technologies. CompSysTech. 2009.

- [140] Martín-Fernández, F., Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Implementación de Comunicaciones Seguras en Plataformas Móviles para Asistencia a la Conducción. Enviado a X Congreso de Ingeniería del Transporte. 2012.

- [147] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Herramientas para la Seguridad Cooperativa en Redes Ad-hoc. II Simposio sobre Seguridad Informática - Congreso Español de Informática CEDI. 2007.

- [153] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Reputation Lists and Groups to Promote Cooperation. International Conference on Computer Systems and Technologies. CompSystech. 2011.

- [154] Molina-Gil, J.; Caballero-Gil, P.; Caballero-Gil, C., Hernández-Goya, C. Agregación de Datos para Autenticar Información en VANETs XI Reunión Española sobre Criptología y Seguridad de la Información. Vol. 6927, RECSI 2010, 2010

1.5.4. Otras Contribuciones

- [29] Caballero-Gil, C., Molina-Gil, J. Primer Premio del Concurso de Emprendedores "Conocer es Valer". http://emprendeull.ning.com/profiles/blogs/entrega-de-premiosdel-concurso-conocer-es-valer. Universidad de La Laguna. Importe: 3.000 Euros. 2011.

- [35] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Sistema de Comunicaciones Seguras en una Red Ad-hoc Vehicular Espontanea y Autogestionada. Patente No. P201000865. Fecha de prioridad: 29 de Junio de 2010. International Patent No. PCT/ES 2011/000220. 29 June 2011. Universidad de La Laguna. Tenerife. Spain. Licencia de Comercialización Adquirida por Empresa DETECTOR, S.A., 2011

- [139] Martín-Fernández F. Proyecto Fin de Carrera Dirigido por Caballero-Gil P., Caballero-Gil C. Implementación de comunicaciones seguras en la plataforma Symbian para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), June 2011.

Capítulo 2

Redes Móviles Ad-hoc (MANETs)

Para resolver el problema de la autenticación y gestión de claves públicas, las redes convencionales cableadas e inalámbricas centralizadas utilizan principalmente CAs. Sin embargo, la autenticación en redes ad-hoc en general, y MANETs en particular, es en general mucho más difícil de resolver que en las redes cableadas debido sobre todo a su descentralización y auto-organización. Dichas características, junto a la movilidad de los nodos, hacen necesaria la definición de un esquema del ciclo de vida de cada nodo y de la red. Precisamente los tres problemas mencionados: ciclo de vida, autenticación y gestión de claves públicas son analizados en el presente capítulo.

2.1. Estado del Arte

Para el estudio y posterior escritura de este capítulo ha sido necesaria la consulta de innumerables trabajos tales como artículos, proyectos, transparencias, libros, tesis, etc., sobre seguridad en general, encaminamiento seguro, autenticación de nodos, gestión de claves, etc., en redes inalámbricas en general y MANETs en particular. A continuación se hace un breve repaso del estado del arte en algunas de esas cuestiones.

La autogestión de claves públicas en redes ad-hoc basada en cadenas de certificados se apoya en grafos que representan los certificados y que exhiben el fenómeno de "El Mundo es un Pañuelo" (small-world phenomenon) [121] según el cual es muy probable que cualesquiera dos vértices del grafo estén conectados mediante un camino corto. Este

fenómeno fue analizado en 1960 mediante una serie de experimentos que demostraron que en media se puede unir a cualquier par de personas en EE.UU. mediante una cadena de conocidos de cómo máximo 5 ó 6 personas. Desde entonces se conoce esto como el principio "de los seis grados de separación". Este fenómeno emerge de forma natural en los sistemas de seguridad auto-organizados, tales como PGP.

Gestión del Ciclo de Vida

En una MANET, dada la movilidad de los nodos y la ausencia de infraestructura centralizada, la gestión del ciclo de vida tanto de los nodos como de la propia red puede recaer en manos de los nodos. El objetivo de dicha gestión es controlar la topología de la red mediante el control de inserciones, accesos, borrados, etc. Entre las características deseadas para cualquier esquema de gestión del ciclo de vida destacan: reducir el consumo de energía de los nodos, reducir las comunicaciones necesarias para dicha gestión, y aumentar la eficacia de la red ante los distintos procesos. Un método ampliamente estudiado de control de la topología de una red inalámbrica se basa en el ajuste de las potencias de transmisión de los nodos para definir las actividades que puede hacer cada nodo. En los últimos años, el problema de control de topología y gestión del ciclo de vida de las redes ad-hoc ha sido estudiado por varios grupos de investigadores [50],[63],[120],[122],[132],[170]. Sin embargo en todos esos trabajos se ha supuesto que los nodos son estacionarios mientras que aquí nos enfrentamos al problema de control de topología y gestión del ciclo de vida de nodos móviles. Precisamente uno de los principales problemas que abordamos es el de la autenticación de dichos nodos móviles para el control de acceso a la red.

En [108] se encuentra un estudio de distintos tipos de autenticación de nodos en redes ad-hoc incluyendo, entre otros, los siguientes:

- Modelo Wi-Fi [111]: Utiliza el estándar IEEE 802.11 para definir el método de autenticación.

- Modelo Bluetooth [186]: Estándar IEEE 802.15 en el que la contraseña para la autenticación debe ser introducida a mano en cada uno de los dispositivos.

- Modelo Resurrecting Duckling [188]: Modelo que propone el contacto físico para realizar la autenticación basada en contraseñas compartidas.
- Modelo de predistribución de claves [77]: A cada dispositivo se le da un conjunto de claves con las que se puede autenticar frente a sus nodos vecinos.
- Modelo password [9]: Se comparte una contraseña corta que debe ser introducida a mano en cada uno de los dispositivos.
- Modelo de cadenas de claves [126]: No requiere ni la presencia de autoridad certificadora ni el uso de certificados, pero proporciona sólo autenticación unidireccional.
- Modelo basado en identidad [184]: Utiliza información asociada a la identidad como clave pública y necesita de una autoridad certificadora para generar y repartir inicialmente dichas claves.

En cuanto a las MANETs, recientemente se ha propuesto un número significativo de protocolos de autenticación específicos [75], [93], [123]. Sin embargo, el logro de los requisitos básicos de seguridad atendiendo a las características específicas de dichas redes continúa siendo considerado un tema abierto. Uno de los acercamientos más elementales encontrado en la bibliografía utiliza terceras partes de confianza (TTPs, Trusted Third Parties), [200] para garantizar la validez de todas las identidades y claves de los nodos, de modo que cada nodo que desee entrar a formar parte de la red tiene que conseguir un certificado de una TTP. Un segundo paradigma de autenticación que se ha utilizado en MANETs se basa en las llamadas cadenas de confianza [167]. Una tercera solución típica es la autenticación basada en el establecimiento de límites a la localización de los nodos, que utiliza el hecho de que muchas MANETs se ubican en áreas pequeñas por lo que se puede realizar autenticación física entre nodos cercanos [69].

Gestión de Claves Públicas

El uso de técnicas de cifrado de clave pública y/o de firmas digitales requiere sistemas de certificación y gestión de las claves públicas para establecer relaciones de confianza en las claves de las distintas entidades. Con frecuencia, este servicio es proporcionado por

una TTP en la que confían todos los nodos de la red. En el caso de sistemas criptográficos de clave pública establecidos en redes tradicionales, la TTP suele ser una autoridad de certificación. Sin embargo, dicho modelo no resulta adecuado para las redes ad-hoc, en las que las tendencias encontradas en la bibliografía se inclinan por modelos tales como el de una autoridad de certificación o CA (Certificate Authority) distribuidora, donde la función de la autoridad certificadora se distribuye entre los nodos de la red de forma que cada miembro de la red es propietario de una clave privada y otra pública, y debe guardar en el registro las claves públicas de los demás nodos. Dos alternativas más simples son el modelo de clave pública auto-certificada donde el certificado está dentro de la propia clave pública y el modelo de clave publica certificada, en el que los dispositivos intercambian su clave pública por un canal seguro de forma que así ya se considera certificada. Finalmente el modelo auto-organizado de certificación implica que los nodos de la red distribuyen sus propios certificados de clave pública, y firman certificados para otros nodos de manera que para que un nodo valide el certificado de otro nodo, ambos deben encontrar una cadena de certificados entre ellos.

Precisamente dentro del primer modelo reseñado se encuentra el trabajo [220], que propone un esquema umbral (m, n), para distribuir las funciones de la CA entre un subconjunto de m nodos de una red con n nodos. El sistema contiene tres tipos de nodos: clientes, servidores y combinadores. Los nodos cliente son los usuarios del servicio de gestión de claves. Los nodos servidores y combinadores, conjuntamente, proporcionan la funcionalidad de la CA. Cada nodo servidor mantiene una clave que le permite generar certificados parciales. Los nodos combinadores, que son también servidores, combinan certificados parciales para formar un certificado válido. Cuando el cliente desea renovar su certificado, solicita la renovación al menos a m nodos servidores. Si esta solicitud es tramitada, cada nodo servidor genera un nuevo certificado parcial. Los certificados parciales son enviados a un combinador, que genera un nuevo certificado válido para el cliente. Los certificados de todos los clientes son almacenados por los nodos servidores, de modo que estos actúan también como repositorios de claves. En [134] se propone un esquema umbral similar al anterior, sólo que en este caso las funciones de la CA se distribuyen entre todos los nodos de la red, y no entre un subconjunto de nodos servidores especializados.

En [8] se propone un servicio de gestión de claves públicas basado en criptografía simétrica. Los nodos de la red comparten una clave secreta de grupo, que se utiliza para tareas de autenticación y para generar claves de cifrado. La clave de grupo no expira, mientras que las claves de cifrado sí se actualizan en periodos regulares. Este esquema es adecuado en redes con nodos de capacidad limitada, en los que la criptografía de clave pública resulta excesivamente pesada.

En [109] se propone una solución enmarcada en el segundo modelo antes mencionado, en la que los certificados son generados por los propios nodos de la MANET, sin necesidad de una CA. Cada nodo almacena un conjunto reducido de certificados correspondientes a las claves públicas de nodos que considera válidas.

En [99] buscan mecanismos seguros la gestión de claves públicas en MANETs y en particular intentan añadir seguridad al algoritmo de encaminamiento AODV. En [130] se estudian los problemas de seguridad de las MANETs y en particular de gestión de claves. Los autores de [164] también estudian los problemas de seguridad en MANETs y proponen mecanismos de gestión de claves pero van más alla y proponen la posibilidad de crear una red autónoma y estudian sus problemas específicos.

RFID

En este trabajo proponemos el uso de tecnología RFID para controlar la topología de las MANETs mediante la adhesión de etiquetas a los nodos. En particular abordamos el problema de la autenticación mutua lector-etiqueta, que ha recibido bastante atención por la comunidad científica.

Como la tecnología RFID se está utilizando cada vez en más aplicaciones, los investigadores están prestando más atención a los problemas de seguridad y privacidad, como son los accesos no autorizados a información de identificación de las etiquetas, o la existencia de potenciales adversarios que puedan engañar al lector mediante el uso de información obtenida de etiquetas de identificación válidas. Estos problemas pueden ser resueltos a través de técnicas de autenticación [117]. En [7] se presentó un completo listado de publicaciones sobre autenticación entre etiqueta y lector. Allí se pueden encontrar varios protocolos basados en diferentes herramientas, como funciones hash, códigos de autenticación de men-

sajes, cifrados en bloque, funciones pseudo-aleatorias, etc. Sin embargo, la autenticación entre etiquetas y lectores según el estándar en EPC Gen2 puede ser considerada todavía un problema abierto porque la mayoría de las propuestas requieren demasiados recursos y/o no cumplen con la norma.

2.2. Simulación con NS-2

La naturaleza especial de las MANETs, donde la mayoría de las aplicaciones se basan en la colaboración entre nodos o bien se basan en la división de los nodos en grupos, sugiere que los mencionados acercamientos tradicionales al problema de la autenticación y gestión de claves puedan no ser siempre apropiados. Uno de los objetivos principales de este trabajo consiste en analizar los principales problemas de seguridad relativos a la autenticación de nodos y a la gestión de claves, para a continuación diseñar, implementar y evaluar varias soluciones prácticas novedosas para sendos problemas así como para la gestión de la propia red y de posibles subredes, comparándolas con las existentes en la bibliografía. Para ello es imprescindible el manejo de simulaciones de las redes en cuestión. En concreto, se propuso en primer lugar un esquema de autenticación de nodos basado en la primitiva criptográfica de la demostración de conocimiento nulo, que constituye una solución elegante, práctica y con un gran nivel de seguridad para el problema de la autenticación fuerte de nodos. Por otra parte, también se propone aquí una nueva solución al problema de la gestión de claves públicas en MANETs basada en la solución original de cadenas de confianza [52], [109], pero aportando como innovación una nueva propuesta para el algoritmo de actualización de repositorios de certificados usando nociones de Teoría de Grafos. Para ambos casos se implementaron completas simulaciones usando la herramienta de simulación más habitual en redes inalámbricas, el simulador NS-2, de cuyas ejecuciones se han extraído numerosos datos y se han sacado conclusiones que han permitido diversas mejoras y comparaciones con métodos anteriores.

En esta sección se describen algunas características fundamentales de la herramienta NS-2 utilizada así como los detalles y principales problemas encontrados durante su utilización en este trabajo. Su elección se debe a que es uno de los simuladores de mayor

difusión dentro del sector de las telecomunicaciones.

En la actualidad existen implementaciones de varios simuladores de redes: Opnet, Omnet ++, Glomosim, etc. Sin embargo, el Network Simulator-2 o NS-2 se ha convertido en un estándar debido a su amplia utilización [59]. Su estructura permite obtener una visión global de las redes, facilitando la relación de conceptos de distintas áreas, con el desarrollo de nuevos mecanismos de comunicación. A pesar de estas ventajas, su principal inconveniente es la dificultad asociada a la iniciación y el desarrollo de nuevos protocolos en este entorno.

NS-2 es un simulador en tiempo discreto que surgió en 1989 con el desarrollo de REAL Network Simulator [84]. Probablemente una de las principales razones que explican su éxito es el hecho de que la distribución posee licencia GPL, condición que impulsa su desarrollo y libre difusión. Inicialmente, NS-2 fue ideado para redes cableadas pero pronto se desarrolló una ampliación para su uso en el análisis de redes inalámbricas.

El Network Simulator se apoya en dos lenguajes de programación. Por un lado, el usuario introduce las especificaciones del escenario que desea analizar a través del lenguaje OTcl, versión extendida de Tcl. Por otro lado, la implementación de los protocolos se encuentra en C++. Como resultado de una simulación, se pueden obtener datos matemáticos para un estudio posterior, o bien, trazas específicas para visualizarlas en la herramienta NAM (Network AniMator) del NS-2.

En general, a la hora de simular mediante NS-2 cada uno de los protocolos diseñados para MANETs y VANETs, hemos seguido los siguientes pasos:

- Implementación del protocolo a analizar mediante la incorporación de código en C++ y OTcl dentro del núcleo de NS-2.
- Descripción de la simulación mediante OTcl, detallando concretamente el escenario inalámbrico a simular. En particular, la definición de la configuración de cada nodo incluye la determinación del nivel MAC a emplear, el tipo de cola para el nivel de enlace, el modelo de propagación de la señal, así como el tipo de encaminamiento ad-hoc que seguirán los terminales: AODV, DSR, DSDV, OLSR, etc.
- Ejecución de la simulación en NS-2, activando o desactivando los niveles de trazas para analizar distintos tipos de resultados.

- Análisis de resultados, mediante la herramienta denominada NAM que permite la visualización del comportamiento de los terminales de la red. Con ello, se puede apreciar cómo, por ejemplo, un paquete de datos se va encaminando a través de los distintos dispositivos, cómo se pierden paquetes a veces, etc. También es posible extraer métricas cuantitativas como resultados de las simulaciones. Para ello, a partir de los ficheros de traza que aporta la simulación, es preciso realizar un post-procesado, por ejemplo, en lenguaje PERL o awk.

NS-2 dispone de una gran cantidad de protocolos implementados, que en algunos casos han sido aquí ampliados usando lenguajes C++ y tcl.

En la simulación de una MANET, una de las cuestiones más importantes es el tema del movimiento de los nodos, ya que dicho movimiento no se puede predeterminar porque es totalmente aleatorio. Además, los nodos deben poder comunicarse con nodos que no están dentro de su rango de emisión, a través de nodos intermedios que hacen de routers de sus paquetes. Estas dos cuestiones han tenido que ser tratadas en las simulaciones realizadas en este trabajo.

A continuación se comenta a modo ilustrativo un ejemplo de red básica sin seguridad, simulada usando NS-2. En la Fig. 2.1 se representan 15 nodos situados inicialmente de manera aleatoria en el plano. Estos nodos se mueven dentro del plano pasando por diferentes situaciones, de forma que sus vecinos o nodos más cercanos, pueden cambiar a medida que aquéllos se mueven por el plano. Los diferentes nodos pueden perder la conexión con el resto de nodos si se van a un trozo del plano donde no llegue la onda expansiva de ningún otro nodo.

Los parámetros usados en la simulación estándar de los esquemas propuestos en las secciones siguientes son los siguientes:

- área: 1000 * 1000 m^2.
- nº de nodos = 100.
- máxima velocidad = de 5 m/s a 20 m/s.
- mínima velocidad = 1 m/s.

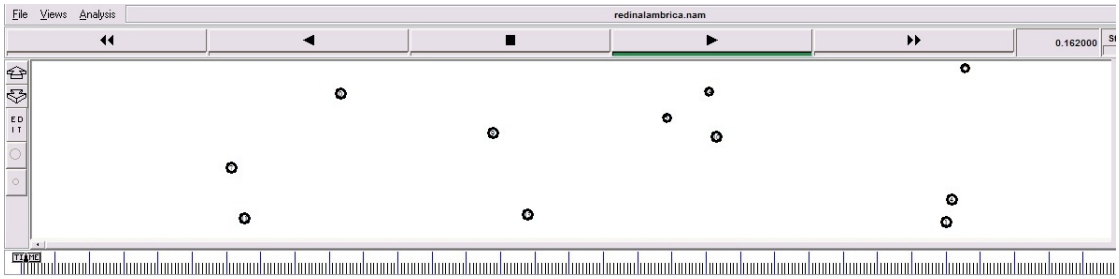


Figura 2.1: Simulación NS-2 de 15 Nodos

- t^{o} simulación = entre 100 s y 300 s.

Esta simulación incluye movimiento aleatorio según el modelo de movilidad con pausas y velocidades variables, conexiones y desconexiones aleatorias, y envío de paquetes al azar. Tanto para el movimiento aleatorio de los nodos como para el envío al azar se utilizan generadores de escenarios. La creación de escenarios de envío de paquetes entre nodos de forma aleatoria se realiza con una herramienta llamada cbrgen.tcl. Esta herramienta se ejecuta con una serie de parámetros para obtener los diferentes escenarios aleatorios.

La forma de ejecutarla es como sigue:

ns gengrafo.tcl [-nn n^{o} de nodos][-seed semilla][-mc n^{o} máximo de conexiones] > vida_de_red.tcl

Para la creación de escenarios de movimiento aleatorios se utiliza otra herramienta llamada setdest, que se ejecuta con los siguientes parámetros.

./setdest[-n numero de nodos] [-p tiempo de pausa] [-s máxima velocidad] [-t tiempo de movimiento] [-x max en x] [-y max en y] > movimiento.tcl

Para poder simular que los nodos se encienden y se apagan aleatoriamente en el plano se les debe dotar de energía. De esta manera podemos asignar energía a cada nodo, energía que van perdiendo con el envío y recepción de cada paquete. También es posible quitar o poner energía a nodos en momentos puntuales con *setenergy*. Para hacer todo esto de forma aleatoria, se han introducido las sentencias correspondientes en el generador de envíos *cbrgen.tcl* indicando cuándo entra un nodo en la red inalámbrica y cuándo desaparece de la red.

La implementación de todas estas y otras opciones ha sido posible gracias a la información obtenida del manual de NS-2 [84]. Para automatizar la ejecución de la simulación se han generado numerosos ejecutables con diferentes opciones de envío y recepción de datos, encendido y apagado de los nodos y movimiento en el plano, todo esto de forma aleatoria.

2.3. Sistema SLCM de Gestión del Ciclo de Vida Auto-Organizada

Las MANETs no tienen ningún tipo de infraestructura establecida, razón por la cual los nodos deben adaptarse sólos a las cambiantes situaciones dinámicas que se derivan de su movilidad. Debido a la descentralización de los nodos y a las necesidades de seguridad de las comunicaciones inalámbricas, lo ideal es que la gestión de las MANETs sea auto-organizada, lo que constituye un reto de investigación. Con el fin de hacer frente a las propiedades intrínsecas de las MANETs, un nuevo sistema de gestión descentralizado para MANETs llamado Gestión del Ciclo de Vida Auto-Organizado (SLCM por sus siglas en inglés Self-organizing Life Cycle Management), es aquí propuesto y evaluado.

En cuanto a la seguridad, la autenticación de los nodos es el componente más crítico ya que define el control de accesos a la red. Por otra parte, la retransmisión masiva o broadcast inalámbrico es también un mecanismo fundamental para la difusión eficiente de los datos en estas redes. Ambos aspectos han recibido especial atención al definir el sistema SLCM propuesto. En particular, tanto un algoritmo de control fuerte de accesos basado en el paradigma criptográfico de las demostraciones de conocimiento nulo, como un protocolo de retransmisión en tres etapas, son aquí propuestos. Esta sección incluye la evaluación del desempeño del sistema SLCM propuesto de forma que los resultados experimentales obtenidos muestran que el esquema SLCM mejora significativamente la calidad y la seguridad de la gestión del ciclo de vida de las MANETs auto-organizadas.

2.3.1. Planteamiento del Problema

A diferencia de la arquitectura en otro tipo de redes, en una MANET cada nodo puede funcionar como host y router al mismo tiempo, y cada movimiento de un nodo afecta a la topología de la red. Por lo tanto, el encaminamiento en redes ad hoc es uno de los problemas más estudiados en la bibliografía [112], [219]. Además, la falta de infraestructura central también hace que sea difícil, si no imposible, la existencia de una autoridad que gestione el funcionamiento y la seguridad de la red. El trabajo descrito en esta sección ha sido elaborado con el objetivo de resolver estos problemas.

La limitación de recursos y en particular, las limitaciones en la comunicación y la computación, la implementación gradual y la necesidad de escalabilidad, la falta de infraestructura central o fija, y la falta de fiabilidad de los medios de comunicación de radio son algunos de los principales retos que se deben tener en cuenta al diseñar un protocolo de gestión de MANETs.

En esta sección nos centramos en el seguimiento y la gestión de la configuración de las MANETs, es decir, en el proceso de controlar a los nodos que acceden y se comunican así como los datos que se envían en estas redes con el fin de maximizar su seguridad y garantizar su eficiencia. En particular, aquí se describen todas las fases del esquema de gestión de ciclo de vida propuesto. Un requisito básico para configurar la MANET se encuentra en la capacidad de auto-organización de los nodos de la red.

En el presente trabajo, cada nodo legítimo presenta sus credenciales a otro nodo legítimo cuando intenta acceder a la red, de acuerdo a un proceso de autenticación basado en el paradigma de las ZKPs. Hasta ahora, sólo algunas publicaciones han mencionado la propuesta de sistemas de autenticación de MANETs usando ZKPs [33], [6], pero ninguna incluye dicha propuesta de autenticación basada en ZKP en el contexto de un sistema de gestión completa del ciclo de vida para tratar el problema relacionado con los cambios de topología debido a la movilidad, que es exactamente el objetivo principal del trabajo expuesto en esta sección.

2.3.2. Objetivos de Seguridad e Hipótesis de Trabajo

El esquema SLCM propuesto está pensado para MANETs de pequeño y mediano tamaño que requieran seguridad en las comunicaciones a pesar del gasto adicional que implica el envío de paquetes de control. Como la MANET no tiene ningún tipo de infraestructura fija, su capacidad para soportar el encaminamiento es limitado, por lo que el esquema propuesto no resulta apropiado para grandes redes ya que aumentaría demasiado la complejidad y el número de paquetes de control.

La eficiencia, fiabilidad y seguridad son los principales objetivos de diseño para la gestión del ciclo de vida auto-organizada para MANETs que proponemos en este trabajo. Con el fin de describir en detalle los objetivos de seguridad, debemos distinguir entre los nodos que están fuera de la red y los que están dentro. Un nodo externo es un nodo que no es un miembro autorizado de la red, mientras que un nodo interno es un miembro autorizado legítimo de la red. El objetivo de seguridad de esta investigación es el desarrollo de mecanismos que protejan a una MANET auto-organizada sin ningún tipo de autoridad central contra los comportamientos maliciosos tanto de los nodos externos, como de los nodos internos.

Dado que desde dentro los nodos tienen acceso a todos los recursos de la MANET, es fácil para ellos lanzar ataques más sofisticados. En este trabajo se propone un sistema de respuesta que proporciona la capacidad para cortar con eficacia el acceso a las funcionalidades de la MANET por parte de nodos internos comprometidos. Además, el sistema ofrece un nivel de protección contra nodos internos que lanzan ataques de suplantación, intentando hacerse pasar por otros nodos internos.

La detección de ataques por parte de nodos externos es una de las tareas de los Sistemas de Detección de Intrusiones (IDS, Intrusion Detection System). A continuación se describen los principales objetivos de seguridad para la defensa de la red contra los nodos externos.

Un requisito importante es que cualquier paquete transmitido por un nodo externo debe ser rechazado inmediatamente por el primer nodo interno que reciba dicha información con una probabilidad muy alta. En otras palabras, no se debe permitir que se difundan a

través de la MANET paquetes enviados por nodos ajenos a la red. Al cumplir con este requisito, por lo que se deshabilita de forma efectiva la capacidad de los nodos externos de encaminar cualquier paquete a cualquier nodo que no sea su vecino, se puede evitar de forma satisfactoria un gran número de ataques lanzados por nodos externos, tales como los ataques por Denegación de Servicio (DoS, Denial of Service), de agujero de gusano, del intermediario (MitM, Man-in-the-Middle), inundaciones SYN, etc. Sin embargo, el requisito previamente mencionado dicta que cada paquete tiene que ser autenticado en cada salto (hop), lo que a su vez significa que el mecanismo de autenticación debe ser muy eficiente.

Por otro lado, en un esquema realista se debe suponer que un nodo externo puede tener la capacidad de suplantar la identidad de un nodo interno, usando datos como su dirección IP y direcciones MAC, por lo que estos datos no deben considerarse fiables en el esquema.

Se debe suponer también siempre que los nodos externos tienen acceso al canal inalámbrico y que pueden escuchar el tráfico de información. Por lo tanto, si la información transmitida es confidencial, debe usarse un cifrado extremo a extremo para protegerla y en cualquier caso para que no pueda usarse esa información para lanzar ningún tipo de ataque.

Si se utiliza un sistema para descubrir posibles nodos internos comprometidos, el sistema debe ser capaz de excluirlos para que no propaguen ningún paquete dentro de la red. Respecto a este tema, la lista de certificados revocados (CRL, Certificate Revocation List) puede ser utilizada para revocar los certificados de los nodos internos comprometidos. Lo ideal sería que esta CRL sea actualizada y enviada a toda la red cada vez que se detecta a un nodo malicioso para que cada nodo de la red pueda actualizar la CRL que tiene almacenada. En el esquema propuesto, si un nodo no actualiza su CRL porque está apagado o fuera de cobertura, podrá actualizar la versión de la CRL que tenga almacenada en el momento en que acceda de nuevo a la red. Con el fin de detectar posibles nodos internos comprometidos, los nodos de la red deben verificar la información enviada por sus vecinos de forma que si varios nodos reciben información incorrecta o inconsistente en repetidas ocasiones enviada desde la misma fuente, este grupo de nodos debe introducir la información del nodo sospechoso en la CRL. Lo conveniente es establecer para ello un número mínimo de nodos que deben estar de acuerdo con firmar la revocación del nodo sospechoso de manera

que dicho umbral depende de varios factores, tales como el tamaño de la MANET.

2.3.3. Protocolo GRI de Broadcast Optimizado

En este trabajo no se proponen nuevos esquemas de encaminamiento ya que las simulaciones realizadas para nuestro sistema SLCM muestran que los sistemas existentes de encaminamiento en redes ad-hoc, tales como DSR u OLSR dan buenos resultados sin saturar las comunicaciones.

A continuación se describe un protocolo optimizado para broadcast en MANETs, que se utiliza en el esquema SLCM. El nuevo protocolo de retransmisión llamado *GRI* (de sus siglas en inglés Go-Return-Information) es un esquema de broadcast optimizado diseñado para resolver algunos problemas existentes en las comunicaciones inalámbricas en redes sin autoridad centralizada. El protocolo consta de tres simples fases llamadas: Ida, Retorno e Información.

En la primera etapa de Ida, el nodo origen que inicia la transmisión del broadcast GRI envía una solicitud de respuesta a todos los nodos que están dentro de su rango de transmisión y cada nodo que recibe este mensaje lo reenvía a sus nodos vecinos.

En la fase de Retorno, los nodos que están más lejos del nodo que inició la difusión del broadcast GRI, es decir, los nodos que no tienen a nadie nuevo a quien enviarle el mensaje, comienzan la fase de retorno del broadcast GRI, enviando sus identificadores hacia el nodo que inició la difusión del broadcast GRI. Los nodos tienen un límite de tiempo de espera para enviar su respuesta de retorno al nodo origen, así que cada nodo de la red debe responder durante ese tiempo hacia el nodo origen. Cuando la respuesta de retorno pasa por los nodos intermedios, estos añaden su identificación al paquete de respuesta y lo retransmiten hacia el nodo origen.

En la fase de Información, el nodo que inició la difusión del broadcast GRI recibe toda la información de los nodos legítimos y conectados a la red y posteriormente, envía un nuevo broadcast con toda la información recibida sobre todos los nodos de la red. Con este simple protocolo es posible informar a toda la red sobre qué nodos están conectados en ese momento, generando un menor número de paquetes de control en la red que si cada nodo por separado lanzara en paralelo un broadcast con el mismo objeto.

En situaciones especiales, como por ejemplo si los nodos están colocados en línea y el nodo que inicia el broadcast se encuentra en un extremo, el tamaño del paquete de retorno podría llegar a ser muy grande, porque contendrá la información obtenida de todos los nodos de la red. No obstante, este no es un caso habitual.

Como ejemplo ilustrativo, la Fig. 2.2 muestra una comparación entre los paquetes generados en una situación habitual sin y con protocolo GRI obtenida mediante la herramienta TraceGraph. Si comparamos el número de paquetes generados por 20 nodos mediante un broadcast masivo y mediante un broadcast GRI se puede apreciar que el número de paquetes generados por el broadcast GRI está entre el 40 % y el 60 % del número de paquetes generados mediante un broadcast masivo. De esta forma, el gráfico muestra claramente que los resultados son mejores usando broadcast GRI que masivo.

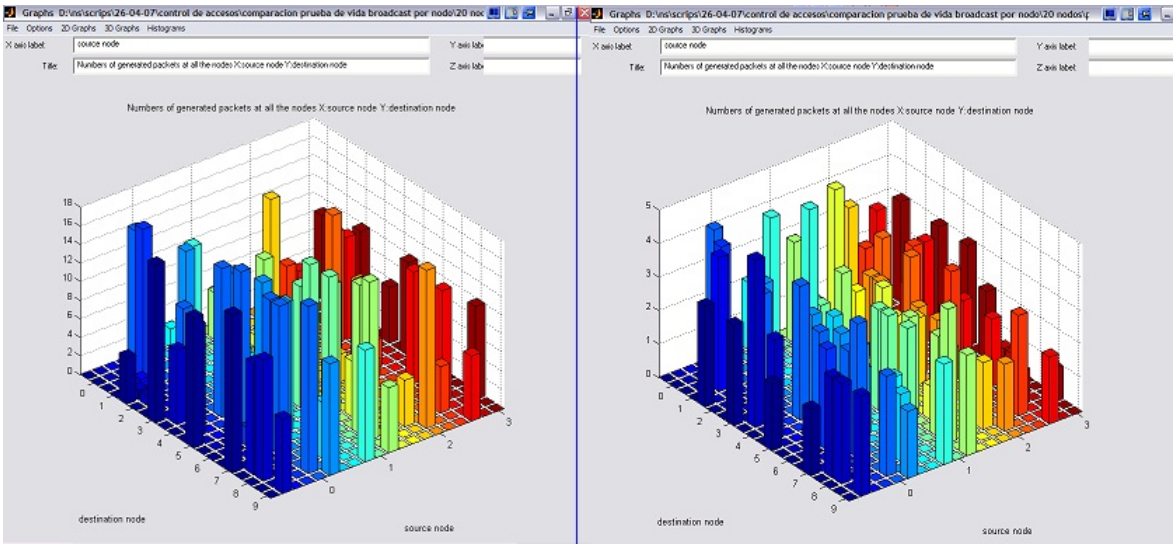


Figura 2.2: Broadcast Masivo frente a Broadcast GRI

2.3.4. Aspectos Generales de la Propuesta

El esquema SLCM que aquí se presenta ha sido diseñado como un esquema de autenticación para pertenencia a un grupo, porque cuando un nodo quiere convertirse en parte de la red, antes debe ser aceptado por suficientes nodos legítimos. El número mínimo de nodos internos necesarios para la inserción de cualquier nodo debe ser lo suficientemente

grande como para asegurar que posibles atacantes no puedan capturar suficientes nodos internos como para introducir nuevos nodos maliciosos. Este número depende fundamentalmente del tamaño de la red.

Según los autores de [136], en cualquiera de los protocolos de gestión de grupos es necesario establecer métodos robustos para insertar y eliminar nodos, y para permitir el acceso sólo a los miembros legítimos del grupo. Por esta razón, aquí no sólo se describe el procedimiento para controlar el acceso a la red de los nodos internos, sino también los procedimientos para la actualización de la red asociada a las inserciones y eliminaciones de nodos. En particular, describimos cómo decidir qué nodos se eliminan de la red basándonos en el momento en el que el nodo se desconectó, por lo que si un nodo ha estado desconectado durante mucho tiempo (en comparación con un parámetro umbral preestablecido), se elimina de la red.

El paradigma criptográfico de la ZKP es la base teórica del procedimiento de control de acceso que se describe a continuación. En particular, el protocolo se aplica para el caso particular del problema del circuito hamiltoniano. Este problema fue elegido para nuestro diseño principalmente porque la actualización de una solución debido a una inserción o eliminación de un vértice del grafo no requiere un gran esfuerzo computacional. Estas operaciones son comunes en nuestra aplicación debido al gran dinamismo de las redes analizadas. Sin embargo, se podrían describir sistemas similares sobre la base de otros problemas NP-completos en grafos donde la actualización de una solución después de cambios individuales en el grafo también sea una tarea fácil. Este es el caso de problemas como el del Recubrimiento de Vértices, Conjunto Independiente o Clique, por ejemplo [32].

El correcto desempeño del sistema propuesto es sólo posible gracias a la utilización de una aplicación de chat a través del broadcast GRI propuesto, ya que permite que algunos nodos legítimos y en cobertura puedan enviar mensajes a todos los nodos de la red que se encuentren conectados. Dicha aplicación de chat permite publicar toda la información relacionada con la actualización de la red. No será necesario que los mensajes de chat se retransmitan en secreto porque la información publicada es inútil para los nodos ilegítimos, ya que sólo sirve para que los nodos legítimos actualicen su información acerca de la autenticación.

Todos los datos recibidos a través del broadcast GRI de la aplicación de chat son almacenados por cada nodo que se encuentre en cobertura en una cola *FIFO (First In First Out)* durante un intervalo de tiempo suficiente. La duración de este período, que se denota por T, es un parámetro esencial, ya que indica el tiempo máximo permitido que un nodo legítimo puede estar fuera de línea sin perder su condición de nodo interno, y también la frecuencia con la que se deben realizar las pruebas de vida que se describen a continuación. Por lo tanto, este parámetro debe ser acordado por todos los nodos legítimos de la red.

El ciclo de vida de la red tiene tres fases principales, como se muestra en la Fig. 2.3. La inicialización es la primera fase. En ella, cada miembro de la red original recibe, sin importar si está fuera de línea o en línea, una pieza de información secreta que juega el papel de clave secreta de la red. El conocimiento de dicha clave secreta de la red se utilizará luego para realizar el control de acceso y demostrar la fiabilidad de cada nodo con el fin de acceder a recursos protegidos o para ofrecer algún servicio a la red.

Después de la fase de inicialización, los nodos legítimos pueden participar en la red, por lo que el ciclo de vida del nodo comienza. (Ver Fig. 2.4).

Un nodo legítimo que se haya desconectado de la red debe demostrar su pertenencia de la red a través de un control de acceso por un nodo legítimo que se encuentre en línea. Con el fin de hacerlo, el nodo debe demostrar el conocimiento de la clave secreta de la red mediante un esquema de reto-respuesta.

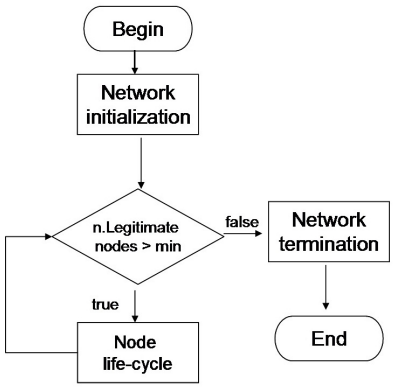


Figura 2.3: Ciclo de Vida de la Red

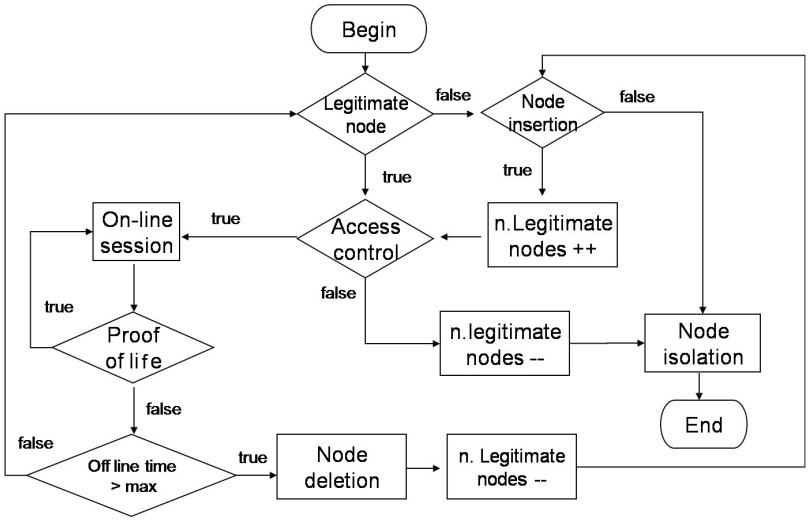


Figura 2.4: Ciclo de Vida del Nodo

Cuando a un nodo legítimo se le da permiso para acceder al estado conectado, tiene pleno acceso tanto a los recursos protegidos, tales como la emisión de mensajes a través de la aplicación de chat mediante broadcast GRI, como para proporcionar servicios de red tales como la inserción de nuevos nodos y el control de acceso a nodos internos.

La clave de red secreta se actualiza continuamente de acuerdo a los cambios en la topología de la red, por lo que la clave secreta de un nodo legítimo caduca si el nodo deja de estar en línea durante mucho tiempo y no puede actualizarla. En ese caso, el nodo tendría que volver a ser insertado en la red por un nodo en línea legítimo si quiere participar nuevamente en la red.

En nuestra propuesta, la clave de red secreta se basa en la dificultad del HCP, por lo que el número de nodos legítimos es un parámetro muy influyente para dicha dificultad. Por lo tanto, si el número de nodos legítimos disminuye y se vuelve demasiado pequeño, la red finaliza y su ciclo de vida termina automáticamente.

A pesar del posible acceso a toda la información enviada tanto a través del broadcast GRI como en el intercambio de datos entre un nodo interno candidato a conectarse a la red y el nodo verificador durante el procedimiento de control de acceso debe desaparecer como tal por lo que ningún adversario puede acceder a ningún tipo de información que sea

significativa.

La notación utilizada en la propuesta descrita a continuación es la siguiente:

- $G_t = (V_t, E_t)$ denota el grafo no dirigido utilizado en la fase t del ciclo de vida de la red.
- $v_i \in V_t$ representa tanto un vértice del grafo G_t como un nodo legítimo de la red.
- $n = |V_t|$ es el orden de G_t, que coincide con el número de nodos legítimos de la red.
- $m \leq |E_t|$ es una cota inferior del número de aristas del grafo G_t.
- r es un número aleatorio grande.
- $N_{G_t}(v_i)$ denota a los vecinos del nodo v_i en el grafo G_t.
- $\Pi(V_t)$ representa una permutación aleatoria sobre el conjunto de vértices V_t.
- $\Pi_i(V_t)$ denota una permutación aleatoria sobre V_t elegida por v_i.
- $\Pi(G_t)$ indica el grafo isomorfo G_t que corresponde a la permutación $\Pi(V_t)$.
- $c \in_r C$ indica que un elemento c se elige al azar con una distribución uniforme de un conjunto C.
- HC_t designa el circuito hamiltoniano utilizado en la fase t.
- $\Pi(HC_t)$ representa el circuito hamiltoniano HC_t en el grafo $\Pi(G_t)$.
- $N_{HC_t}(v_i)$ denota los vecinos del nodo v_i en el circuito hamiltoniano HC_t.
- S y A representan al nodo que pretende acceder y al nodo autenticador, respectivamente, tanto durante la fase de inserción como durante la ejecución del procedimiento de control de acceso basado en ZKP.
- $S \rightleftharpoons A$ simboliza cuando el nodo S contacta con el nodo A.
- $A \leftrightarrow S : data$ significa que A y S se ponen de acuerdo en los datos $data$.

- $A \xrightarrow{s} S : information$ significa que A envía $information$ a S a través de un canal seguro.
- $A \xrightarrow{o} S : information$ significa que A envía $information$ a S a través de un canal abierto.
- $A \xrightarrow{b} network : information$ representa cuando A retransmite mediante broadcast GRI $information$ a todos los nodos legítimos y en línea de la red.
- $A \xleftrightarrow{b} network : information$ representa un procedimiento de dos etapas, donde A retransmite mediante broadcast GRI la información a todos los nodos legítimos en línea de la red, y recibe sus respuestas.
- h representa una función hash pública.
- T denota la longitud umbral del máximo periodo permitido fuera de línea para los nodos internos.

2.3.5. Fases del Sistema SLCM

Esta sección especifica todos los detalles acerca de las fases de inicialización, inserción de nodos, control de acceso, pruebas de vida y eliminación de nodos que se pueden dar en la red.

Inicialización

El conjunto de vértices del grafo a utilizar en el esquema SLCM se corresponde exactamente con el conjunto de nodos que conforman la red durante todo el ciclo de vida. En consecuencia, el proceso de inicialización parte de un conjunto V_0 de n vértices correspondientes a los nodos de la red inicial. Además, cada subíndice del vértice puede ser utilizado como *ID (IDentificación)* del nodo correspondiente en la red. El primer paso de la inicialización consiste en generar de forma conjunta en secreto al azar una permutación Π de este conjunto. El algoritmo para generar el ciclo HC_0 implica tres pasos básicos. En primer lugar, a cada nodo se le asigna un número diferente $v_i \in [1, n]$ de acuerdo con el

orden de las direcciones IP de los nodos. Entonces cada nodo genera una permutación aleatoria $\Pi_i(V_t)$ y la comparte con el resto de nodos iniciales a través una conexión Bluetooth segura. Finalmente, cada nodo calcula el producto de todas las permutaciones con el fin de obtener la permutación $\Pi(V_t)$ que utiliza en el esquema. Una vez hecho esto, cada nodo legítimo debe conocer el circuito hamiltoniano HC_0 que corresponde exactamente a dicha permutación $\Pi_i(V_t)$. El grafo parcial formado por las aristas que se corresponden con el circuito hamiltoniano HC_0 es entonces completado añadiendo n grupos de $[2m/n] - 2$ aristas elegidas al azar fuera de HC_0, e independientemente por los n nodos iniciales y comunicados entre ellos a través de la conexión Bluetooth segura, produciendo el conjunto de aristas iniciales E_0. Cada uno de estos n grupos de aristas deben tener al menos un vértice final v_i, $i = 1, 2, ..., n$, correspondiente al nodo que lo generó al azar. La cardinalidad de cada grupo de aristas debe ser lo suficientemente grande como para que la cardinalidad del conjunto de aristas distintas resulte $|E_0| \geq m$ y garantize la dificultad del HCP en G_0.

Algoritmo de Inicialización

Entrada: V_0, con $|V_0| = n$

1. Los n nodos de la red generan conjunta, secreta y aleatoriamente el circuito $HC_0 = \Pi(V_0)$.
2. Cada nodo $v_i \in V_0$ construye el conjunto $N_{G_0}(v_i) = \{\{v_j \in_r V_0\} \cup N_{HC_0}(v_i)\}$ con $|N_{G_0}(v_i)| = [\frac{2m}{n}]$.
3. $v_i \xrightarrow{b} network : N_{G_0}(v_i)$ Cada nodo retransmite a todos el conjunto de vecinos creado por el v_i
4. Cada nodo combina todos los conjuntos de nodos vecinos, formando el grafo inicial con el conjunto de aristas $E_0 = \bigcup_{i=1,2,...,n} \{(v_i, v_j) : v_j \in N_{G_0}(v_i)\}$.

Salida: $G_0 = (V_0, E_0)$, con $|E_0| \geq m$

Inserción

La fase de inserción descrita en esta sección funciona bajo el supuesto de que existe confianza mutua y una conexión Bluetooth segura entre el nodo legítimo autenticador A y el

nuevo nodo solicitante S. El primer paso que el nodo A debe llevar a cabo es asignar al nuevo nodo S el menor número de vértice v_i no asignado a ningún nodo en el conjunto de vértices V_t. Esto significa que se puede usar un número previamente utilizado por otro nodo que haya sido eliminado o usar un número nuevo v_{n+1}. Después, se debe transmitir vía broadcast GRI un aviso de dicha asignación a todos los nodos que se encuentren conectados, con el fin de evitar otra inserción simultánea con el mismo identificador. Tras recibir la respuesta de la red, si A recibe menos de $n/2$ respuestas, se detiene el procedimiento de inserción porque el número de nodos conscientes de la inserción no se considera lo suficientemente grande. De lo contrario, A elige la actualización correspondiente del circuito hamiltoniano secreto HC_t seleccionando al azar dos vértices vecinos v_j y v_k con el fin de insertar el nuevo nodo v_i entre ellos, eligiendo al azar un conjunto de $[2m/n] - 2$ nodos en V_t de tal forma que ninguno de ellos sean vecinos en HC_t, y finalmente retransmite el conjunto de vecinos $N_{Gt+1}(v_i)$ de S en el nuevo grafo G_{t+1} hacia todos los nodos legítimos y en línea de la red.

Algoritmo de Inserción

Entrada: En la etapa t un nodo suplicante S quiere convertirse en nuevo miembro de la red.

1. $S \rightleftharpoons A$ y nodo S convence al nodo A para aceptar su entrada a la red, mediante contacto físico o confianza previa.

2. A asigna a S el número de vértice v_i tal que $i = min\{l : v_l \notin V_t\}$

3. $A \overset{b}{\leftrightarrow} red : v_i$, A retransmite dicho número a la red mediante broadcast GRI.

4. - Si A recibe menos de $n/2$ respuestas, se detiene el procedimiento de inserción.
   - En otro caso:
     (a) A elige al azar $\{v_j \in_r V_t, v_k \in_r N_{CH_t}(v_j)\}$
     (b) A elige al azar $N_{G_{t+1}}(v_i) = \{v_j, v_k\} \cup \{w_1, w_2, ..., w_{[\frac{2m}{n}]-2} \in_r V_t$ tal que $\forall w_l \notin \{v_j, v_k\}$
     (c) $A \overset{b}{\rightarrow} red : N_{G_{t+1}}(v_i)$, A envía a todos los nodos de la red mediante broadcast el conjunto de vecinos sugeridos.

(d) Cada nodo en línea calcula $V_{t+1} = V_t \cup \{v_i\}$, $E_{t+1} = E_t \cup N_{G_{t+1}}(v_i)$ y $HC_{t+1} = \{HC_t \setminus (v_j, v_k)\} \cup \{(v_j, v_i) \cup (v_i, v_k)\}$

(e) $A \xrightarrow{o} v_i : G_{t+1}$, A envía abiertamente al nodo suplicante el grafo actualizado.

(f) A, $A \xrightarrow{s} v_i : HC_{t+1}$ envía de forma segura al nodo suplicante el circuito hamiltoniano actualizado.

Salida: El nodo solicitante S es un miembro legítimo de la red.

Control de Accesos

Si un nodo legítimo S que ha estado fuera de línea durante un tiempo t se quiere volver a conectar en línea a la red en la etapa r, primero contacta con un miembro A legítimo y en línea. Posteriormente, A debe comprobar si el período en el que el nodo S ha estado fuera de línea no es mayor que T. En este caso, S tiene que ser autenticado por A a través de una ZKP sobre su conocimiento de la solución secreta HC_t en el grafo G_t. El ajuste del parámetro T se basará en el tiempo medio que los nodos legítimos de la red han estado fuera de línea con anterioridad añadiendo la desviación correspondiente típica. Este valor debe ser actualizado regularmente después de cada control de acceso que se realize con éxito. La inicialización de T se realiza con un valor suficientemente grande.

Algoritmo de Control-de-Accesos

Entrada: En la etapa r un nodo suplicante S que ha estado fuera de línea desde la fase t se quiere conectar en línea a la red.

- $S \rightleftharpoons A$
- $S \xrightarrow{o} A : G_t$, S envía abiertamente a A el grafo G_t.
- A comprueba si $t \leq r - T$
  - si $t \leq r - T$, entonces la autenticación de S falla
  - en otro caso:
    - $A \leftrightarrow S : l$, A y S acuerdan un nº de iteraciones l.
    - $\forall j \in \{1, 2, \ldots, l\}$

1. S escoge una permutación $\Pi_j(V_t)$, y $\Pi_j(G_t)$ y $\Pi_j(HC_t)$, grafo isomorfo a G_t y correspondiente circuito hamiltoniano, respectivamente
2. S genera dos números aleatorios grandes r_1 y r_2
3. $S \xrightarrow{o} A : \{h(\Pi_j(G_t)||r_1), h(\Pi_j(HC_t)||r_2)\}$, S envía abiertamente a A el hash del grafo y del circuito.
4. $A \xrightarrow{o} S : b_j$, A envía abiertamente a s el reto $b_j \in_r \{0, 1\}$
5. S envía abiertamente a A:
   (a) si $b_j = 0$ entonces $S \xrightarrow{o} A : \{\Pi_j(G_t), r_1\}$, S envía abiertamente a A el isomorfismo
   (b) si $b_j = 1$ entonces $S \xrightarrow{o} A : \{\Pi_j(G_t), \Pi_j(HC_t), r_2\}$, S envía abiertamente a A el circuito hamiltoniano en el grafo isomorfo.
6. A verifica
   (a) si $b_j = 1$, que el isomorfismo recibido es correcto y que la función hash h sobre el resultado de Π_j en V_t concatenado con r_1 produce el valor recibido en el paso 3,
   (b) si $b_j = 0$, que la función hash h sobre $\Pi_j(HC_t)||r_2$ produce el valor recibido en el paso 3, y que $\Pi_j(HC_t)$ es un circuito hamiltoniano válido en $\Pi_j(G_t)$,

∗ si $\exists j \in \{1, 2, \ldots, l\}$ de tal manera que la verificación es negativa, entonces S queda aislado.

∗ en otro caso, $A \xrightarrow{s} S$: la información necesaria para tener acceso completo a los recursos protegidos por la red.

Salida: Nodo S está conectado y en línea en la red.

En el segundo paso del algoritmo, se utiliza un esquema criptográfico de compromiso de bits basado en una función hash criptográfica, en el que después de la selección aleatoria del isomorfismo, se envía el resumen del isomorfismo y correspondiente circuito hamiltoniano isomorfo. Para abrir el compromiso, S revela una de esas dos piezas de información permitiendo al receptor recalcular el hash correspondiente para comparar el resultado con el valor hash recibido.

Prueba de Vida

Cada nodo legítimo en línea tiene que confirmar su presencia de forma activa cada cierto intervalo de tiempo de duración T a través de la difusión de una prueba de vida. Durante la fase de retorno del broadcast GRI lanzado por alguno de los nodos, cada nodo añade su propia prueba de vida al paquete retransmitido, de forma que cuando el nodo origen recibe todas las pruebas de vida de todos los nodos legítimos en línea, retransmite la lista y así todos los nodos saben quiénes están conectados. Con el fin de resolver el problema producido por la posible emisión simultánea de pruebas de vida, se introduce un contador de tiempo aleatorio de manera que cada nodo espera un tiempo aleatorio antes de enviar su prueba de vida, y si escucha otra prueba de vida durante ese tiempo, entonces renuncia a su propia difusión y añade su información a la prueba de vida que escuchó.

Algoritmo Prueba-de-Vida

Entrada: En la etapa t, A es un nodo legítimo y conectado.

- A inicializa su $reloj = 0$ justo después de su última prueba de vida
- si $reloj > T$ entonces A envía su prueba de vida mediante broadcast GRI $A \overset{b}{\leftrightarrow}$ red: Su prueba de vida
  - Si en la segunda fase de retorno del GRI A recibe menos de $n/2$ pruebas de vida como respuesta, se detiene la prueba de vida y vuelve a poner a 0 su reloj.
  - de lo contrario: $A \overset{b}{\rightarrow} red$: Pruebas de vida recibidas, A retransmite en la tercera fase del GRI a todo el mundo las pruebas recibidas.

Salida: En la etapa $t + 1$, el nodo A sigue siendo un nodo legítimo y conectado.

Eliminación del Nodo

Cada nodo que no haya demostrado que está conectado enviando su prueba de vida en el periodo establecido por el umbral fijado, es eliminado de la red, y el correspondiente vértice se debe eliminar del grafo y del circuito hamiltoniano. Esta forma de proceder garantiza un crecimiento limitado del grafo que se utiliza en la autenticación, y al mismo

tiempo, permite que los nodos legítimos de la red siempre se correspondan exactamente con los vértices de ese grafo.

Algoritmo de Eliminación

Entrada: En la fase t un nodo v_f es un nodo legítimo y fuera de línea.

- A inicializa su $reloj = 0$
- si $reloj > T$ entonces
  1. $\forall v_i \in V_t$: A comprueba las pruebas de vida de v_i en la cola FIFO que tiene almacenada
  2. A actualiza $V_{t+1} = V_t \setminus \{v_f \in V_t \text{ sin prueba }\}$
  3. A actualiza $E_{t+1} = E_t \setminus \{(v_f, v_j) : v_f \in V_t \text{ sin prueba}, v_j \in N_{G_t(v_f)}\} \cup \{(v_j, v_k) : v_j, v_k \in N_{HC_t(v_f)}\}$
  4. A actualiza $HC_{t+1} = HC_t \setminus \{(v_j, v_f), (v_f, v_k)\} \cup (v_j, v_k) : v_f \in V_t$ sin prueba, $v_j, v_k \in N_{HC_t(v_f)}$
- Si A fue el que inició el broadcast GRI de prueba de vida en el grafo, concluye el borrado de v_f, A añade esta información al último paso del broadcast GRI durante la prueba de vida: $A \xrightarrow{b} network : v_f$ es eliminado.

Salida: En la fase $t + 1$ el nodo v_f ha sido borrado de la red y del grafo.

2.3.6. Análisis de Seguridad

En el sistema SLCM descrito no existe ningún tipo de información revelada en ninguna de las fases que interfiera con la seguridad del esquema global. Por lo tanto, el sistema SLCM puede considerarse totalmente seguro. En esta sección tratamos este tema.

Para la inicialización de la red debe existir un número mínimo de nodos que garantize el secreto de la clave compartida de red, es decir, que garantize la dificultad de HCP en el que está basada dicha clave. Además, estos nodos deben ser legítimos y no tener su seguridad comprometida, es decir, deben ser confiables para el resto de los nodos. Después de la inicialización, la red seguirá operando siempre y cuando el número de nodos no caiga por debajo del umbral establecido, en el que la clave ya no se considera segura.

Esta propuesta asume un entorno ideal donde todos los nodos legítimos son honestos y donde ningún adversario puede poner en peligro a un nodo legítimo de la red para leer su información secreta almacenada. Esa hipótesis de trabajo es adecuada como modelo básico con el fin de decidir en qué circunstancias el esquema SLCM diseñado es aplicable a las MANETs. Por ejemplo, una posible adaptación de la propuesta con el fin de evitar esa hipótesis podría ser la consideración de un esquema umbral para cada paso del esquema, de forma que cada prueba de vida, inserción, eliminación o control de acceso debe ser realizada por un grupo de nodos o bien por todos los nodos cada vez, en lugar de realizarlo un único nodo. De esta forma, un único nodo deshonesto no podría alterar el correcto funcionamiento de la red.

Otro de los requisitos del esquema es la necesidad de establecimiento de un canal seguro tanto para la inicialización de la red como para el procedimiento de inserción, donde se supone que existe confianza entre pares de nodos. Sin embargo, en la práctica este requisito se puede conseguir fácilmente en muchos casos gracias al hecho de que un gran número de dispositivos inalámbricos móviles se pueden comunicar entre sí a través de la tecnología Bluetooth, que sin embargo no es válida para las comunicaciones en general debido a su corto alcance de retransmisión.

Con respecto a posibles ataques, debido a la falta de una estructura centralizada, es natural pensar en posibles ataques DoS contra la aplicación de chat como su principal objetivo. Con el fin de proteger el esquema frente a esta amenaza se debe asegurar que los mensajes de chat, aunque públicamente legibles, puedan ser enviados solamente por miembros de la red legítimos y en línea. Otro aspecto importante relacionado con el uso de la aplicación de chat es la sincronización necesaria de los nodos en línea, por lo que se hace necesario un reloj de red común a todos los nodos. Este requisito se ha implementado durante las simulaciones gracias al broadcast GRI y a la sincronización del reloj durante las pruebas de vida periódicas. Dicho método no es 100 % preciso, pero tiene márgenes de error aceptables.

Las MANETs son en general vulnerables frente a diversas amenazas como al ataque de suplantación o el $MitM$. Este tipo de ataques son difíciles de prevenir en entornos donde la estructura y composición de la red son dinámicos, y no se puede asumir la presencia

de estructuras centralizadas. Sin embargo, nuestra propuesta es resistente a los ataques de suplantación de identidad porque el control de acceso se realiza a través de una ZKP, lo que hace inútil la lectura de cualquier información publicada a través de la aplicación de chat o enviada en abierto durante un control de acceso. Por otro lado, el objetivo de un ataque MitM puede ser cambiar un mensaje enviado y obtener alguna información útil de alguno de los nodos intermedios. Una vez más, el uso de ZKPs en el protocolo implica que la lectura de cualquier paquete enviado no revela ninguna información útil sobre el secreto, por lo que no es posible cambiar el mensaje ya que sólo los nodos legítimos, cuyo acceso a la red haya sido autorizado, pueden utilizar y entender la información enviada a través de la aplicación de chat.

Otro ataque activo que podría ser especialmente peligroso en MANETs es el llamado ataque Sybil. Este ataque sucede cuando un nodo intenta conseguir y usar varias identidades. El caso más extremo de este tipo de ataques es el establecimiento de una autoridad centralizada falsa que puede confirmar la identidad de miembros de la red. Sin embargo, este ataque en particular no es posible contra nuestro esquema, debido a su naturaleza distribuida. En el sistema SLCM, la responsabilidad del control general que proporcionaría un ataque Sybil exitoso está compartido entre todos los nodos en línea de la red. Si un nodo autenticador detecta que un nodo está tratando de obtener acceso a la red mediante el uso de un identificador que ya está siendo utilizado, denegará la petición durante el control de acceso y el nodo correspondiente será aislado. Lo mismo sucede cuando cualquier nodo en línea detecta que un nodo autenticado está tratando de insertar un nuevo nodo en la red con una nueva identificación, y otro nodo ya tiene asignado ese ID de vértice. Una vez más, dicha inserción debe ser denegada y el nodo solicitante correspondiente debe ser aislado. De todos modos, si un atacante Sybil entra en la red, cualquiera de sus vecinos lo detectará tan pronto envíe alguna prueba de vida usando alguno de sus diferentes identificadores.

2.3.7. Evaluación del Rendimiento

En esta sección se analiza la eficiencia de la propuesta, tanto desde el punto de vista del consumo de energía como del punto de vista de la complejidad computacional. Para el análisis del rendimiento de la propuesta se utilizó el simulador de redes NS-2 con el protocolo

de enrutamiento DSR. Se crearon scripts Tcl basados en NS-2 con el fin de producir archivos de traza de salida que se han utilizado tanto para hacer el procesamiento de datos como para visualizar la simulación. Para comprobar el funcionamiento de estas simulaciones se utilizó la herramienta de visualización de red *NAM* y el analizador de ficheros de traza de NS-2, *Tracegraph*. Para la simulación de movilidad se utilizó el programa *setdest* con el fin de generar archivos con el patrón de movimiento basados en un algoritmo de punto de referencia al azar. Se destacan a continuación algunas de las principales conclusiones extraídas de dichas simulaciones del esquema SLCM.

El consumo de energía es el resultado de las transmisiones de datos y actividades del procesador debido sobre todo a las tareas de autenticación en la red. En el esquema SLCM hay dos fases donde la carga computacional es más importante: el control de acceso basado en ZKP y el control periódico de la cola FIFO almacenada. Una reducción en el número de iteraciones de la ZKP tiene un efecto directo sobre el número total de mensajes intercambiados en las inserciones, pero hay que mantener un equilibrio entre la robustez del protocolo y su rendimiento.

Las pruebas de vida periódicas implican aproximadamente el 90 % del número total de mensajes intercambiados. Sin embargo, hemos encontrado que estas pruebas de vida obligatorias implican una técnica de incentivos para estimular la cooperación ya que los nodos que son retransmisores de inserciones o eliminaciones en la red, o autenticadores en los controles de acceso están exentos de la obligación de difundir sus pruebas de vida.

Con el fin de reducir los costos de comunicación de datos del protocolo, un aumento en el período umbral T podría ser una opción, pero de nuevo se debe mantener un equilibrio aceptable. De acuerdo con nuestros experimentos, T debe depender directamente del tiempo medio que los nodos están fuera de línea y del número de nodos legítimos y/o en línea a fin de evitar una posible sobrecarga en el ancho de banda de las redes de gran tamaño. De los experimentos también se deduce que, como era de esperar, el número de paquetes generados en la red crece con el número de nodos autenticados en la red. Además, aunque el número total de paquetes en la red crezca, el número de paquetes en cada área de la red se mantiene casi constante, viéndose afectado sólo si la densidad en la zona es mayor.

La energía que un nodo necesita no se ve afectada por el crecimiento del número

de nodos de la red, pero si se ve afectada por su densidad. Sin embargo, el tamaño de almacenamiento que cada nodo necesita sí aumenta a medida que el tamaño de la red crece. Este aspecto, junto con el problema de encaminamiento son las principales razones por las que es necesario establecer un límite superior para el tamaño de la MANET.

En particular, en principio se desarrollaron varios ejemplos sencillos de simulaciones NS-2 con unos pocos nodos. Estas simulaciones incluyen los archivos de escenario que describen los patrones de movimiento de los nodos y los archivos que describen el tráfico de comunicación en la red que posteriormente se utilizan para producir los archivos de traza que se analizaron para medir los distintos parámetros.

Los archivos de traza se utilizaron para visualizar la simulación utilizando NAM, mientras que los valores de medición se utilizan como datos para crear gráficas usando Tracegraph. Un ejemplo del grafo final y el circuito hamiltoniano asociado a la red se muestra en la Fig. 2.5 donde el color verde se utiliza para indicar el circuito hamiltoniano en el grafo original y el azul en el grafo isomorfo.

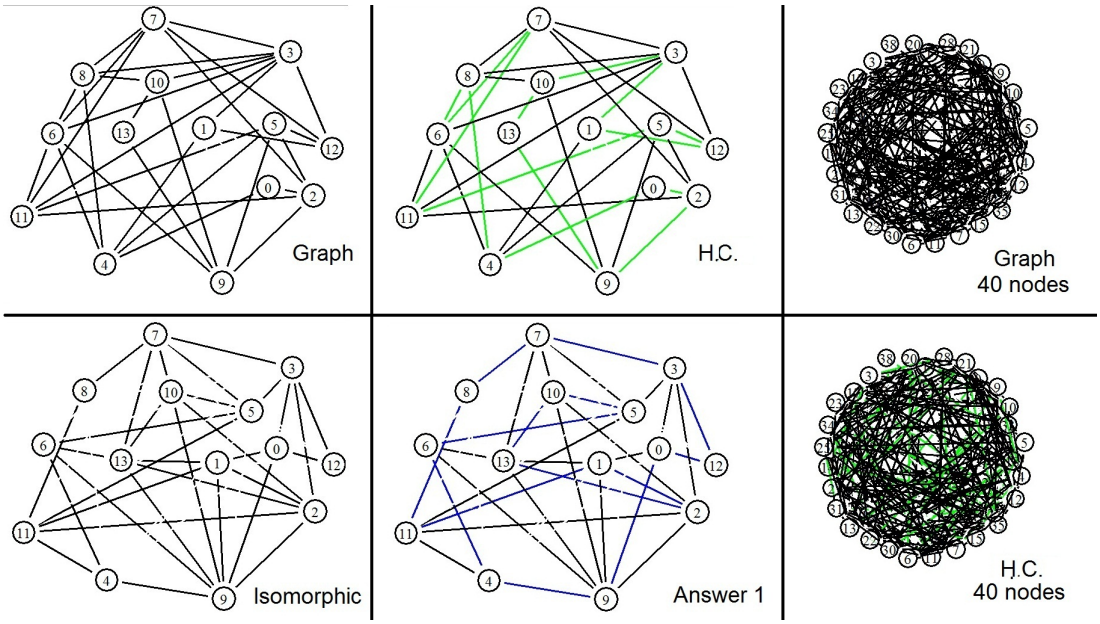


Figura 2.5: Ejemplo de ZKP basada en el HCP

Como se ha mencionado, el HCP es un problema NP-completo. De hecho, buscar un HC con algoritmos exactos es computacionalmente intratable, y la única aproximación

práctica es a través de algoritmos heurísticos. Incluso la mayoría de los algoritmos heurísticos son inútiles para diferentes tipos de grafos aleatorios con más de 200 nodos [66]. Sin embargo, dado que el protocolo no requiere la solución del problema, sino la construcción de un grafo a partir de una solución escogida, la dificultad del HCP no es una desventaja en contra de la eficiencia del sistema, sino una ventaja a favor de la seguridad que aporta al sistema.

Realizamos diferentes simulaciones para comprobar los efectos de usar diferentes parámetros mediante la variación de densidad y topología de la red. Se usó un número de nodos desde 10 hasta 100, en un área entre 400 y 1000 m^2, y con un período de simulación desde 60 hasta 200 segundos. También se han variado las probabilidades de inserción y eliminación entre el 5 % y el 25 % y se modificó la tasa de movilidad y el alcance de la antena de los nodos de 2 a 15 m/s, y de 100 a 250 metros, respectivamente. Este último rango cubre las diferentes frecuencias de acceso a la red inalámbrica.

Para la simulación se distinguieron siete estados diferentes en los que cada nodo se puede encontrar en la red (ver Fig. 2.6).

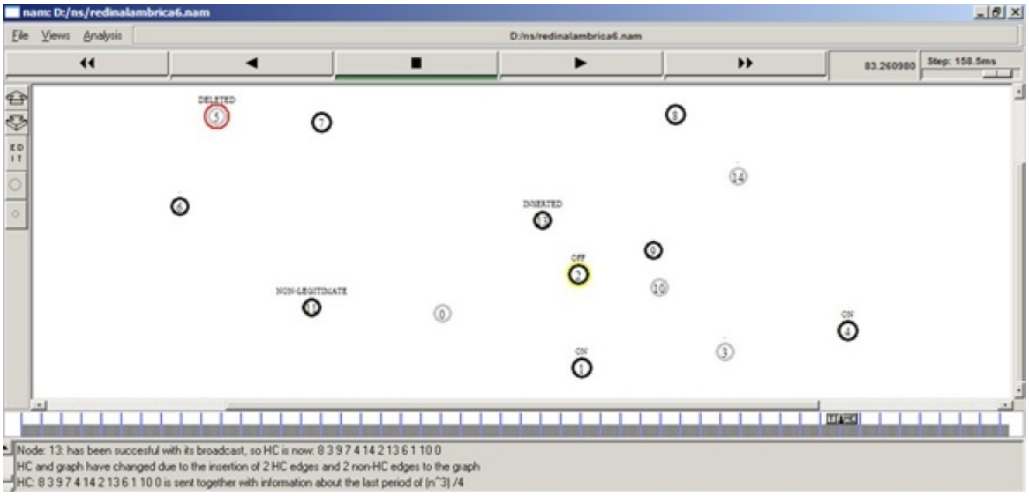


Figura 2.6: Ejemplo de Simulación de Red con NS-2

Tiempo	Evento	H.C.
0.1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 son legítimos	8,3,9,7,4,2, 6,5,1,10,0
1.29	Inserción del Nodo 14, Nodo 4 envía broadcast	8,3,9,7,4,14, 2,6,5,1,10,0
1.30	Nodos 3, 1, 0 no responden a la prueba de vida	
11.65	Nodo 1 se apaga	
13.97	Prueba de vida iniciada por Nodo 3	
14.27	Nodos 1, 2 no responden a la prueba de vida	
17.27	Prueba de vida iniciada por Nodo 2	
17.57	Nodos 1, 5 no responden a la prueba de vida	
21.71	Nodo 5 se apaga	
31.40	Nodo 1 se enciende y Nodo 2 le realiza la ZKP	
31.46	Nodo 4 se apaga	
32.51	Prueba de vida iniciada por Nodo 1	
32.78	Nodos 4, 5, 6 no responden a la prueba de vida	
38.51	Prueba de vida iniciada por Nodo 6	
38.79	Nodos 4, 5 no responden a la prueba de vida	
41.46	Nodo 1 se apaga	
53.25	Nodo 1 se enciende y el Nodo 0 le realiza la ZKP	
59.61	Prueba de vida iniciada por Nodo 6	
59.99	Nodos 4, 5 no responden a la prueba de vida	
64.26	Nodo 5 es eliminado de la red	8,3,9,7,4,14, 2,6,1,10,0
64.71	Nodo 2 se apaga	
72.58	Nodo 4 se enciende y el Nodo 0 le realiza la ZKP	
75.41	Inserción del Nodo 13, Nodo 14 envía broadcast	8,3,9,7,4,14, 2,13,6,1,10,0

Tabla 2.1: Ejemplo de Traza

- *Autenticado* (Authenticated): Nodo legítimo conectado y que ha superado con éxito un control de acceso.
- *No legítimo* (Non-legitimate): Nodo que no pertenece a la red, y por tanto sería candidato para entrar en la red.
- *Encendido* (On): Cuando un nodo se enciende y debe solicitar a otro nodo legítimo, conectado y autenticado que le dé acceso a la red. Este nodo, que está encendido pero aún no está autenticado, se entiende apagado a efectos de las comunicaciones en la

red.

- *Apagado* (Off): Por este estado pasa todo aquel nodo que pertenece a la red, pero está temporalmente fuera de línea. Dicho nodo puede encenderse y conectarse a la red tras pasar un control de acceso en el que demuestra que conoce el secreto de la red, o bien se puede quedar apagado hasta que expire su umbral de tiempo de vida, en cuyo caso el resto de nodos lo eliminarán de la red.

- *Eliminado* (Deleted): Cuando el nodo está fuera de línea durante un tiempo superior al umbral T, el nodo es eliminado tanto del HC como del grafo de red por parte de todos los nodos de la red, de forma que queda eliminado a todos los efectos de la red.

- *Fuera de servicio* (Out-of-Service): Un nodo que es legítimo y está en línea, pero no responde a una prueba de vida iniciada por otro nodo porque está fuera de la cobertura de la red, tendrá que demostrar mediante un control de acceso que pertenece a la red cuando se encuentre con otro nodo de la red que sí respondió a la última prueba de vida lanzada.

- *Insertado* (Inserted): Un nodo no legítimo que recibe suficiente información de la red de un nodo legítimo después de un procedimiento de inserción, cambia su estado de no-legítimo a *Insertado*.

Para estudiar el comportamiento del esquema SLCM propuesto, las simulaciones se llevaron a cabo utilizando la misma densidad de nodos con diferente número de nodos en diferentes áreas y durante un tiempo suficiente para estudiar los efectos de las pruebas de vida. De dichas pruebas se recogieron datos sobre el número de conexiones y el número de paquetes generados, transmitidos y perdidos, que se muestran en la Fig. 2.7.

A continuación se analizan diferentes aspectos de los resultados experimentales, que muestran la calidad y seguridad del esquema propuesto, teniendo en cuenta particularmente la relación con el número de nodos. La Fig. 2.7 muestra que, de acuerdo con las simulaciones de la propuesta, tanto el número de conexiones como el número de paquetes generados aumentan linealmente con el número de nodos. Esto sucede cuando la densidad de la red se mantiene al aumentar el número de nodos. La imagen también muestra que el

número de paquetes transmitidos que se pierden también aumenta con el número de nodos, pero de una manera más contenida que en el caso de los paquetes generados. Esto ocurre cuando aumenta el número de nodos, ya que en ese caso también aumenta el número de conexiones. Para que las simulaciones mantengan una densidad constante, se aumenta el tamaño del plano al aumentar el número de nodos. De este modo, la interferencia entre los nodos se mantiene siempre en cotas similares. Por tanto, los resultados obtenidos en relación con la pérdida de paquetes pueden ser considerados de forma positiva.

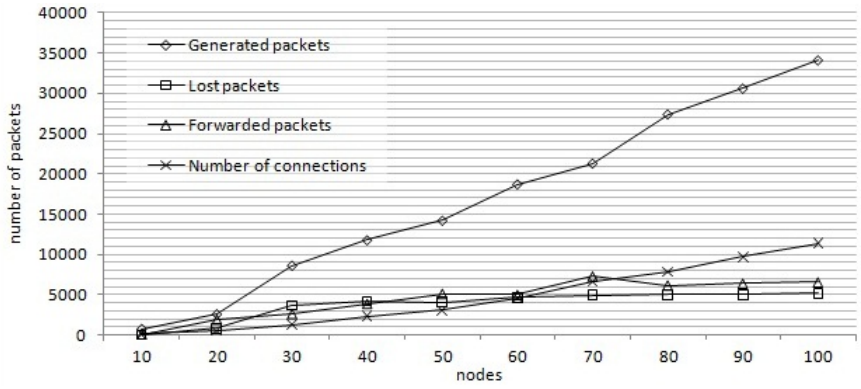


Figura 2.7: Número de Paquetes

La Fig. 2.8 refleja el promedio de energía y la potencia máxima consumida por cada nodo. Estos parámetros se calculan a partir del tiempo de procesamiento de paquetes de cada nodo. Este cuadro permite ver que el tiempo de procesamiento máximo se incrementa de la misma forma que el número de nodos, aunque hay algunas excepciones. Esto se debe a que con una mayor densidad de nodos que inicien pruebas de vida, existe más necesidad de cómputo en la red. La imagen muestra que el tiempo promedio de procesamiento es bastante bajo y no sigue un patrón que pueda ser utilizado para relacionar con el número de nodos. De todos modos, podemos concluir que en promedio, la energía consumida por los nodos no aumenta demasiado cuando la red crece.

La Fig. 2.9 muestra tanto el retardo medio de la señal entre los nodos como el mayor retardo que se produce en la simulación para diferentes números de nodos. En ambos casos vemos un gran crecimiento con 30 nodos y luego un ligero aumento. El máximo retardo que se produce con más de 30 nodos es casi constante en 7 segundos, mientras

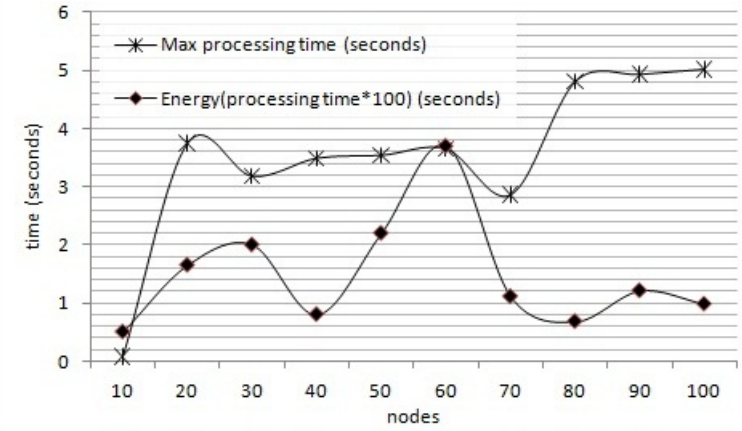


Figura 2.8: Tiempo de Procesamiento

que el tiempo medio de retardo aumenta hasta 30 nodos y luego fluctúa. Estos resultados muestran un buen comportamiento general de la propuesta con respecto al retardo de los mensajes producidos por el crecimiento de la red.

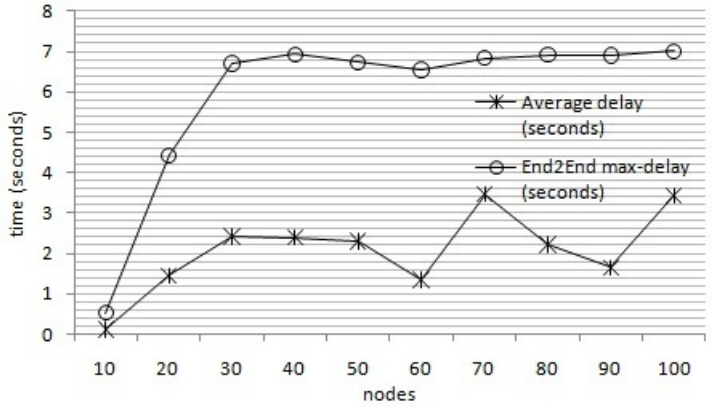


Figura 2.9: Retardo entre Nodos

La Fig. 2.10 muestra los requerimientos máximos de almacenamiento en bits para cada nodo. De hecho, ya que la propuesta no requiere casi ningún almacenamiento, el crecimiento mostrado se debe a que cada nodo puede tener que almacenar las claves públicas y otros datos de los restantes nodos de la red. Además, cada nodo puede almacenar un número de certificados firmados por y para los demás nodos. Hemos comparado la necesidad de almacenamiento utilizando claves de 1024 bits usadas habitualmente en RSA, y claves de

160 bits usadas habitualmente en criptografía de curva elíptica, que se considera que tienen una seguridad criptográfica equivalente.

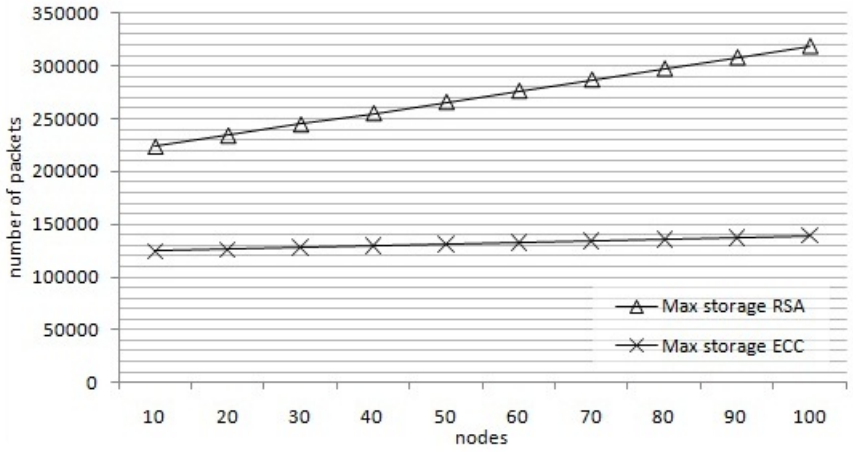


Figura 2.10: Requerimientos de Almacenamiento Máximos

Por tanto, algunas conclusiones generales que se pueden deducir de las simulaciones y de la correspondiente evolución del rendimiento del sistema SLCM son:

- El protocolo SLCM es escalable para cualquier tipo de redes con distintos niveles de cambio en su topología.
- La densidad de nodos es un factor clave para el tiempo medio de inserciones, pero no es tan grande como se podría suponer a priori.
- Para la elección correcta del parámetro T se debe tener en cuenta el tiempo medio que los nodos legítimos se encuentran fuera de línea, el número de nodos de la red, el ancho de banda de las conexiones inalámbricas y la capacidad de cálculo y almacenamiento de los nodos.
- Uno de los aspectos positivos de la propuesta es que los requisitos de hardware de los dispositivos son muy bajos.

2.4. Gestión de Claves Públicas

La criptografía de clave pública constituye una solución criptográfica para el difícil problema de la gestión de claves secretas a la vez que permite el uso de todo un completo abanico de aplicaciones criptográficas tales como la difundida y práctica firma digital. Sin embargo el uso de la criptografía de clave pública es imposible sin antes resolver la cuestión de la certificación de claves públicas.

En esta sección analizamos precisamente el problema de la gestión y certificación de claves públicas en MANETs. Además de analizar detenidamente el problema y algunas de las soluciones encontradas en la bibliografía, proponemos esquemas mejorados de certificación de claves públicas auto-organizados que permiten el uso práctico de criptografía de clave pública en MANETs. Todo este estudio se completa con simulaciones NS-2 de las que se extraen numerosas conclusiones.

2.4.1. Grafos Certificados

El esquema propuesto para la certificación de claves públicas en MANETs se basa en el planteamiento descrito en [54] en el que se sustituye la autoridad de certificación centralizada habitual en las redes cableadas e inalámbricas con infraestructura, por un escenario auto-organizado en el que la certificación se realiza mediante cadenas de certificados emitidos y firmados por los propios nodos de la red. Dicho esquema utiliza la información almacenada por cada nodo y el hecho de que cada nodo confía en sus nodos vecinos. Esta última característica constituye la esencia de una infraestructura auto-organizada y puede utilizarse, y de hecho es utilizada, para generar confianza a partir de la confianza existente entre nodos vecinos.

En el esquema basado en grafos certificados cada nodo tiene una clave pública, una clave privada, y un repositorio incluyendo la lista de certificados de todos los nodos en los que confía (out-bound) y la lista de certificados de todos los nodos que confían en él (in-bound). Así, cada certificado es siempre almacenado dos veces, por su emisor firmante y por su dueño.

Cuando un nodo desea comprobar la validez de la clave pública de otro nodo, tiene

que encontrar una cadena de certificados desde él hasta el otro nodo, en el grafo que resulta de combinar su repositorio con el repositorio del otro nodo.

Las claves públicas y los certificados se representan como un grafo dirigido $G = (V, E)$, conocido como grafo certificado en el que cada vértice u representa simultáneamente a una clave pública y a su dueño, y cada arco (u, w) simboliza un certificado de la clave pública w firmado con la clave privada correspondiente a u. Una cadena de certificados es un camino dirigido en dicho grafo.

Nótese que la velocidad de creación de un grafo certificado que contenga suficientes conexiones como para permitir la comunicación entre dos nodos cualesquiera de una red depende de la motivación de los usuarios para distribuir certificados y de su movilidad ya que como los nodos comparten sus repositorios con los nodos cercanos, cuanta más movilidad tengan los nodos, pasarán cerca de más nodos y por tanto intercambiarán repositorios con más nodos.

La autenticación de la clave pública de un nodo v por otro nodo u se realiza buscando en la unión de sus repositorios una cadena de certificados correctos y no caducados entre u y v en el grafo resultante de la unión de ambos repositorios ya que:

1. El primer certificado de la cadena puede ser comprobado directamente por u ya que fue firmado por él mismo.
2. Cada uno de los demás certificados de la cadena puede ser comprobado usando la clave pública del certificado previo de la cadena.
3. El último certificado es el de la clave pública del usuario destino v.

La elección de los certificados que almacena cada nodo en su repositorio debe realizarse cuidadosamente para satisfacer a la vez los dos requerimientos: limitación de almacenamiento de los nodos, y utilidad del repositorio para encontrar cadenas de certificados para el mayor número de nodos posibles.

El algoritmo más sencillo que se ha propuesto hasta el momento para la construcción del repositorio de cada nodo es el conocido como algoritmo de máximo grado, cuyo nombre proviene de que el criterio que se sigue para la elección de los certificados es el grado de los vértices del grafo certificado [54].

Existe otro algoritmo más sofisticado propuesto por los mismos autores, llamado Shortcut Hunter Algorithm, en el que se eligen los certificados tales que cuando se borran del grafo, la longitud del camino mínimo entre los dos nodos conectados con ese certificado se incrementa en más de dos unidades. El análisis de ambos algoritmos muestra que cualquier usuario podrá encontrar en el grafo resultante tras un número de iteraciones suficiente al menos una cadena de certificados entre cualquier par de nodos con una alta probabilidad.

En nuestras propuestas descritas a continuación se utiliza un subgrafo G_u del grafo certificado G conteniendo el repositorio con los certificados verificados por el nodo u, y un grafo independiente denotado como ${G_u}^r$ conteniendo los certificados recopilados por u. Antes de que un certificado caduque, su emisor debería distribuir una versión actualizada, pero puede ocurrir que no lo haga. En ese caso, el grafo ${G_u}^r$ (llamado repositorio recopilado) resulta muy útil ya que proporciona una buena estimación sobre la parte del grafo certificado no incluida en G_u (llamado repositorio verificado).

A continuación se describen brevemente las 4 fases necesarias para la gestión de los repositorios de claves públicas:

1. Inicialización de Claves

   Creación de un par de claves pública y privada por cada usuario.

2. Inicialización de Grafo Certificado

   Mediante la emisión de certificados a los nodos de confianza, se logra definir el grafo certificado inicial que en realidad nadie conoce en su totalidad.

   Puede haber muchas razones por las que u crea que K_v pertenece a v, como por ejemplo que hayan intercambiado sus claves a través de un canal seguro. La naturaleza dinámica de las MANETs hace que los usuarios cada vez obtengan más información sobre los demás, distribuyan más certificados y evalúen mejor su confianza en los certificados que distribuyen.

   La distribución y revocación de claves públicas son las únicas operaciones conscientes e intencionadas que realizan los nodos, ya que el resto, incluyendo la validación y el intercambio de certificados, se realizan automáticamente.

3. Actualización de Repositorio Recopilado

   El intercambio de certificados con los nodos vecinos para la creación del repositorio recopilado es un procedimiento de bajo costo que permite a los nodos compartir sus repositorios. Se describe como sigue:

   (a) Cada nodo u retransmite resúmenes de los certificados almacenados en sus repositorios G_u y ${G_u}^r$ a sus vecinos. Los vecinos que reciban este mensaje responden a su vez con los valores hash de los certificados de sus dos repositorios.

   (b) Cada nodo compara lo recibido con lo que tiene y solicita a sus vecinos sólo los certificados que no tiene.

   (c) Si la memoria local del nodo se queda pequeña, borra los certificados caducados del repositorio recopilado, por orden de fecha.

   (d) Así, los nodos van acumulando certificados en su repositorio recopilado ${G_u}^r$ de forma que tras un corto periodo de tiempo, estos repositorios contienen casi todo el grafo certificado G. Después, los nodos sólo tienen que intercambiarse los nuevos certificados generados.

4. Actualización de Repositorio Verificado

   El repositorio verificado G_u del nodo u se actualiza aplicando sobre su repositorio recopilado ${G_u}^r$ un algoritmo para escoger los certificados más adecuados para incorporar a G_u. Tras la ejecución de dicho algoritmo, u debe comprobar, comunicándose con sus emisores, la validez de cada certificado de ${G_u}^r$ a incorporar en G_u

2.4.2. Algoritmo de Máximo Grado con Dos Cadenas

A continuación proponemos un algoritmo para la selección de los certificados de ${G_u}^r$ a incorporar en G_u durante la construcción del repositorio verificado. El objetivo es almacenar la mínima información necesaria que permita a los nodos comprobar si otros nodos son fiables y de esta manera poder establecer comunicación con el máximo número de ellos. Para lograrlo, en la versión concreta descrita en esta sección los nodos almacenan exactamente dos cadenas conteniendo los nodos con mayores grados entre los que pueda

escoger en cada momento según un conjunto de restricciones. Además en nuestro esquema simplificamos el modelo considerando todas las comunicaciones y confianzas bidireccionales de forma que el grafo certificado correspondiente es no dirigido.

Cada nodo u tendrá en su repositorio verificado G_u a todos los nodos en los que confía que están a distancia 1 en el grafo certificado.

Tal como se describió en la subsección anterior, tras las fases 1 y 2 de inicialización, los nodos inician su contador de tiempo para la actualización del repositorio recopilado a un número aleatorio menor de x unidades de tiempo donde el valor x concreto dependerá del tamaño de la red. En la simulación usamos $x = 3$. Cuando llegue ese momento el nodo ejecuta la fase 3 actualizando su repositorio recopilado comunicándose con todos los nodos que estén a su alcance en ese momento.

Durante la fase 4 de actualización del repositorio verificado se utiliza el algoritmo descrito a continuación, donde cadena representa cada una de las cadenas del repositorio verificado del nodo ejecutor y B.cadena cada una de las del nodo con el que se ha autenticado.

Algoritmo Máximo Grado con 2 Cadenas

```
00: ...//Autenticación con nodos, intercambiando repositorios verificados
01: //Actualización de grados de los nodos en los repositorios.
02: para (i = 0; i < tam(cadena); i++)
03:    para (j = 0; j < tam(B.cadena); j++)
04:       si (cadena(i) == B.cadena(j))
05:          si (grado(sgte(cadena, i)) < grado(sgte(B.cadena, j))
06:             actualizaCadena(i, B.cadena, j);
07:          fin si
08:       fin si
09:    fin para
10: fin para
```

El repositorio verificado G_u es inicializado por u primero incluyendo los certificados emitidos por los nodos que han firmado directamente su clave pública, que son los mismos a los que dicho nodo les ha firmado la clave pública con su clave privada. A continuación,

el nodo u intercambia con dichos nodos sus repositorios verificados para determinar su repositorio recopilado inicial ${G_u}^r$ mediante la unión de todos ellos.

Tras cada autenticación los nodos intentan actualizar sus repositorios verificados a partir de sus repositorios recopilados para intentar garantizar la confianza con el máximo número de nodos de la red mediante el almacenamiento del menor número posible de claves en sus repositorios verificados.

Una vez seleccionados los certificados a incorporar en los repositorios verificados, los nodos antes de nada deben comprobar la confianza en dichos certificados.

En el algoritmo propuesto se utilizan dos cadenas para confirmar G_u a partir de ${G_u}^r$

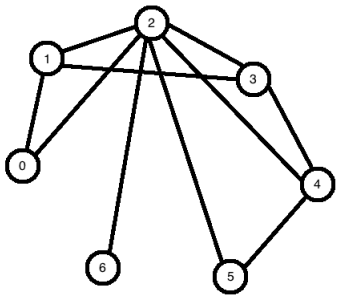
- **Cadena1** $C1_u$**:** En esta lista se almacena una cadena que comienza en uno de los nodos de mayor grado de los contenidos en ${G_u}^r$, y no repite vértices.

- **Cadena2** $C2_u$**:** En esta lista se almacena una cadena sin repetir vértices, que comienza por uno de los nodos con segundo mayor grado de los contenidos en G_u, o directamente en el que haya, si sólo hay uno.

En la Fig. 2.11 y tabla 2.2 se puede ver un ejemplo de cómo sería el repositorio G_u y las posibles cadenas $C1_u$ y $C2_u$ que se podrían formar dependiendo de con quien intercambie repositorios en un momento dado.

En la comunicación del repositorio recopilado ${G_u}^r$, si un nodo u confía en el nodo v, durante el proceso de unión de repositorios comprueba si el nodo v está en alguna de sus cadenas $C1_u$ ó $C2_u$. En caso de que así sea, comprueba si tiene en alguno de sus repositorios el primer elemento de una de las cadenas de v, en caso de no tenerlo, lo añade a su cadena correspondiente, y aplica el mismo procedimiento de forma recursiva.

En caso de que el nodo v no se corresponda con ninguno de los nodos vértice de ninguna de las dos cadenas de u, se comprueba si el primer nodo de alguna de las dos cadenas del nodo v está en alguna de las cadenas de u. Si es así se añade el nodo v a la cadena correspondiente de u.



Matriz de adyacencia

	0	1	2	3	4	5	6
0	-	1	1	0	0	0	0
1		-	1	1	0	0	0
2			-	1	1	1	1
3				-	1	0	0
4					-	1	0
5						-	0
6							-

Vector de grados

nodo	0	1	2	3	4	5	6
grado	2	3	6	3	3	2	1

Figura 2.11: Vector y Repositorio

2.4.3. Simulación del Algoritmo de Máximo Grado con Dos Cadenas

Para la implementación de la simulación NS-2 del algoritmo de máximo grado con dos cadenas, el código se organizó en los siguientes archivos:

- *Redinalambrica7.tcl:* Fichero principal que se encarga de capturar los ficheros de movimiento y vida de red generados y crear la simulación de la red y su fichero de traza.

- *Gengestionclaves.tcl:* Fichero en el que se desarrolla toda la vida de la red: inserciones, apagados, encendidos, conexiones, actualizaciones de repositorios, etc. Este fichero crea una serie de ficheros complementarios con información referente a la red creada, que el generador de grafos utiliza para crear el grafo certificado asociado a dicha red. Dichos ficheros son:

- *grafogestionclaves.txt:* Almacena el número de elementos en la primera línea y luego guarda para cada nodo una línea con su repositorio verificado inicial, una línea con su cadena C1 y otra con su cadena C2. Sirve para crear el grafo certificado de la red, así como para pintar las aristas que se encuentran en los repositorios en las sucesivas conexiones de la red.

- *conexionesgestionclaves.txt:* Guarda las conexiones de la red, conteniendo una línea

w	$C1_u$	$C2_u$
0	0 2 1 3 4 5	0 1 3 4 5 2 6
	0 2 3 1 0	
	0 2 3 4 5	
	0 2 4 3 1	
1	1 2 3 4 5	1 3 4 5 2 6
	1 2 4 3	
2	2 1 3 4 5	2 1 3 4 5
	2 3 1 0	2 3 1 0
	2 4 3 1 0	2 3 4 5
		2 4 3 1 0
3	3 2 1 0	3 1 2 4 5
	3 2 4 5	3 4 5 2 1 0
4	4 2 1 3	4 3 1 0 2 6
	4 2 3 1 0	
5	5 2 1 3 4	5 4 3 1 2 6
	5 2 3 1 0	
	5 2 3 4	
	5 2 4 3 1 0	
6	6 2 1 3 4 5	6 2 1 3 4 5
	6 2 3 1 0	6 2 3 1 0
	6 2 3 4 5	6 2 3 4 5
	6 2 4 3 1 0	6 2 4 3 1 0

Tabla 2.2: Posibles Cadenas a Almacenar

por conexión con: tiempo de la conexión, primero y segundo nodo de la conexión. Sirve para saber en qué momento se conectan dos nodos. Se pintan sus aristas en ese momento.

- *noconexionesgestionclaves.txt:* Recopila los intentos fallidos de conexión con el mismo formato del fichero de conexiones. Sirve para saber cuándo intentan conectarse dos nodos y no pueden. Pintamos sus aristas para dejar de manifiesto que no tienen ningún nodo en común.

- *nuevosnodosred.txt:* Almacena los nodos insertados en la red. En cada línea se guarda un nodo insertado con el formato, momento de la inserción, nodo insertado, lista de repositorio verificado de dicho nodo. Sirve para no dejar que se pinten las aristas de los

nodos que aún no han sido insertados en la red. Una vez se supere el tiempo indicado de la inserción, las aristas de este nodo podrán ser pintadas.

- *nuevosenrepositorio.txt:* Recolecta las aristas que han insertado los nodos en el transcurso de la vida de la red. El formato utilizado en este fichero es:
  - primera línea: momentos de las inserciones
  - segunda línea: nodos que insertan en sus repositorios
  - tercera línea: nodos origen de las aristas
  - cuarta línea: nodos destino de las aristas

  Sirve para impedir que se pinten las aristas que aún no han sido integradas en los repositorios de los nodos. Una vez se supere el tiempo en el que el nodo añade la arista, esta podrá ser pintada.

- *Gengrafogestionclaves.tcl:* Fichero que captura lo que ha sucedido en la red (generado por gengestionclaves.tcl). Muestra el grafo certificado de la red en cada instante y colorea las conexiones que se han realizado entre los diferentes nodos y las nuevas aristas introducidas en los repositorios de los nodos o con la inserción de un nuevo nodo en la red.

A continuación, de modo esquemático se muestran en las Fig. 2.12 y 2.13 en forma de diagramas de flujo respectivamente, un esquema general correspondiente a la implementación de la vida de un nodo en un esquema simulado de gestión de claves, y la actualización de repositorios.

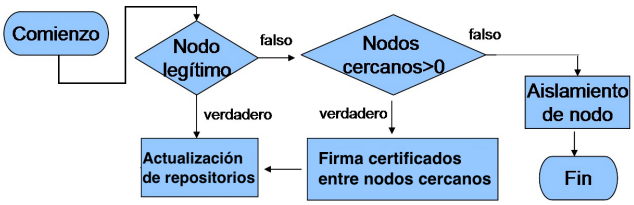


Figura 2.12: Vida de un Nodo en Gestión de Repositorios

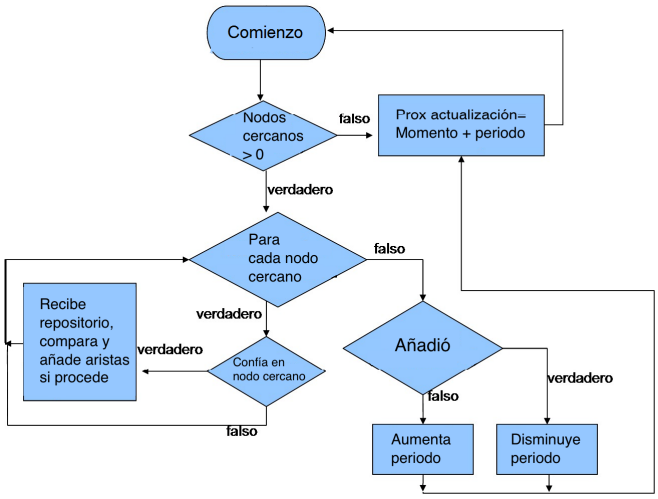


Figura 2.13: Actualización de Repositorios de un Nodo

En la Tabla 2.3 y Fig. 2.14 que se muestran a continuación, se puede ver un ejemplo de traza y correspondientes transformaciones del grafo certificado y de las conexiones de la red.

Momento	Evento
0.1	Nodos 0,1,2,3,4,5,6,7,8 están encendidos
0.1	Formadas dos subredes, 0,1,3 y 2,4,5,6,7,8
1.1	Broadcast GRI para conocimiento de la red
2.85	3 actualiza repositorios con 1 y 0. No añade al repositorio. Aumenta periodo
2.95	1 actualiza repositorios con 0 y 3. No añade al repositorio. Aumenta periodo
3.14	Nodo 7 se apaga
3.27	0 actualiza repositorios con 1, 3, 8. No confía en 8. No añade aristas. Aumenta periodo
5.26	4 actualiza repositorios con 0, 5, 8. No confía en 0. Añade arista 5-7. Disminuye periodo
5.28	8 actualiza repositorios con 0, 4, 5. No confía en 0. No añade aristas. Aumenta periodo
5.31	2 no tiene nodos cercanos para actualizar repositorios. No añade aristas. Suma periodo
5.36	5 actualiza repositorios con 4, 6, 8. No añade aristas. Aumenta periodo
5.85	6 actualiza repositorios con 5. Añade arista 5-8. Disminuye periodo
7.85	3 actualiza repositorios con 1. No añade aristas. Aumenta periodo
7.95	1 actualiza repositorios con 3. No añade aristas. Aumenta periodo
8.26	4 actualiza repositorios con 8. No añade aristas. Aumenta periodo
8.27	0 actualiza repositorios con 8. No confía en 8. No añade aristas. Aumenta periodo
8.85	6 actualiza repositorios con 5. No añade aristas. Aumenta periodo
10.28	8 actualiza repositorios con 0, 4. No confía en 0. No añade aristas. Aumenta periodo
10.31	2 actualiza repositorios con 5. No añade aristas. Aumenta periodo
11.78	Conexión entre 5 y 4. Se comprueba la confianza. Comienza el intercambio de datos.
	...
16.51	12 Nuevo nodo en la red, cambia firmas con 0,1,3,4,6,8.
	Broadcast GRI para conocimiento de nueva red.
	...
43.90	10 Nuevo nodo en la red, cambia firmas con 0,3,4,5,8,12.
	Broadcast GRI para conocimiento de nueva red.
	...
81.87	9 Nuevo nodo en la red, cambia firmas con 0,2,3,4,5,8,10.
	Broadcast GRI para conocimiento de nueva red.
	...

Tabla 2.3: Ejemplo de Traza en Gestión de Claves

Inicialización del Grafo Certificado

Durante la inicialización de la red se crean los repositorios iniciales en el momento en el que va a comenzar la red. En este momento los nodos firman las claves públicas de sus nodos cercanos con su clave privada, y su clave pública es firmada con las claves privadas de sus nodos conocidos. A continuación se intercambian las firmas generadas y de esta manera

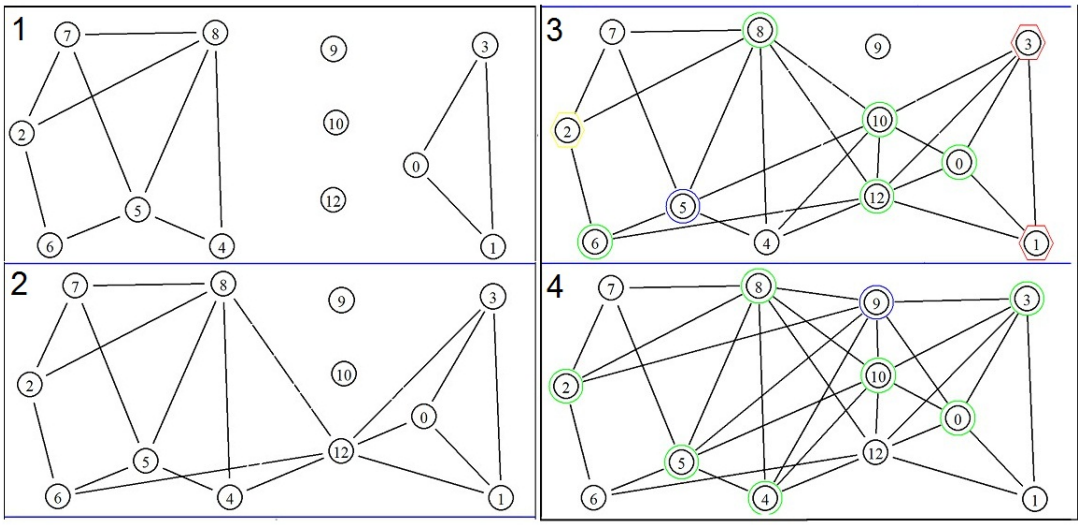


Figura 2.14: Transformaciones del Grafo Certificado

se construyen las aristas correspondientes en el grafo certificado.

En la simulación esto se realiza mediante el procedimiento Iniciar-Red al que se le pasa como parámetro el número de nodos encendidos que van a iniciarse en la red.

Para cada nodo inicial se estudian los nodos que están a distancia 1 de dicho nodo y tras comprobar que están encendidos, se añaden al repositorio inicial de dicho nodo. Además, se comprueba la longitud de los repositorios de cada uno de esos nodos y se añade dicha longitud, que es su grado, a una lista de grados.

```
Calcula-nodos-distancia1 $nodoA 1.0000001 $nodosiniciales
```

A continuación se añaden en una matriz los grados que conoce cada nodo, que serán los de sus nodos vecinos. Finalmente, para cada nodo se ordenan los nodos conocidos de mayor a menor grado y se colocan en su cadena $C1$ el de mayor grado y en su cadena $C2$ el segundo de mayor grado.

Para el grafo certificado de la red, inicialmente se sitúan los nodos iniciales de la red con las conexiones hacia sus vecinos, lo que permite formar el grafo certificado inicial. Las conexiones de nodos nuevos en la red no se dibujarán hasta el instante en el que el nuevo nodo entra en la red.

Conexiones en la Red

```
while {([lindex $grados_ordenados $contador] > [lindex $vector_grados($X) $sig_elemento])
&& ($contador<=[llength $lista_ordenada])} {
        set contador [expr $contador + 1]
}
#cuando salga del while tendremos la posicion a introducir el elmento
set lista_ordenada [linsert $lista_ordenada $contador $sig_elemento]
set grados_ordenados [linsert $grados_ordenados $contador [lindex $vector_grados($X) $sig_elemento] ]

set vector_grados($elemento) [lreplace $vector_grados($elemento) $elemento $elemento $n_elementos]
```

Antes de cualquier tipo de conexión, ya sea para intercambiar datos o repositorios, se debe comprobar si existe confianza entre el nodo A y el nodo B. Para realizar esta comprobación se ha implementado la función Comprobar_confianza.

En primer lugar se inicializa la variable puede_conectarse a 0 y a continuación comienzan las comprobaciones. Se comprueba en el repositorio inicial del nodo A, denotado en el programa distancia1, si se encuentra el nodo B. En todos los casos, en caso de comprobarse la confianza, se marca la variable puede_conectarse a 1.

```
set puede_conectarse 0
#COMPROBAR SI LOS NODOS SE CONFIAN EL UNO EN EL OTRO PARA COMUNICARSE
        #Comprobamos vector de distancia1
        if {([lsearch -exact $distancia1($nodo_a) $nodo_b ] !=-1 )} {
                set puede_conectarse 1
                puts "#\n# Nodo $nodo_b en distancia1 de $nodo_a , distancia1 de $nodo_a: $distancia1($nodo_a)\n#"
        #COMPROBAR EL VECTOR DE DESTINOS(REPOSITORIO DESTINOS)
        } elseif {([lsearch -exact $c1($nodo_a) $nodo_b ] !=-1 )} {
                set puede_conectarse 1
```

Si no, se comprueba si el nodo A se encuentra en las cadenas del nodo B o viceversa.

A continuación se comprueba si alguno de los nodos de distancia1 del nodo A o del nodo B se encuentra en las cadenas del otro nodo o si alguno de los nodos de 1 ó 2 del nodo A se encuentra en los repositorios distancia1, 1 ó 2 del nodo B.

```
for {set M 0} {$M <  [llength $distancia1($nodo_a) ] } {incr M} {
        set elemento_buscado [lindex $distancia1($nodo_a) $M]
        if {([lsearch -exact $ c2($nodo_b) $elemento_buscado] != -1)||([lsearch -exact $c1 ($nodo_b) $elemento_buscado] != -1)
        || ([lsearch -exact $distancia1($nodo_b) $elemento_buscado] != -1) } {
                set puede_conectarse 1
                puts "#\n# Distancia 1 de Nodo $nodo_a en destinos de $nodo_b ,
                distancia1 de $nodo_a: $distancia1($nodo_a) ,destinos de $nodo_b: $ c2 ($nodo_b)\n#"
        }
```

Finalmente, si no puede conectarse se indica con un mensaje en el nam.

Para actualizar repositorios, primero se comprueba el fichero donde se guardan las conexiones para comprobar si es posible la conexión. En caso de que la conexión no sea posible por que los nodos no confían el uno en el otro, las conexiones que fallan se marcan

```
if {$puede_conectarse == 0} {
      puts "#\n#NODO $nodo_a no confia en $nodo_b , distancia1 de $nodo_a: $distancia1($nodo_a), distancia1 de $nodo_b:
      $distancia1($nodo_b) destinos de $nodo_a: $c2 ($nodo_a) ,destinos de $nodo_b: $c2 ($nodo_b)\n#"
}
```

en noconexionesgestionclaves.txt.

```
Comprobar_confianza  $nodo_origen $nodo_destino

if {$puede_conectarse == 1} {
      set fichero [open "./escenario/conexionesgestionclaves.txt" a]
            set lineaameter $momento_conexion
            lappend lineaameter $nodo_origen
            lappend lineaameter $nodo_destino
            puts $fichero $lineaameter
      close $fichero
      set contador_conexiones [expr $contador_conexiones + 1]
      puts "#\n# $nodo_origen se conecta a $nodo_destino en el momento $momento_conexion\n#"
      puts "set tcp_($contador_conexiones) \[\$ns_ create-connection \
            TCP \$node_($nodo_origen) TCPSink \$node_($nodo_destino) 0\]";
```

En el fichero que contiene el grafo certificado de la red se cargan las conexiones producidas en la red así como los intentos de conexiones que no se lograron en la red. En estos ficheros se introducen tanto el momento en el que se produce la conexión o el intento, como los nodos implicados (origen y destino). Además se introduce un comentario para visualizar el suceso en la herramienta de visualización NAM.

```
set fich [open "./escenario/conexionesgestionclaves.txt" r]
      set conexion [gets $fich]
      while { $conexion != "" } {
            set momento [lindex $conexion 0]
            set nodoorigen [lindex $conexion 1]
            set nododestino [lindex $conexion 2]
            $ns at $momento "$ns trace-annotate  \" NODO $nododestino
                              CONECTA CON $nodoorigen  en momento $momento.\" "
            $ns at $momento " $n($nodoorigen) add-mark m0 green circle"
            $ns at [expr $momento + 0.3]  "$n($nodoorigen) delete-mark m0"
            $ns at $momento "$n($nododestino) add-mark m1 blue circle"
            $ns at [expr $momento + 0.3] "$n($nododestino) delete-mark m1"
```

A continuación se buscan en el fichero grafogestionclaves.txt las aristas que cada nodo conoce para dibujarlos. Se debe tener en cuenta que este fichero contiene las aristas finales de la red, y por tanto se hace uso de otros ficheros, nuevosnodosred.txt y nuevosen-repositorio.txt, donde se almacenan las aristas y los nodos introducidos en la red para no dibujar estas aristas hasta el momento en el que se crean realmente en la red. Para esto se utiliza la función Dibujar-si-puede a la que se le pasan como parámetros de entrada el

momento de la conexión, los nodos implicados, el color para la arista y el nodo que contiene dicha arista en su repositorio.

```
set fichero [open "./escenario/grafogestionclaves.txt" r]
 if {$nodoa == $nodoorigen } {
       #MARCAMOS CONEXIONES DISTANCIA 1
       for {set j 0} {$j < [llength $distancia1]} {incr j} {
             set nodob [lindex $distancia1 $j]
             Dibujar-si-puede $momento $nodoa $nodob green $nodoorigen
```

La función Dibujar-si-puede llama a su vez a Puede-dibujar-aristas con los mismos parámetros exceptuando el color de la arista. Puede-dibujar-aristas realiza la comprobación si la arista fue introducida en medio de la simulación o no, y comprueba en caso de ser introducida, si el momento en el que se introdujo fue anterior al momento en el que se encuentra la simulación.

Un dato importante a tener en cuenta es que no se accede a los ficheros donde se guardó esta información cada vez. Lo que se hace es guardar dichos ficheros en listas con lo que se agiliza la búsqueda.

```
proc Puede-dibujar-aristas { momento nodoa nodoorigen nododestino} {
global tiempos_anaden_nodo nodos_anaden_aristas nodoorigen_que_se_anade nododestino_que_se_anade puede ns
set puede 1
#antes de nada compruebo si ese nodo a añadido alguna arista para sino no buscar
      if {[lsearch -exact $nodos_anaden_aristas $nodoa]!=-1} {
            for {set j 0} {$j < [llength $tiempos_anaden_nodo]} {incr j} {
                  #compruebo si el nodoa añade esa arista en un momento futuro
                  if {[lindex $tiempos_anaden_nodo $j] > $momento} {
                        if {[lindex $nodos_anaden_aristas $j]==$nodoa} {
                              #si coinciden es que se añade
                              if {([lindex $nodoorigen_que_se_anade $j]==$nodoorigen)&&
                                 ([lindex $nododestino_que_se_anade $j]== $nododestino)} {
                                    set puede 0
                              #      $ns at $momento "$ns trace-annotate  \" NODO $nodoa  NO PUEDE CREAR CONEXION
                              } elseif {([lindex $nodoorigen_que_se_anade $j]==$nododestino)&&
                              ([lindex $nododestino_que_se_anade $j] == $nodoorigen)} {
                                    set puede 0
                              #      $ns at $momento "$ns trace-annotate  \" NODO $nodoa  NO PUEDE CREAR CONEXION
                              }
                        }
                  }
                  # si el momento es mayor que cuando se añade, puede dibujarse la arista
            }
      }
}
```

Finalmente la función Dibujar-si-puede comprueba el color de la arista y marca el momento inicial y final del cambio de color de la arista. Los colores tienen diferentes momentos de inicio para apreciar el instante en el que se dibuja la siguiente arista, y cuáles

son las aristas que comparten ambos nodos.

Inserción de Nodos

La inserción de un nuevo nodo en la red produce cambios a tener en cuenta tanto en la red como en su grafo certificado. En el procedimiento Inserciones-Apagados-Red al que se le pasa como parámetro el momento aleatorio en el que se produce la inserción o apagado del nodo en la red, se escoge aleatoriamente un nodo y se comprueba si está encendido o apagado. En caso de estar encendido se apaga. En caso de estar apagado se comprueba si el nodo es legítimo de la red o si no lo es. En caso de ser legítimo, se enciende, mientras que en caso contrario se introduce un nuevo nodo en la red.

```
calculalugar $nodoM $tiempo $nodoemisor
#########################################################################
####En este momento sabemos la posicion de los dos nodos, comprobamos la distancia
#########################################################################
set distancia [expr sqrt(($posxfinalnodoa - $movimientox)*($posxfinalnodoa - $movimientox)+
($posyfinalnodoa - $movimientoy)*($posyfinalnodoa - $movimientoy))]

if {$distancia <= $distanciaBroadcast} {
        #se añade a la lista para realizar la coneccion con el.
        lappend listacercanos $nodoM}
```

La inserción de un nuevo nodo en la red comienza con el cálculo de los nodos cercanos a éste en el momento de la inserción. En ese momento se tienen los nodos que están en el repositorio distancia1 y se conoce el grado del nodo que corresponde a la longitud de dicho repositorio. Se añade en la matriz de grados el grado del nuevo nodo a los nodos cercanos y los grados de los nodos cercanos al nuevo nodo. Antes de esto se añade el nuevo nodo al conjunto distancia1 de los nodos cercanos, con lo que el grado de dichos nodos aumenta.

Después se comprueba si los nodos con los que se conecta este nuevo nodo tenían nodos en sus repositorios C1 y C2. De no ser así, se introduce en la cadena que corresponda, al ser en este caso, el nuevo nodo el que tendría mayor grado de los conocidos por ese nodo.

```
if {[llength $c1 ($elemento)]==0} {
        lappend $c1 ($elemento) $nodo
} elseif {[llength $c2 ($elemento)]==0} {
        lappend $c2 ($elemento) $nodo
}
```

Tras esto, el nuevo nodo lanza un broadcast GRI con el que los nodos conecta-

dos podrán saber a cuántos nodos están conectados en esta red. Para esto, cada nodo le pasará sus nodos conocidos y estos nodos serán insertados en una lista sin duplicar, permitiendo saber cuántos nodos conforman la red para el tamaño del repositorio. Este broadcast GRI es necesario debido a que el nuevo nodo puede ser el nexo entre dos subredes y el tamaño de los repositorios tendría que crecer en este caso. Si no se diese esta posibilidad no sería necesario dicho broadcast y sería suficiente con sumar una unidad al número de nodos conocidos por sus nodos vecinos ya que en la actualización de repositorios, el resto de nodos se irían enterando de la presencia del nuevo nodo en la red.

Se introduce entonces el momento, nuevo nodo y lista distancia1 en el fichero nuevosnodosred.txt para la posterior construcción del grafo certificado de la red.

```
broadcastidayvuelta $nodo $momento
#ya está, ahora se añade en el fichero de nuevos nodos
set fich [open "./escenario/nuevosnodosred.txt" a]
        set aux $momento
        lappend aux $nodo
        lappend aux $distancia1($nodo)
        puts $fich $aux
close $fich
```

A continuación se ordenan los nodos conocidos por orden de grado y se introduce el de mayor grado en C1 y el de segundo mejor grado en C2. Se introducen las etiquetas de inserción y termina el proceso de inserción.

Para la inserción de un nuevo nodo en la red, se lee del fichero nuevosnodosred.txt y se dibujan las conexiones de color blanco hasta el momento en el que dichos nodos entran en la red.

```
set fich [open "./escenario/nuevosnodosred.txt" r]
        set datosnodo [gets $fich]
        while {$datosnodo != ""} {
                set momento [lindex $datosnodo 0]
                set nuevonodo [lindex $datosnodo 1]
                set nodoscercanos [lindex $datosnodo 2]
                for {set x 0} {$x < [llength $nodoscercanos]} {incr x} {
                        set nodoaux [lindex $nodoscercanos $x]
                        $ns at 0.00001 "$ns duplex-link-op $n($nuevonodo) $n($nodoaux) color white "
                        $ns at $momento "$ns duplex-link-op $n($nuevonodo) $n($nodoaux) color black "
                }
        $ns at $momento "$ns trace-annotate  \" NODO $nuevonodo ENTRA A FORMAR PARTE DE LA RED.\" "
        set datosnodo [gets $fich]
        }
close $fich
```

Apagado de Nodos

Teniendo en cuenta que aunque un nodo se apague, las certificaciones que este nodo ha realizado siguen siendo válidas y su repositorio sigue almacenado, por lo que el apagado de un nodo en la simulación es sencillo. Se introducen las etiquetas y comentarios y se coloca el tiempo de actualización a un valor alto.

```
set encendido($nodo) 0
set tiempo_intercambio($nodo) 10000

...
puts "#\n# Se apaga nodo\n#"
puts "\$ns_ at $momento \"\$node_($nodo) setenergy 0.0 \""
puts "\$ns_ at $momento \"\$ns_ trace-annotate \\\"Nodo $nodo se apaga en $momento \\\" \" "
#..... Node $nodo turn off.\\\"\" "
puts "\$ns_ at $momento \"\$node_($nodo) label APAGADO\""
```

Cuando un nodo se apaga. No influye nada en los repositorios de los nodos. Simplemente el nodo apagado dejará de participar en la red mientras se encuentre apagado. En caso de encenderse, no tendrá que insertarse y simplemente será como si fuese legítimo en todo momento.

Actualización de Repositorios

La actualización de repositorios es el método con el que los nodos obtienen la información para comunicarse con el resto de nodos de la red con confianza.

Por tanto, un buen algoritmo de actualización de repositorios debe permitir que almacenando la menor cantidad de información posible, todos los nodos de la red se puedan comunicar entre sí.

Lo primero que hay que tener en cuenta en la simulación es el momento en el que se realiza la actualización de repositorios. Para ello está el procedimiento Comprueba-actualización-repositorios al que se le pasa el momento anterior y posterior entre los cuales se comprueba si existen nodos que deban actualizar sus repositorios.

Este procedimiento se lanza en el bucle del programa principal antes y después de Inserciones-Apagados-Red, que es donde se realizan los encendidos y apagados de los nodos en la simulación.

Por tanto, en el bucle de red, el cual se encuentra separado en fragmentos de tiempo, se tiene n: el momento en el que se inicia el fragmento de tiempo, un momento aleatorio en medio del fragmento de tiempo que es cuando sucederá un encendido o apagado en la red y un momento final del fragmento de tiempo en la red.

```
set tiempo_aux [expr $tiempo_aux + $periodo]
set tiempo_b [$rng uniform $tiempo_a $tiempo_aux]
Comprueba-actualizacion-repositorios $tiempo_a $tiempo_b
Inserciones-Apagados-Red $tiempo_b
```

En el procedimiento *Comprueba-actualizacion-repositorios* se comprueba si existe algún nodo cuyo momento de actualización se encuentre entre los instantes pasados como parámetros.

```
for {set M 0} {$M <  $opt(nn) } {incr M} {
    if {($tiempo_intercambio($M)>$tiempo_a) && ($tiempo_intercambio($M)<$tiempo_b)} {
```

En caso de encontrarse, se introducen en una lista ordenada donde se encuentran todos los nodos que realizan actualización de repositorios entre el momento tiempo_a y el momento tiempo_b pasado. Puede suceder que el fragmento de tiempo sea suficientemente grande como para que tras una actualización de un nodo, sume el tiempo del periodo y la siguiente actualización siga siendo en el mismo fragmento de tiempo. En este caso la lista se actualiza y se introduce el elemento en la posición que le corresponda según el momento que le toque actualizarse.

A continuación se van extrayendo de esa lista y calculando los nodos cercanos a éste.

```
Calcula-nodos-distancia1 $M $tiempo_intercambio($M) $opt(nn)
```

Se comienza entonces la actualización de repositorios. La primera tarea consiste en crear las conexiones de intercambio entre paquetes. Seguidamente se comprueba si tienen o no confianza para actualizar repositorios o comunicarse.

Si logra conectarse, se introduce en el fichero conexionesgestionclaves.txt y se dibuja un círculo verde alrededor del nodo con el que se conecta, y si no, se introduce en noconexionesgestionclaves.txt y se dibuja un círculo rojo.

```
conexion-udp $M $L $tiempo_intercambio($M)
set t2 [expr $tiempo_intercambio($M) + 0.02]
conexion-udp $L $M  $t2
Comprobar_confianza  $M $L
```

Si los nodos se conectan y confían el uno en el otro, actualizan la información de los grados de los nodos que conoce cada uno con el repositorio del otro. A continuación, se guardan todos los nodos que conoce en la lista auxiliar todos_conocidos y comienza el intercambio. Se comprueba cuál de las dos cadenas es la más corta y se lanza el procedimiento inserta_vertice con el nodo que actualiza, nodo que le pasa su repositorio y la cadena que va a intentar actualizar. Esto se hace de esta forma para que las cadenas crezcan de forma equitativa y que a la hora de comprobar la confianza entre distintos nodos, el proceso sea más rápido debido a que las ramas formadas por las dos cadenas son lo más cortas posibles.

```
if { [llength $c1($M)] <= [llength $c2  ($M)]} {
        set  quecadena  1
        puts "#\n# COMPRUEBA C1 $M\n#"
        inserta_vertice $M $L $quecadena
        puts "#\n#COMPRUEBA C2 $M\n#"
        set  quecadena  2
        inserta_vertice $M $L $quecadena
} else {
```

El procedimiento inserta_vertice se encarga de insertar en el repositorio del nodo indicado un nuevo nodo (si procede) en el extremo de la cadena indicada. Para esta inserción se pueden dar tres casos distintos. Primero, el nodo con el que actualiza se corresponde con el vértice de la cadena que va a actualizar. En este caso, se comprueban los primeros nodos de sus dos cadenas, y si no tiene alguno, lo introduce y continúa en el bucle del procedimiento Comprueba-actualizacion-repositorios.

```
if {$L == $nodo_vertice} {
        set aux [lindex $c1($L) 0]
        if {[lsearch -exact $todos_conocidos $aux] == -1} {
                if {$aux!=$M} {
                        if {[expr [llength $distancia1($M)] + [llength $c1($M)] +[llength $c2($M)] -2]
                        < [expr [llength $conocenumnodos($M)] / 2.0]} {
```

Si el nodo no se corresponde con el nodo vértice, se comprueban las cadenas de dicho nodo para ver si algún nodo de sus cadenas se corresponde con el nodo vértice. En caso de que así sea, se comprueba si tiene el nodo anterior y posterior si existe, y coge el

de mayor grado o el nodo que no contenga en su repositorio. Se debe tener en cuenta si el nodo es el primero de la cadena, el nodo anterior es el propio nodo con el que intercambia. En caso de insertar un nuevo nodo en el repositorio, continuaría intentando insertar desde el bucle en el procedimiento Comprueba-actualizacion-repositorio.

2.4.4. Algoritmo de Máximo Grado por Sectores

En el método de máximo grado por sectores descrito a continuación se intenta conseguir que el subgrafo repositorio verificado de cada nodo crezca hacia todas las partes del grafo certificado de la red.

La idea de este método surgió tras analizar el algoritmo de máximo grado con 2 cadenas y comprobar que pueden existir problemas en la forma de expandir sus cadenas en los repositorios verificados en determinados casos de redes con una distribución alargada. Por ejemplo, en la Fig. 2.15 se puede ver cómo la actualización de ciertos repositorios hace que sus dos cadenas que tienden a expandirse hacia los nodos con mayor grado, crezcan hacia el mismo lado del grafo de la figura, y por tanto no consigan alcanzar nodos del otro lado.

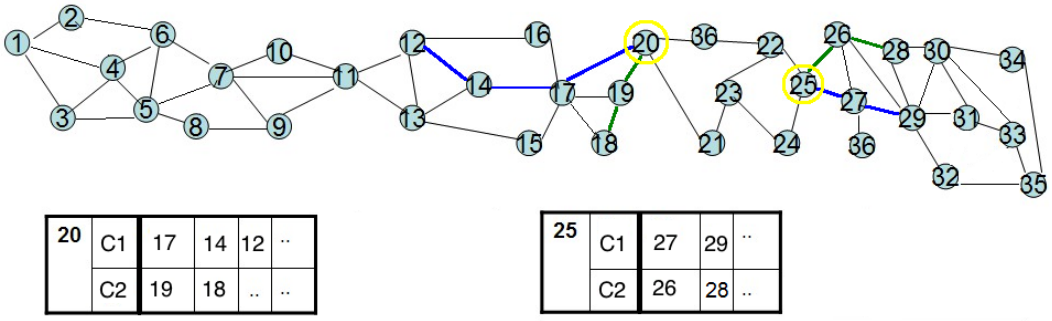


Figura 2.15: Problema en Algoritmo de Máximo Grado con Dos Cadenas

Se puede observar que el repositorio del nodo 25 tiende a crecer hacia el subgrafo que está a su derecha, ignorando el subgrafo de la izquierda. Igual sucede hacia el otro lado con el nodo 20, cuyo repositorio se expande hacia el subgrafo izquierdo, ignorando en este caso el subgrafo derecho. Estos dos nodos a pesar de estar a una distancia de dos saltos

no podrán comunicarse entre ellos porque no podrán encontrar ni siquiera un nodo común conocido.

La inicialización es igual que en el método del algoritmo de máximo grado con dos cadenas. Para la actualización del repositorio verificado se ejecuta el siguiente algoritmo, donde *almacen* representa el repositorio verificado del nodo ejecutor y $B.almacen$ el del nodo con el que se ha autenticado.

Algoritmo Máximo Grado por Sectores

```
00: ...//Autenticación con nodos, intercambiando repositorios verificados
01: //Actualización de los grados de los nodos en los repositorios.
02: para (i = 0; i < tam(almacen); i++)
03:    para (j = 0; j < tam(B.almacen); j++)
04:       si (almacen(i) == B.almacen(j))
05:          si ((tam(almacen) < limAlmacen)&&(noTenemos(sgte(B.almacen(j)))))
06:             añade(sgte(B.almacen(j)));
07:          sino si ((tam(almacen) == limAlmacen) &&
              (grado(B.almacen(j)) > minGrado))
08:             eliminaNodoMinGrado();
09:             añade(B.almacen(j));
10:             actualizaNodoMinGrado();
11:          fin si
12:       fin si
13:    fin para
14: fin para
```

Las estructuras de datos que se usan para almacenar los nodos en este método varían respecto al método anterior. Para este método los repositorios usados se denotan Distancia1 y Origen-Destino.

- *Distancia1:* Este repositorio guarda la misma información que en el algoritmo de máximo grado con dos cadenas.

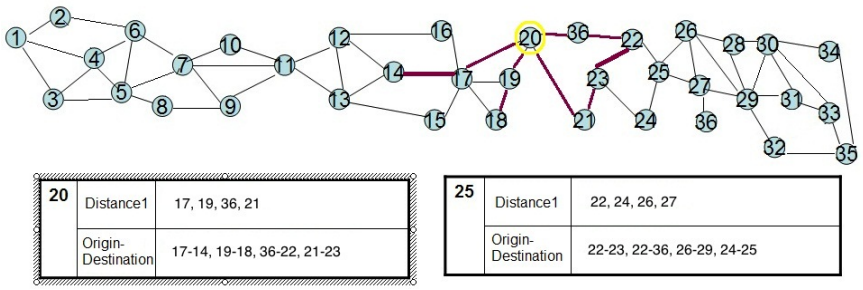
- *Origen-Destino:* En este repositorio se guardan las aristas que añade cada nodo al realizar la actualización de su repositorio mediante el intercambio de información con otros nodos con los que se encuentra.

Al igual que en el algoritmo anterior los nodos antes de intercambiar sus repositorios deben comprobar la confianza con los nodos con los que se conecta. En principio, el repositorio Origen-Destino contendrá las aristas correspondientes a los vértices de mayor grado de todos los vecinos del nodo en cuestión.

Para actualizar los repositorios, los nodos deben confiar en los nodos con los que intercambia su repositorio. Para ello el nodo B le envía al nodo A los dos nodos de su repositorio compuesto por las tablas (distancia1 y origen-destino) con mayor grado, en caso de que el tercer nodo y sucesivos tengan igual grado que el segundo mejor, también los envía. El nodo A comprueba si tiene los nodos recibidos, y si no los tiene los añade a su repositorio origen-destino. En caso de tenerlo, comprueba el siguiente de igual grado hasta encontrar uno que no tenga o quedarse sin nodos por comprobar. Si se encuentra al menos una arista origen-destino nueva que añadir, el tiempo para actualizar repositorios disminuye en una unidad cada vez hasta alcanzar el mínimo que está fijado en dos unidades. Si no se encuentra ninguna arista que añadir por ninguno de los nodos a distancia 1 en el momento de actualización, el tiempo para la próxima actualización de repositorios aumenta en una unidad para ese nodo.

Con el algoritmo de máximo grado con dos cadenas se puede observar que al no tener limitaciones de crecimiento en x número de cadenas, éstas pueden crecer hacia todos lados por lo que con cada nodo con el que se intercambia, cogerá siempre sus mejores cadenas. Por el contrario, con este método el número de nodos que intentará almacenar en los repositorios será mayor que con el algoritmo de máximo grado con dos cadenas.

En la Fig. 2.16 se puede comprobar para el mismo ejemplo de la Fig. 2.15, que aplicando el algoritmo de máximo grado por sectores que se describe a continuación, no se presenta el problema de crecimiento que tiene el algoritmo de Máximo Grado con dos Cadenas.



20		
	Distance1	17, 19, 36, 21
	Origin-Destination	17-14, 19-18, 36-22, 21-23

25		
	Distance1	22, 24, 26, 27
	Origin-Destination	22-23, 22-36, 26-29, 24-25

Figura 2.16: Ejemplo del Algoritmo de Máximo Grado por Sectores

2.4.5. Simulación del Algoritmo de Máximo Grado por Sectores

La simulación de la red para el esquema de gestión de claves usando el algoritmo de máximo grado por sectores es similar a la simulación realizada para el algoritmo de máximo grado con 2 cadenas. Sin embargo hay algunas diferencias destacables con respecto a la actualización y uso de los repositorios que almacenan cada nodo.

Inicialización de la Red

En este caso no es necesario introducir los nodos de mayor grado en dos estructuras C1 y C2. Simplemente se realizan los intercambios de claves con los nodos vecinos y se inicializan los repositorios distancia1 de cada nodo. A continuación se intercambian dichos repositorios distancia1 para inicializar el repositorio origen-destino.

En la implementación del repositorio origen-destino se distinguen dos listas para cada nodo, una que contiene el nodo origen y otra que contiene el nodo destino correspondiente en la misma posición de la lista origen.

Inserción de Nodos

En el proceso de inserción no es necesario guardar los nodos con mayores grados en C1 y C2, ya que en este algoritmo dichos nodos se almacenan todos en el mismo repositorio origen-destino.

Apagado de Nodos

El apagado de un nodo, al igual que en el algoritmo de máximo grado con 2 cadenas, no influye en nada a la red ni a los repositorios almacenados por cada nodo.

Actualización de Repositorios

El momento de actualización de repositorios no varía con respecto al algoritmo anterior. Se realiza antes y después de los intervalos de tiempo indicados en cada inserción y apagado de nodos en la red en el bucle del programa principal.

Comprueba-actualizacion-repositorios comienza igual que en el algoritmo anterior pero varía el momento en que debe actualizar los repositorios.

Inicialmente se comprueba si existen nodos que vayan a actualizar sus repositorios entre los tiempos pasados. En ese caso se ordenan todos los que vayan a actualizar en orden según su momento de actualización. Para cada uno de esos nodos, tras marcarlos con un círculo azul e indicar que comienza su cambio de repositorio, se calculan los nodos cercanos a él que están encendidos en ese momento. Con esos nodos se establece una comunicación donde se intercambian los repositorios y se comprueba si pueden establecer comunicación llamando a la función "Comprobar_confianza". Si es posible, se introduce el momento y nodos que han conectado en el fichero conexionesgestionclaves.txt. En caso contrario se introducen en noconexionesgestionclaves.txt.

A todos los nodos con los que conecta se les pone un círculo verde alrededor. A continuación, cada nodo que conecta actualiza los grados conocidos con el intercambio de información de repositorios. En ese momento se cogen todos los nodos del repositorio de B y se ordenan por grado. El nodo A introduce el nodo de mayor grado del nodo B en caso de no tenerlo.

Seguidamente introduce el nodo que tiene el segundo mayor grado. Y en caso de tener ya dicho elemento en el repositorio, sigue buscando nodos mientras tenga el mismo grado que el nodo con segundo mayor grado.

Finalmente se escribe un mensaje indicando si el nodo ha añadido alguna arista o no y en consecuencia el periodo aumenta o disminuye. A continuación se recalcula la lista

```
#AHORA CON LA LISTA ORDENADA POR GRADOS, COGEMOS LOS 2 PRIMEROS
ELEMENTOS Y LOS SIGUIENTES CON IGUAL GRADO DEL 2º
#si el primer elemento no lo tengo en distancia1 ni en destino del almacen lo añado a mi almacen
set primerelementoaintroducir [lindex $lista_ordenada 0]
if {  ([lsearch -exact $distancia1($M) $primerelementoaintroducir]==-1) &&
([lsearch -exact $destino_almacen_repositorio($M) $primerelementoaintroducir]==-1) &&
($primerelementoaintroducir != $M) } {
      #comprobar que el origen del elemento q vamos a introducir es conocido
      if { ([lsearch -exact $distancia1($M) [lindex $lista_origen_ordenada 0]]!=-1) ||
      ([lsearch -exact $destino_almacen_repositorio($M) [lindex $lista_origen_ordenada 0]]!=-1) } {
            #comprobamos si el almacen del nodo que va a introducir esta lleno,
            #si es asi comprobamos si el nodo a entrar es mejor q alguno de sus extremos
            if {[expr [llength $distancia1($M)] + [llength $destino_almacen_repositorio($M)]] <
            [expr [llength $conocenumnodos($M)] / 2.0]} {
```

lista_ordenada_final para introducir el elemento en la posición que le corresponda en caso de que el momento de la nueva actualización de repositorios siga estando dentro del periodo indicado en la función Comprueba-actualizacion-repositorios.

2.4.6. Estudio Comparativo

Tras realizar 25 simulaciones para 15, 20, 30 y 60 nodos y obtener los resultados en media para las simulaciones de los diferentes tipos de redes comprobamos que el método Máximo Grado por Sectores tiene en general mejor o igual comportamiento que el algoritmo de máximo grado con dos cadenas, aunque los resultados son bastante parecidos. A continuación se muestran los resultados.

```
set ningunometido 0
set gradosegundoelto [lindex $grados_ordenados 1]
while {($ningunometido==0) && ($recorredor < [llength $grados_ordenados]) &&
([lindex $grados_ordenados $recorredor]==$gradosegundoelto) } {
      if { ([lsearch -exact $distancia1($M) [lindex $lista_ordenada $recorredor]]==-1) &&
      ([lsearch -exact $destino_almacen_repositorio($M) [lindex $lista_ordenada $recorredor]]==-1) &&
      ([lindex $lista_ordenada $recorredor] != $M)} {
            if { ([lsearch -exact $distancia1($M) [lindex $lista_origen_ordenada $recorredor]]!=-1) ||
            ([lsearch -exact $destino_almacen_repositorio($M) [lindex $lista_origen_ordenada $recorredor]]!=-1) } {
                  #comprobamos si el almacen del nodo que va a introducir esta lleno,si es asi comprobamos
                  si el nodo a entrar es mejor q alguno de sus extremos
                  if {[expr [llength $distancia1($M)] + [llength $destino_almacen_repositorio($M)]] <
                  [expr [llength $conocenumnodos($M)] / 2.0]} {
                        set elto_anadido [lindex $lista_ordenada $recorredor]
                        lappend destino_almacen_repositorio($M) $elto_anadido
                        lappend origen_almacen_repositorio($M) [lindex $lista_origen_ordenada $recorredor]
                        puts "\$ns_ at $tiempo_intercambio($M) \"\$ns_ trace-annotate  \\\"  $M AÑADE 2º.
                        $recorredor elemento ( $elto_anadido ) de $L al repositorio: $destino_almacen_repositorio($M) |
                        Origen $origen_almacen_repositorio($M) | Distancia1 $distancia1($M) |
                        Tiempo del proximo intercambio disminuye \\\" \" "
                        set anade_algo 1
                        set ningunometido 1
                        lappend tiempos_anaden_nodo $tiempo_intercambio($M)
                        lappend nodos_anaden_aristas $M
                        lappend nodoorigen_que_se_anade [lindex $lista_origen_ordenada $recorredor]
                        lappend nododestino_que_se_anade $elto_anadido
                  } else {
                        puts "\$ns_ at $tiempo_intercambio($M) \"\$ns_ trace-annotate  \\\"
                        $M NO AÑADE POR REPOSITORIO LLENO DISTANCIA1: $distancia1($M) |
                        REPOSITORIO: $destino_almacen_repositorio($M)  \\\" \" "
                  }
            }
      }
      set recorredor [expr $recorredor + 1]
}
```

```
if {($tiempo_intercambio($M) > $tiempo_a) && ($tiempo_intercambio($M) < $tiempo_b)} {
puts "#tiempo $M $tiempo_intercambio($M)"
      if {[llength $lista_ordenada_final] < 1} {
            set lista_ordenada_final $M
      } else {
            set contador 0
            set elemento [lindex $lista_ordenada_final $contador]
            #puts "#elemento1 $elemento"
            while {($tiempo_intercambio($elemento) < $tiempo_intercambio($M)) &&
            ( $contador < [llength $lista_ordenada_final])} {
                  set contador [expr $contador + 1]
                  set elemento [lindex $lista_ordenada_final $contador]
                  if {$elemento == "" } {break}
            #     puts "#elemento $elemento"
            }
            set lista_ordenada_final [linsert $lista_ordenada_final $contador $M]
            puts "#Listadespues $lista_ordenada_final "
      }
}
```

15 nodos	Máximo Grado 2 Cadenas	Máximo Grado por Sectores
Conexiones totales	876,0	966,9
Conexiones logradas	812,24	909,7
Conexiones falladas	63,8	57,15
Aristas añadidas	27,55	102,28
No añadidas por repositorio lleno	13,4	0
Actualizaciones de repositorios	576,3	628,7
20 nodos	Máximo Grado 2 Cadenas	Máximo Grado por Sectores
Conexiones totales	1069,8	1011,52
Conexiones logradas	1037,9	985,4
Conexiones falladas	31,9	26,12
Aristas añadidas	36,16	56,4
No añadidas por repositorio lleno	2,3	0
Actualizaciones de repositorios	435,4	420,2
30 nodos	Máximo Grado 2 Cadenas	Máximo Grado por Sectores
Conexiones totales	2885,8	2764,7
Conexiones logradas	2870,3	2749,8
Conexiones falladas	15,52	14,92
Aristas añadidas	75,07	80,51
No añadidas por repositorio lleno	2,6	0
Actualizaciones de repositorios	714,39	690,24
60 nodos	Máximo Grado 2 Cadenas	Máximo Grado por Sectores
Conexiones totales	5291,9	5309,0
Conexiones logradas	5183,	5216,5
Conexiones falladas	108,9	92,5
Aristas añadidas	142,8	191,9
No añadidas por repositorio lleno	1,0	0
Actualizaciones de repositorios	1455,52	1475,9

2.4.7. Revocación de Certificados

Con el tiempo las claves pierden parte de su seguridad debido a que un nodo malicioso tiene cada vez más tiempo para probar por fuerza bruta distintas combinaciones, y obtener más información gracias a las comunicaciones realizadas con dicha clave. Por este motivo es recomendable cambiar las claves cada cierto tiempo para evitar que algún nodo malicioso la rompa y pueda descubrir todas las comunicaciones cifradas con ella.

Por este motivo se hace necesaria la expiración y en general la revocación de certificados, que puede producirse de dos formas distintas: de forma implícita o explícita.

La revocación implícita consiste en la revocación del certificado por finalización de

su fecha de caducidad. Tras esa fecha el certificado deja de ser válido a todos los efectos y por tanto las firmas de otros certificados firmados mediante la clave privada de éste también dejan de ser válidas.

La revocación explícita sucede cuando un nodo cree que su certificado puede haber sido vulnerado por algún nodo fraudulento. Entonces dicho nodo envía una revocación explícita de dicho certificado para que los nodos de la red lo quiten o lo sustituyan por un nuevo certificado en las firmas correspondientes a los nodos en los que confía.

Ambas formas de revocación hacen que las cadenas de certificados que se mantienen en los repositorios locales de los nodos sean suprimidas a partir de la firma de cualquier certificado con una clave privada cuyo certificado haya sido revocado o bien no pasa nada con dichas cadenas si la firma correspondiente es sustituida por una firma con un nuevo certificado del mismo nodo.

En caso de eliminación de una cadena por revocación de algún certificado, un nodo debe intentar recuperar los certificados perdidos o encontrar nuevas cadenas sustitutas. En cada broadcast GRI se debe publicar la lista de certificados revocados explícitamente.

Para revocar un certificado implícitamente por caducidad es necesario o bien que exista una autoridad que mantenga un reloj en la red, o bien que se disponga de un *reloj de red* distribuido. En las MANETs objeto de este trabajo no existe ninguna autoridad central, pero esto no supone ningún problema porque aquí se utiliza un *reloj de red distribuido*. Dicho reloj se inicia al comienzo de la red y se mantiene de forma individual en cada nodo. En las pruebas de vida de los nodos *el reloj de red* se sincroniza para la toda la red.

La revocación explícita se realiza utilizando el broadcast GRI donde el nodo que quiere revocar o renovar su certificado debe comunicarlo al resto de la red llegando al menos a más de la mitad de los nodos que conforman la red en ese momento para que sea válido. Luego los nodos que no han actualizado sus repositorios lo actualizan al entrar en contacto con nodos que sí lo hayan hecho.

2.5. Gestión de Topología Mediante RFID

Esta sección plantea brevemente una alternativa centralizada para la gestión de MANETs basándonos en el uso de tecnología RFID (Radio Frequency IDentification), que permite controlar la topología variable de estas redes.

En el esquema propuesto de uso de RFID para la gestión de MANETs se asumen las siguientes ideas:

- Cada nodo de la MANET lleva una etiqueta (tag) adherida para poderlo localizar.
- A la MANET se le añaden los lectores de etiquetas como nuevos nodos especiales, que tienen conexión segura con un servidor central, de forma que la MANET pasa a ser realmente una red híbrida.
- Se logra tener la topología de la MANET controlada, lo que facilita varias cuestiones de gestión como el encaminamiento de paquetes y el broadcast, la distribución de tareas, el control de accesos, las inserciones y eliminaciones de nodos, etc.

En conclusión, el esquema propuesto en esta sección constituye una alternativa centralizada al esquema SLCM descentralizado presentado al comienzo del capítulo.

En particular se aborda el problema fundamental de la autenticación usando RFID describiendo un nuevo esquema ligero para la autenticación mutua entre lectores y etiquetas RFID, que cumplen la norma EPC Gen2 así como los requisitos prácticos de la tecnología RFID de bajo coste y las limitaciones de recursos de las etiquetas pasivas. Además, el esquema implica una mínima interacción entre etiquetas y lectores. La propuesta no se basa en la seguridad de los lectores RFID, ya que estos son portátiles. En lugar de eso, basa su seguridad en la confianza en un servidor central, ya que todos los secretos compartidos son almacenados sólo por las etiquetas y dicho servidor, sin posibles accesos por parte del lector en ningún momento.

2.5.1. Estándar EPC Gen2 de RFID

La tecnología RFID implica el uso de etiquetas y lectores con el propósito de identificarse a través de ondas de radio. El sistema típico de RFID consiste en etiquetas,

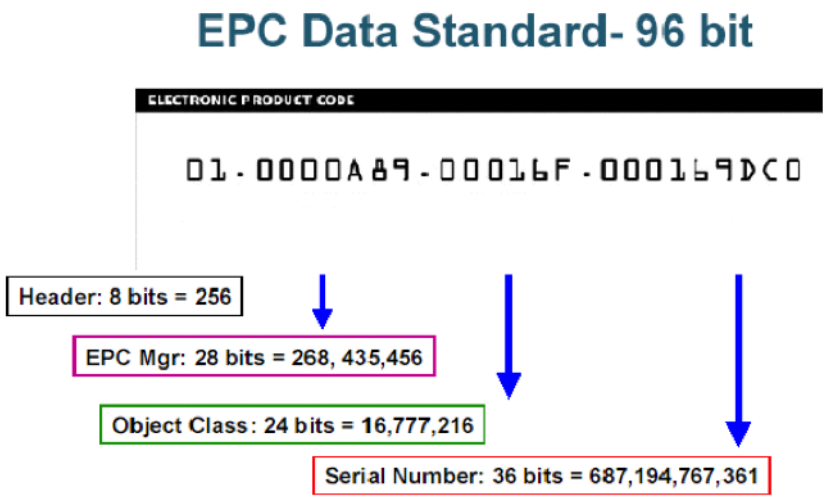


Figura 2.17: Formato de Paquete EPC Data

lectores, y un servidor central con una base de datos que contiene la información sobre las etiquetas que maneja. Las etiquetas y los lectores están conectados a través de comunicación inalámbrica, mientras que los lectores y el servidor se conectan a través de un canal seguro. En 2004, el estándar EPC Class 1 Gen 2 (EPC Gen2) fue ratificado para las implementaciones de RFID [173]. El estándar EPC Gen2 detalla los parámetros que deben tener los lectores que envían y reciben datos desde etiquetas UHF. Especifica también cómo debe ser el uso de los canales y frecuencias, banda ancha y otros aspectos técnicos. Este estándar permite que todos los nodos en una misma red codifiquen sus etiquetas de la misma manera para poder compartir la infraestructura tecnológica e interoperar en ella.

La información que envía la etiqueta al lector se puede desglosar en: 96 bits para el número de identificación único (datos EPC o EPC Data), 16 bits de código CRC, y 32 bits de código de habilitación/deshabilitación (kill code).

En la Fig. 2.17 se muestra el contenido del paquete EPC Data así como el tamaño y el significado de cada uno de sus componentes.

Según este estándar, las etiquetas UHF son elementos pasivos, es decir, que responden a través de la energía que reciben de los lectores, por lo que tienen una capacidad de cálculo muy restringida. También la memoria de la etiqueta es limitada y debe considerarse

insegura y susceptible a ataques físicos. En particular, según la norma EPC Gen2, las etiquetas sólo soportan un generador de números pseudo-aleatorios PRNG (Pseudo-Random Number Generator) de 16 bits y un código de redundancia cíclica CRC (Cyclic Redundancy Code) de 16 bits.

- Header [bits 0 a 7]: Número de versión.
- EPC Mgr [bits 8 a 35]: Número de administración de dominio o fabricante.
- Object Class [bits 36 a 59]: Identificador del objeto o producto.
- Serial Number [bits 60 a 95]: Número de serie.

El CRC permite al lector comprobar que recibió correctamente el paquete EPC Data, ya que el CRC es calculado por la etiqueta que envía el paquete agregándolo al final de éste, y es recalculado por el lector. Los resultados obtenidos tienen que ser los mismos para considerar una recepción correcta.

En EPC, al igual que básicamente en todas las tecnologías inalámbricas, la seguridad en la información es uno de los objetivos fundamentales. Se necesita asegurar aspectos como la confidencialidad, integridad, disponibilidad, autenticidad y autorización. Una etiqueta, por simple que sea, envía su identificador único a cualquier lector de forma que se puede obtener información acerca de él fácilmente, por ejemplo, se puede obtener información personal acerca del pasaporte de una persona.

Aquí se presenta una nueva solución ligera para la autenticación mutua que se ajusta plenamente al estándar EPC Gen2, y que ha sido diseñada pensando en su uso para gestionar y controlar la topología de las MANETs.

2.5.2. Base de la Propuesta

Una solución típica de autenticación mutua basada en clave secreta compartida entre dos entidades consiste en que cada uno tiene que convencer al otro de que conoce la clave secreta compartida. Por lo tanto, para evitar la clonación de etiquetas, se puede utilizar un esquema reto-respuesta basado en criptografía de clave simétrica. Esta es la idea principal detrás del esquema de autenticación mutua propuesto.

Los ataques por repetición representan una posible deficiencia de la tecnología RFID. Con el fin de evitarlos, las soluciones criptográficas más habituales son: uso de secuencias de números crecientes, sincronización del reloj, o uso de números aleatorios (nonces). Las etiquetas pasivas no pueden usar un reloj para sincronizarse porque no tienen ninguna fuente de alimentación. Por otro lado, las secuencias crecientes no son adecuadas para evitar posibles rastreos. Por este motivo en el esquema descrito a continuación se utilizan nonces.

Para proteger los datos transmitidos entre la etiqueta y el lector contra la posible escucha de nodos maliciosos, la solución típica es el cifrado. En particular, la función más simple de cifrado es la operación XOR utilizada en el cifrado en flujo. Sin embargo, en ese caso el problema no es el cifrado, sino la generación y gestión de las claves, porque se necesita producir una nueva clave de cifrado para cada sesión. Esto se soluciona con nuestra propuesta de autenticación ya que no sólo permite autenticar a los nodos, sino que tras dicho proceso ambos comparten una clave secreta compartida.

Finalmente, para evitar el posible seguimiento de la etiqueta se debe actualizar su identificación. Si el conocimiento de la identificación de la etiqueta sólo se comparte entre el servidor y la etiqueta, una forma fácil de actualización es que ambos utilicen el mismo PRNG de forma sincronizada.

2.5.3. Esquema de Autenticación Mutua Lector-Etiqueta

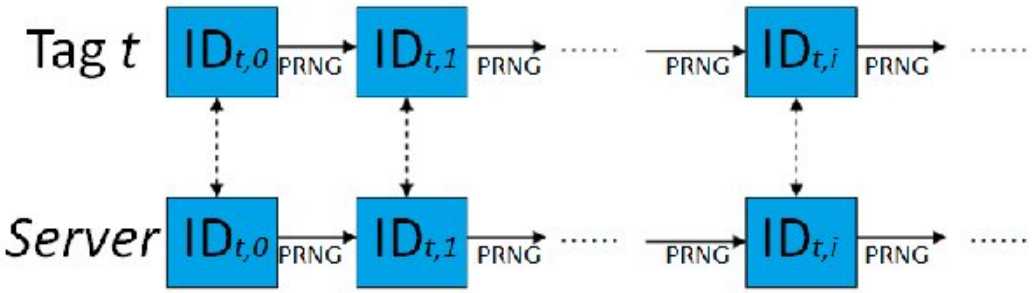


Figura 2.18: Actualización de IDs

A continuación se propone un nuevo mecanismo para proporcionar autenticidad a sistemas RFID de bajo coste que cumplan el estándar EPC Gen2. El método propuesto

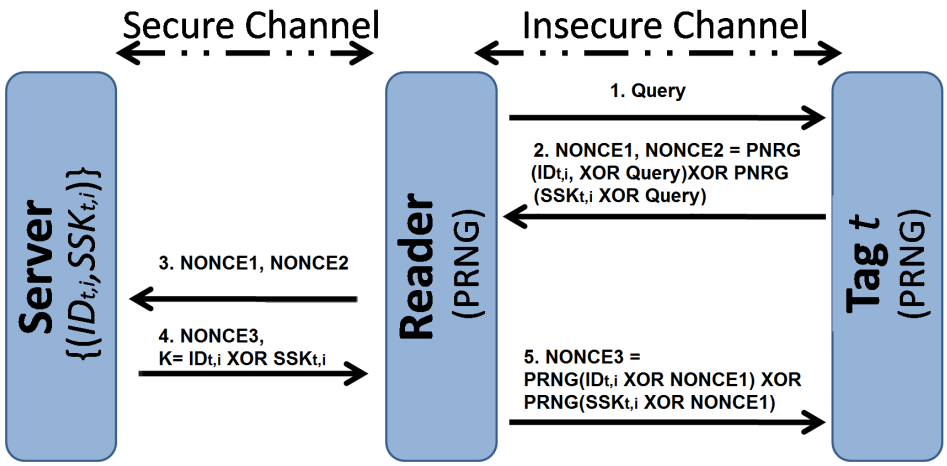


Figura 2.19: Protocolo de Acceso EPC Gen2

puede ser utilizado por el lector y la etiqueta tanto para autenticarse mutuamente como para establecer una clave de sesión secreta compartida. Este método asume que el lector está vinculado a través de un canal de comunicación seguro a un servidor que tiene una base de datos donde cada etiqueta t está relacionada con un par dado por un número de identificación secreto de 16 bits $ID_{t,i}$ y una clave compartida secreta de 16-bit $SSK_{t,i}$ para cada sesión $i = 1, 2, ...$ (ver Fig. 2.18). Se supone en el esquema que el lector y la etiqueta son capaces de usar un generador de números pseudoaleatorios compartido $PRNG$ para actualizar tanto los números de identificación $ID_{t,i}$ como la clave secreta compartida $SSK_{t,i}$. A continuación se describen los pasos del algoritmo propuesto (ver Fig. 2.19) y que pueden generar nonces de longitud $n < 2^{16}$.

Algoritmo Autenticación ligera para RFID

1. El lector envía a la etiqueta un mensaje aleatorio Query de longitud 16.

2. La etiqueta t alimenta el PRNG con ($ID_{t,i}$ XOR Query) y con ($SSK_{t,i}$ XOR Query) para producir dos secuencias cifrantes de $(16 + n)$ bits cuyos últimos n bits son sumados para ser enviados al lector, junto con un NONCE1 de 16 bits.

3. El lector envía los datos recibidos en el paso 2 al servidor, que los compara con todas las salidas correspondientes a los pares almacenados ($ID_{t,j}$, $SSK_{t,j}$).

4. Si el servidor no encuentra ninguna colisión, lo identifica como un posible intento de fraude. De lo contrario, si encuentra una única colisión para ($ID_{t,i}$, $SSK_{t,i}$), envía al lector tanto la clave de sesión K = $ID_{t,i}$ XOR $SSK_{t,i}$, como el resultado NONCE3 de la XOR entre los últimos n bits de las dos secuencias cifrantes de (16 + n) bits producidas por el PRNG sobre ($ID_{t,i}$ XOR NONCE1) y ($SSK_{t,i}$ XOR NONCE1), y actualiza los datos de t a ($ID_{t,i+1}$, $SSK_{t,i+1}$). En otro caso, si el servidor encuentra más de una colisión (aunque la probabilidad de que esto ocurra es insignificante), el servidor informa al lector sobre el fallo para que el proceso se reinicie desde el primer paso.

5. El lector envía a la etiqueta la secuencia NONCE3 recibida.

De acuerdo con el esquema anterior se tienen las siguientes propiedades.

- En el último paso, la etiqueta comprueba si los datos recibidos coinciden con la secuencia producida por ella misma sobre los datos correctos, y si esta verificación es exitosa actualiza su identificación a ($ID_{t,i+1}$, $SSK_{t,i+1}$).

- Después de la ejecución de los cinco pasos del esquema anterior, tanto el lector como la etiqueta pueden utilizar la misma clave de sesión secreta K = ($ID_{t,i}$ XOR $SSK_{t,i}$).

- La información que obtiene un espía del canal inseguro entre el lector y la etiqueta de la escucha repetida de dicho canal es inútil.

La clave de sesión secreta compartida establecida K puede ser utilizada tanto por la etiqueta como por el lector para iniciar el PRNG con el fin de obtener la misma secuencia de clave Z para cifrar y descifrar todos los mensajes intercambiados entre ellos durante la sesión. K puede también ser utilizada luego por la etiqueta y el lector para la autenticación mediante reto-respuesta basada en criptografía simétrica.

En entornos ubicuos podemos suponer que no hay muchos problemas de conectividad, por lo que por simplicidad y seguridad práctica de nuestro esquema, se ha asumido la existencia de una conectividad continua y segura entre los lectores y el servidor. Por otra parte, sin el conocimiento del identificador correspondiente a la etiqueta es muy difícil

construir un valor que el servidor pueda reconocer como válido. Por lo tanto el protocolo propuesto en realidad proporciona autenticación válida a la etiqueta. En cuanto a la protección de la privacidad de la etiqueta, y por tanto del nodo al que está adherida según nuestra propuesta, y a la prevención de ataques de seguimiento, el protocolo propuesto protege tanto la privacidad como los ataques porque la respuesta de la etiqueta en el paso 2 es diferente e impredecible en cada solicitud de autenticación debido a la actualización de sus datos. Nótese también que la actualización de su número de identificación secreto y su clave secreta compartida implica seguridad tanto hacia delante (forward security) como hacia atras (backward security), y resistencia contra ataques de repetición. Además, la etiqueta no proporciona su identificador a cualquier lector y por lo tanto, no hay posibilidad de que un lector legítimo pero malicioso pueda realizar ataques de suplantación de identidad contra ninguna etiqueta.

Finalmente, el ataque MitM es imposible en el esquema propuesto ya que requeriría que el atacante pudiera hacer conexiones independientes con el lector y con la etiqueta con el fin de transmitir mensajes entre ellos para hacerles creer que están hablando directamente el uno con la otra, cuando en realidad toda la conversación es controlada por el atacante. Esto no es posible en nuestra propuesta porque tanto el servidor como la etiqueta detectarían el ataque debido a que el atacante ignora los datos de la etiqueta, que solo conoce el servidor. Por lo tanto, si el servidor detecta un ataque, informa al lector que el mensaje recibido en el paso 2 no produce ninguna colisión. Si con una probabilidad despreciable el servidor encuentra una colisión al azar después de un ataque MitM, entonces la etiqueta detecta el ataque cuando el mensaje recibido en el paso 5 no se corresponde con los datos correctos. Por lo tanto, se puede concluir que el esquema propuesto de autenticación mutua es inmune a los ataques MitM.

En conclusión, la propuesta de autenticación mutua lector-etiqueta se puede considerar suficientemente segura como para usarla con el objetivo planteado de gestionar MANETs mediante el control de su topología adhiriendo etiquetas a los nodos móviles y añadiendo lectores a la red.

Capítulo 3

Redes Ad-hoc Vehiculares (VANETs)

En una VANET, los mensajes intercambiados entre los vehículos influyen en el comportamiento de sus conductores, reduciendo su velocidad y/o escogiendo rutas alternativas en función de la información recibida acerca de la condiciones de la carretera. Posibles usuarios malintencionados podrían intentar explotar esta situación, llevando a cabo alguno de los siguientes ataques:

1. Inyección de información falsa, modificada o repetida, intentando difundir datos erróneos que puedan afectar al resto de vehículos, para el beneficio del atacante, por ejemplo al conseguir liberar una vía, o simplemente por mala intención con el objetivo de producir un atasco.

2. Falsificación de identidad, haciéndose pasar por ejemplo por un vehículo de emergencia.

3. Manipulación de la información enviada, alterando datos como posición, dirección, velocidad, etc., por ejemplo, para intentar escapar de responsabilidades al haber provocado un accidente.

4. Seguimiento de conductores y/o vehículos, amenazando su privacidad y anonimato.

5. Denegación de servicio, provocando la pérdida de la conectividad de la red.

Por lo tanto, la seguridad de las comunicaciones es un factor imprescindible a la hora de impedir dichas amenazas y posibilitar el despliegue de las VANETs. La creación de una VANET sin necesidad de añadir dispositivos adicionales, ni a los vehículos ni a la carretera se puede considerar como un reto añadido.

En este capítulo abordamos el problema de la seguridad de las comunicaciones en VANETs en dos aspectos en concreto: la autenticación de nodos de la red y la gestión segura de grupos o clústers de vehículos.

3.1. Estado del Arte

Para la redacción de este capítulo se ha estudiado la evolución en el tiempo de la investigación en VANETs. Los documentos consultados proponen distintos puntos de vista sobre el tema. Antes de pasar a comentar algunas de las propuestas concretas para este tipo de redes, vamos a contextualizar históricamente el tema.

Viendo el panorama creciente y la evolución que tendrán los sistemas de transporte, es indudable que se trata de una de las áreas con mayor crecimiento y en la que los países desarrollados están más interesados. Por ello, la investigación relacionada con el transporte toma una gran relevancia ya que se enfrenta a problemas asociados a la movilidad, a la seguridad y a la calidad de vida de los usuarios.

Dichos problemas presentan niveles críticos y seguirán aumentando si no se toman las medidas adecuadas. Desde hace algún tiempo dichos problemas han empezado a abordarse mediante las Tecnologías de la Información y las Comunicaciones (TIC), lo que ha permitido proponer soluciones para mitigarlos. En este sentido, la Comisión Europea afirmó que las nuevas tecnologías son un elemento fundamental para el apoyo de los sistemas de transporte [199].

A continuación se recogen algunos de los antecedentes históricos de mayor relevancia que destacan a las TIC como elemento fundamental para el desarrollo de los sistemas de transporte.

Las nuevas tecnologías y en particular las TICs, son partícipes de muchísimos cambios sociales y en el sector del transporte, que de manera paulatina, las ha venido incorporando para su desarrollo. Esto se puede demostrar observando cómo las distintas sociedades han ido involucrándolas para mejorar los sistemas de transporte y para mejorar la seguridad de sus habitantes. Por ejemplo, desde los años setenta en Asia, y más específicamente en Japón, comenzaron a introducirse en el desarrollo del proyecto CACS (Comprehensive Automobile Traffic Control System), que se fundamentaba en el despliegue de un sistema de información avanzado que pudiera ayudar en la congestión del tráfico, seguridad y polución [196]. Siguiendo el mismo objetivo, otro de los proyectos que involucraban a las TICs fue el RACS (Road/Automobile Communication system) [197], que fomentaba la realización de un sistema de comunicaciones entre los vehículos y unidades de carretera.

Asia no fue el único continente donde se empezaron a utilizar las TIC para los sistemas de transporte sino que en América, y particularmente en Estados Unidos también, comenzaron a desarrollar un sistema electrónico de orientación que tenía como objetivo guiar a los conductores para aliviar la congestión urbana [176].

Algunos años después, en 1988, uno de los hitos determinantes en la penetración de las TIC en los sistemas de transporte ocurrió cuando la administración federal de autopistas de los EE.UU. [87], creó un grupo de investigación para desarrollar el concepto de Sistema de Información Avanzado para el Conductor o por sus siglas ADIS (Advanced Driver Information System), que investigaría mejoras para los sistemas de transporte. Para que las TIC fuesen una realidad, se realizó un llamamiento a los diversos actores de la industria, la academia y el gobierno. Al hacerlo se estableció el proyecto Mobility 2000 en el año 1989 [57]. Dicho proyecto fue el primero del área asociada a los Sistemas Inteligentes de Automóviles y Autopistas IHVS (Intelligent Highway Vehicle Systems) donde las TIC eran el elemento fundamental y transversal.

La consolidación final del área ocurrió en el año 1991 cuando el departamento de transporte de Estados Unidos cambió el nombre IHVS por el de ITS América (Intelligent Transportation Society of America), que posteriormente, pasaría a ser conocido como Sistemas Inteligentes de Transporte o ITS (Intelligent Transportation System) [114].

En esa misma década, en Europa también se empezaron a introducir paulatinamen-

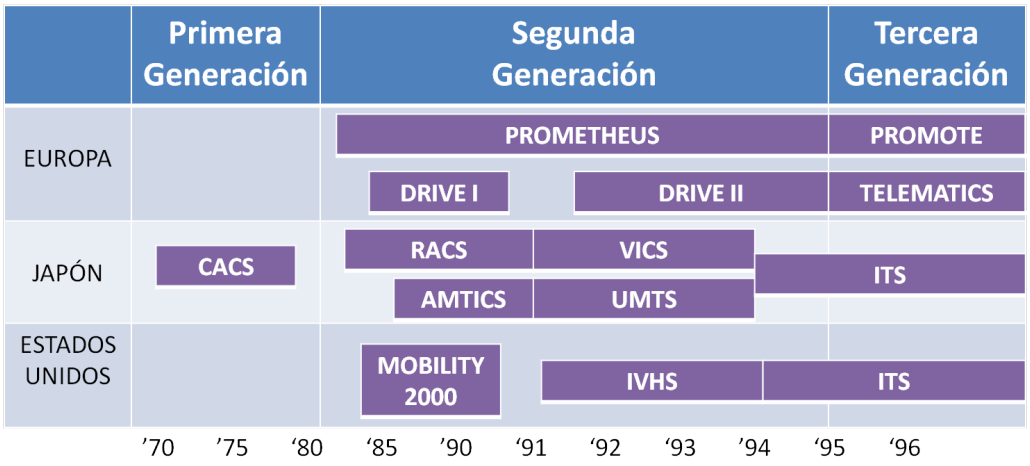


Figura 3.1: Evolución de los ITS en todo el Mundo

te las TIC en los sistemas de transporte a partir de los trabajos de cooperación tecnológica abordados en el proyecto EUREKA [190]. En este sentido, el proyecto PROMETHEUS fue el primero en el año 1989, que tuvo como fin proveer a Europa de un programa para la gestión del tráfico con eficiencia y seguridad [210]. Siguiendo este comienzo, en los años 90 se desarrolló el programa DRIVE enfocado a mejorar la seguridad en la carretera, aumentar la eficiencia y disminuir la polución ambiental [58]. Para ello las TIC se convirtieron en un elemento imprescindible en el desarrollo de los sistemas de transporte pero a la vez, esto requirió que existiese un organismo activo que siguiera fomentando dicho crecimiento. Por lo tanto, en noviembre de 1991 se creó ERTICO (European Road Transport Telematics Implementation Coordination Organization), organización del sector público y privado dedicada a mejorar la infraestructura de transporte, conocida actualmente como ITS Europa, que es el mayor representante de los ITS a nivel europeo [82].

Todo lo descrito aquí forma parte del inicio de las TIC en los sistemas de transporte desde los años 70s hasta los 90s, y sirve para esclarecer ligeramente el contexto histórico. La Fig. 3.1 ilustra una línea de tiempo donde se muestran los diversos proyectos creados a nivel mundial durante ese periodo.

Como consecuencia, a partir del impacto ocasionado por el desarrollo por los proyectos mencionados, los gobiernos han decidido seguir apoyando las propuestas relacionadas

con el área de los ITS dado que resulta importantísimo para el desarrollo económico de las naciones. Por ello, la Unión Europea estableció un plan de acción para el despliegue de los ITS [81], que sigue vigente, y del que ya se están obteniendo algunos frutos, como lograr establecer una política de integración general para la convergencia de servicios en los estados miembro, pero del que aún faltan muchísimas cosas por resolver.

Los ITS se refieren a una gran variedad de herramientas y conceptos relacionados con las áreas de ingeniería, software, hardware y tecnologías de comunicaciones, aplicadas de forma integrada a los sistemas de transporte para mejorar su eficiencia [61], para encontrar soluciones a problemas relacionados con el transporte, como proteger el medio ambiente y generar sostenibilidad, y sobre todo, para mejorar la seguridad vial protegiendo la vida humana [89]. De hecho, durante los últimos 10 años, los ITS se han hecho cada vez más importantes debido a los graves problemas asociados a la movilidad, congestión de tráfico, impacto medioambiental, aumento de muertes en las carreteras, gestión de las infraestructuras, despliegue de servicios, heterogeneidad de tecnologías desplegadas, etc. [3], [107].

Algunos parámetros estadísticos a nivel mundial muestran la necesidad de nuevos y mejores sistemas ITS que ayuden a solucionarlos. Por ejemplo, si observamos el sector automovilístico en la Unión Europea, los costes de los atascos ascendieron hasta el 1 % de su PIB representando, 100 millones de euros por año. Esta cifra puede seguir creciendo si no se aplican los sistemas y modelos adecuados [78]. De la misma forma, y trasladándonos al sector automovilístico de los Estados Unidos, la congestión urbana asciende a 4.200 millones de horas de retraso, generando un consumo adicional de 2.900 millones de galones de combustible, lo que implica un coste de 80.000 millones de dólares al año [204].

Por otra parte, con respecto al problema del impacto medioambiental producido por el crecimiento de la movilidad, donde uno de los mayores responsables es el transporte por carretera debido a la quema de combustibles fósiles [216], en Europa, los índices de CO_2 se han incrementado en un 32 % entre el año 1990 y 2004 [79]. En el caso estadounidense, el 42 % de las emisiones de CO_2 son a causa de la utilización del petróleo [74], concretamente del empleado por el transporte por carretera. Partiendo de este punto, los gobiernos, los operadores de infraestructuras y las autoridades públicas recurren a diversas soluciones tec-

nológicas que de alguna manera u otra forma alivien y mejoren los problemas mencionados y algunos otros problemas de gran relevancia que no han sido mencionados. En este sentido, la Comisión Europea fomenta las investigaciones en los ITS, empleando para ello grandes sumas de dinero que ascienden hasta los 300 millones de euros en 2009 [199].

Algunas de las investigaciones relacionadas con este área son impulsadas a través de los distintos programas marco. Por ejemplo, la UE apoya proyectos tales como [65], [67], [83], [92], [113], [115], [138] [157], [160], [161], [169], [177], y [183], que son llevados a cabo para mejorar las infraestructuras de transporte y sus servicios. A continuación, describimos brevemente algunos de los proyectos referenciados.

- **COOPERS**

  Se enfoca hacia los sistemas cooperativos para la seguridad vial inteligente y específicamente se centra en el desarrollo de aplicaciones telemáticas innovadoras en vehículo e infraestructura. Con su despliegue se busca reducir la brecha digital de la evolución de las aplicaciones telemáticas en la industria del automóvil y la de los operadores. Por lo tanto, pretende mejorar la seguridad vial gracias a la utilización de distintos mecanismos que provean información de tráfico.

- **CVIS**

  Su objetivo es diseñar, desarrollar y probar nuevas tecnologías que permitan a los vehículos comunicarse entre ellos y con las unidades de carretera para minimizar los problemas relacionados con la movilidad y seguridad de los viajeros. Para lograrlo, desarrollan diversos servicios ITS que benefician a los conductores, operadores de infraestructura, empresas de ITS y otras partes interesadas.

- **EVITA**

  Pretende hacer frente a las amenazas a las comunicaciones coche-a-coche y coche-a-infraestructura mediante la prevención de la manipulación no autorizada de las unidades a bordo con el fin de prevenir eficazmente la intrusión en los sistemas en los vehículos y la transmisión de datos corruptos hacia el exterior.

- **GEONET**

Su objetivo principal es implementar una especificación de referencia que permita intercambiar mensajes de seguridad vial entre los vehículos, y a diferencia de los demás proyectos, pretende desarrollar un sistema de direccionamiento geográfico y un protocolo de enrutamiento que sea capaz de soportar el protocolo estándar de Internet versión 6 IPV6.

- **ITETRIS**

  Se dedica al desarrollo de herramientas avanzadas de simulación de tráfico y de comunicaciones inalámbricas para permitir el desarrollo de protocolos y algoritmos adecuados para las VANETs y su análisis a gran escala.

- **IVWSN**

  El primer objetivo de este proyecto es proporcionar un modelo matemático del canal de propagación de radio frecuencia dentro del vehículo, crucial para el diseño de un sistema de comunicación robusto mediante la construcción de un montaje experimental en un coche real.

- **OFAV**

  Su meta es el desarrollo de una arquitectura abierta para los futuros vehículos autónomos, que se convierta en una plataforma estándar compartida por los fabricantes de automóviles para el diseño de los vehículos inteligentes de próxima generación.

- **OVERSEE**

  Desarrolla una plataforma tecnológica vehicular abierta para proporcionar un entorno protegido estándar en los vehículos con el objetivo de garantizar la seguridad de las TIC y la funcionalidad necesaria para que el vehículo no pueda ser dañado por ninguna aplicación.

- **POWERUP**

  Tiene como objetivo desarrollar una interfaz Vehicle-TO-Grid (V2G), y requiere para lograrlo que cualquier tipo de vehículo eléctrico sea compatible con cualquier red europea smart-grid.

- **SAFESPOT**

  Su fin es el diseño de sistemas cooperativos para mejorar la seguridad en las carreteras haciendo especial énfasis en las comunicaciones entre los mismos vehículos y entre los vehículos y la infraestructura. Busca también disminuir la tasa de accidentes de tráfico ofreciendo la detección con antelación de situaciones potencialmente peligrosas para el usuario.

Cada uno de los resultados de los proyectos descritos, están siendo integrados y consolidados a través del foro COMeSafety [64], que a su vez sirve de plataforma para el intercambio de información y el planteamiento de estándares ITS.

En el plan europeo de acción en ITS [80] se ha subrayado que el despliegue de servicios ITS es un factor clave para alcanzar no sólo mayor seguridad en las carreteras sino también, un crecimiento económico-social para los estados miembro. El departamento de transporte de los Estados Unidos a través de la agencia de investigación para la innovación tecnológica incide exactamente en lo mismo, concluyendo que dicho despliegue es de gran preocupación para el desarrollo económico no sólo de algunas naciones sino también del sector de transporte en general [174], [205].

El documento [68] confirma que uno de los factores determinantes para el despliegue de servicios de valor añadido en el contexto de los ITS ha sido el impacto que han tenido las TIC en las diferentes áreas del transporte. En este mismo sentido, [187] destaca que uno de los temas que más atención tendrán a nivel científico por parte de la comunidad de los ITS son los servicios de valor añadido, proporcionados a los conductores. Además, la Comisión Europea ha identificado que el despliegue coherente de los ITS provoca la creación de una gran variedad de servicios que si se gestionan y se ofrecen de forma adecuada permitirán el crecimiento económico de la UE, y al mismo tiempo traerán beneficios sociales a corto y medio plazo [199].

En este contexto, existe una gran cantidad de elementos tecnológicos que están teniendo una alta repercusión en el crecimiento y despliegue de los servicios de valor añadido. Así, es importante destacar el papel desempeñado por los sistemas de navegación global por satélite o GNSS (Global Navigation Satellite Systems) dado que han proporcionado a los ITS

una gran herramienta tecnológica para suministrar información o servicios a sus usuarios mientras estos se desplazan de un lugar a otro [221].

Además, dichos sistemas, según la autoridad europea de los GNSS, están aportando al mercado más de 40.000 millones de dólares por año desde el año 2006 y esto tiende a incrementarse hasta superar los 90.000 millones de dólares a finales del 2011 [97], demostrando así el crecimiento económico no sólo de este sector sino también de los servicios relacionados con la navegación. Los servicios de valor añadido comienzan a ser parte de grandes proyectos de la comunidad ITS, y muestra de ello es la creación reciente de la Asociación de Servicios para la Información al Viajero o TISA (Traveller Information Services Association) [201], que busca generar mecanismos coherentes para su despliegue.

Paralelamente, el consumo de servicios de valor añadido en el ámbito de los ITS empieza a ocupar una gran cuota de mercado y para confirmarlo sólo hay que analizarlo, la autoridad supervisora de los GNSS estima que para el año 2020 habrán 3.000 millones de dispositivos de navegación [98]. Otro de los factores culminantes que benefician el crecimiento de tales servicios es la cantidad de dispositivos móviles en Europa. Según el líder en la producción de mapas para la navegación NAVTEQ, 35,8 millones de dispositivos con capacidades de sistema de navegación fueron vendidos en el año 2008, de los cuales, 3 millones fueron instalados previamente en el automóvil [4]. Ante tal hecho, los operadores móviles en la actualidad están respondiendo mediante fuertes inversiones para desarrollar nuevas tecnologías que hagan viable la prestación de servicios. Incluso se ha comenzado a analizar la alta demanda que van a desatar tales servicios en el sector del transporte. Por ejemplo, según la consultora de mercado BERG INSIGHT, los suscriptores de servicios de navegación tienen un crecimiento proyectado del 42 % en Europa y del 30 % en Norteamérica para el año 2015 [10]. Al ver estos índices, la misma consultora concluyó que para el año 2014 habrá un mercado potencial de 960 millones de dispositivos con capacidades de sistemas de navegación, los cuales pueden ser explotados mediante servicios asociados a la navegación [11].

La investigación realizada en este trabajo en el entorno de las VANETs tiene dos focos principales, por un lado se describe un esquema de autenticación de nodos distribuido, especialmente diseñado para su funcionamiento sin necesidad de soportes externos tales

como autoridades centralizadas o unidades de carretera. Por otro lado se describe una arquitectura para gestionar grupos de nodos en VANETs, aquí denominados clústers, que tiene como objetivo reducir las comunicaciones realizadas en situaciones de tráfico denso, donde se genera una gran cantidad de paquetes de información en un área pequeña, lo que degrada la calidad de las comunicaciones.

Con respecto al requerimiento de minimización de las comunicaciones en los esquemas de autenticación de nodos en VANETs, varios trabajos se centran en diferentes aspectos y aplicaciones. [162] propone un sistema de notificación de estacionamiento que no necesita una gran infraestructura, pero requiere RSUs en los estacionamientos para dar soporte al servicio. [191] propone un esquema de gestión de claves para VANETs, que se utiliza para autenticar los mensajes, identificar los vehículos legítimos y eliminar de la red a los vehículos maliciosos. Sin embargo, dicha propuesta se basa en el uso de una infraestructura de clave pública. [218] se centra en la comunicación vehicular descentralizada sin hacer uso de ningún tipo de infraestructura fija y propone un método para la creación dinámica de comunicaciones seguras en VANETs.

Hay otras referencias bibliográficas que proponen diferentes tipos de esquemas de autenticación de nodos en VANETs autogestionadas, basados en métodos que son totalmente diferentes a los que aquí se presentan. [49] propone un esquema de autenticación que se basa en pseudónimos, mientras que [128] describe un esquema que combina la autenticación, el establecimiento de claves y técnicas de firma ciega. Con respecto a la certificación de claves públicas, [125] presenta un método para la revocación de certificados sobre la base de la distribución epidémica vehículo a vehículo, y [101] propone otro mecanismo de revocación de certificados de seguridad, que necesita una autoridad de certificación y listas de revocación de certificados.

Con respecto al uso de clústers, muchas referencias bibliográficas proponen su uso tanto en MANETs como en VANETs y con diferentes objetivos, tales como la difusión de información, la agregación de datos, las firmas de grupo, la minimización de la sobrecarga en la red, el encaminamiento, etc. Propuestas de clústers en MANETs fueron estudiadas en [48], [94] y [193], pero ninguno de estos trabajos aborda el uso de clústers en VANETs. Por otro lado, los autores de [86] presentan un análisis teórico de clústers en VANETs.

Un problema frecuente en la bibliografía es la selección del líder del clúster ya que la mayoría de los enfoques de clústers para VANETs consideran la existencia de un nodo especial a cargo de la administración del clúster. Dos técnicas simples para la selección del líder se describen en [100] y [179], basadas en la identificación del menor ID o el uso de beacons. La primera técnica no optimiza ninguna característica de la red. La segunda utiliza la retransmisión regular de beacons que anuncian al estado del nodo para que cada nodo pueda conocer el estado de sus vecinos. Este último enfoque trata de minimizar los cambios de clúster, especialmente en el caso de clústers más grandes debido a que un líder sólo tiene en cuenta un cambio de su estado si recibe un mensaje de otro líder de un clúster con más nodos. Sin embargo, este simple criterio no tiene en cuenta factores tales como si los clústers se están moviendo en direcciones opuestas. [76] presenta un algoritmo llamado CASAN para la selección del líder que tiene en cuenta el nivel de confianza de los nodos, por lo que necesita controlar la reputación del resto de nodos. En [12] el líder es seleccionado de acuerdo con la información de la movilidad y las intenciones del conductor.

El trabajo [85] propone un enfoque mixto basado en una función de utilidad que utiliza como parámetros tanto la ubicación más cercana a la media como la velocidad más cercana a la media. Esta propuesta no se adapta a la dinámica del tráfico porque por ejemplo, el intervalo para la formación de clústers es fijo. En nuestra propuesta los factores decisivos para la selección del líder pueden ser vistos como una combinación de los criterios antes mencionados, junto con la consideración de otros parámetros, como la formación de clústers como un proceso continuo.

En cuanto a la formación de clústers en VANET, se pueden encontrar muchos algoritmos diferentes en la bibliografía. En el trabajo [178] se describe CARAVAN, en el cual los clústers están formados por los vehículos que pueden recibir las retransmisiones del resto de miembros del clúster, con el objetivo de reducir el número de pseudónimos para proteger la privacidad. En el trabajo [135] se proponen clústers estáticos en entornos urbanos con el fin de reducir el efecto de atenuación de la señal debido a los obstáculos físicos. Un algoritmo para clústers se ha descrito en [141] donde el líder es el primer vehículo que entra a formar parte del clúster. En el documento [185] se presenta un esquema de clústers basado en la movilidad de las VANETs, que intenta producir clústers con alta estabilidad para facilitar el

encaminamiento. El trabajo [192] propone un sistema multi-canal de comunicaciones basado en clústers para aumentar la calidad del servicio mediante la reducción de la congestión de datos.

Diferentes aplicaciones de clústers en VANET han sido consideradas en la bibliografía. Por ejemplo, el trabajo [110] propone una extensión de seguridad de un esquema de agregación en VANETs que organiza clústers de vehículos para distribuir los datos agregados, lo que al mismo tiempo permite la detección de vehículos maliciosos. Los clústers se han propuesto en [171] para maximizar la retransmisión de la información y la notificación rápida evitando interferencias. Sin embargo, en ese esquema el líder debe conocer la posición exacta de los nodos del clúster, lo que produce unos gastos de gestión considerables.

Después de un análisis de estas y otras referencias bibliográficas sobre clústers en VANET, llegamos a la conclusión de que, en general, los diferentes trabajos no definen en detalle los procesos que los nodos tienen que realizar para la gestión del clúster, ni tampoco muestran esquemas implementados que estudien el comportamiento de las propuestas y demuestren su fiabilidad. Estos son los objetivos tratados en la tercera sección de este capítulo, que se centra en minimizar la sobrecarga producida por la difusión de datos, y proporciona una manera de crear una clave secreta compartida para el uso de cifrado simétrico dentro del clúster.

3.2. Autenticación de Nodos

Esta sección describe una nueva solución a la autenticación de nodos para el despliegue práctico, rápido y seguro de las redes vehiculares. La principal contribución es un método de autenticación autogestionado, que no requiere la participación de ninguna entidad de certificación porque los propios vehículos certifican la validez de las claves públicas de los vehículos en los que confían, y emiten los correspondientes certificados que se guardan en los almacenes de claves locales de cada vehículo de acuerdo a un algoritmo aquí propuesto. Además, el nuevo método de autenticación de nodos incluye un protocolo criptográfico que cada vehículo puede usar para convencer a otro de la posesión de cierto secreto sin revelar nada al respecto. Uno de los aspectos más interesantes de la propuesta es que los dispositivos

requeridos en los vehículos pueden ser simples dispositivos como teléfonos móviles equipados con conexión inalámbrica tal como veremos en el capítulo siguiente de implementación en dispositivos reales. Esta sección incluye un análisis del rendimiento usando simulaciones de los algoritmos propuestos.

3.2.1. Introducción

La propuesta aquí presentada tiene como punto de partida la consideración de que la introducción de un modelo completo de VANETs incluyendo unidades en carretera o RSUs y unidades de a bordo u OBUs sería extremadamente caro, tanto para los usuarios, que tendrían que comprar un vehículo nuevo o instalar dispositivos específicos en sus vehículos, como para el Estado, que tendría que instalar una infraestructura enorme para apoyar los servicios de VANETs. Por lo tanto, este trabajo propone una VANET autogestionada, que no requiera ningún tipo de infraestructura.

El esquema de control de acceso incluido en la propuesta de autenticación de nodos descrito a continuación se basa en el esquema general de ZKP usado en [44] y basado en el problema del grafo isomorfo, para el caso particular del problema del circuito hamiltoniano o HCP.

3.2.2. Caracterización de Nodos y Beacons

En la propuesta presentada a continuación se asume que cada vehículo de la red se caracteriza por los siguientes parámetros:

$ID, (KU_{ID}, KR_{ID}), (ID_i, KU_{ID_i}, Cert(KU_{ID_i}))_{ID_i \in KS_{ID}}$

incluyendo:

- Un identificador único (denominado ID), obtenido como resultado de una función de un sólo sentido sobre un valor único. Por ejemplo, si el dispositivo utilizado es un teléfono móvil el valor puede ser el número de teléfono, mientras que en otros casos se puede utilizar una dirección de correo electrónico. La función de un solo sentido podría ser cualquier función hash.

- Un par de claves pública/privada fijas (denotado (KU_{ID}, KR_{ID})) asociadas a ID, que se utilizan en un criptosistema asimétrico, como RSA.

- Un almacén de claves KS_{ID} asociado a ID, que contiene varias identificaciones y las correspondientes claves públicas y certificados, que el nodo ID mantiene siempre actualizado, de la forma:

ID1	KU_{ID1}	$Cert(KU_{ID1})$	
ID2	KU_{ID2}	$Cert(KU_{ID2})$	
ID3	KU_{ID3}	$Cert(KU_{ID3})$	
.	.	.	
.	.	.	
.	.	.	
ID_{lim}	$KU_{ID_{lim}}$	$Cert(KU_{ID_{lim}})$	

El envío de beacons de multidifusión conteniendo identificadores variables del remitente se requiere tanto para el proceso de descubrimiento de nodos activos como para evitar el rastreo de vehículos. En el mismo paso en el que se envían los beacons, cada nodo se compromete con su secreto enviando a sus vecinos también un testimonio de su secreto. El identificador variable de cada nodo, que es enviado como parte de su beacon, es el hash de los IDs que están presentes en su almacén de claves en cada momento.

En particular, los beacons enviados por un nodo están formados por los siguientes elementos:

- Marco de Control (FC, Frame Control), que indica el tipo de datos que se envían.

- Pseudónimo (Pseu), que es un identificador variable del nodo.

- Marca de tiempo (Tiempo), que permite conocer la hora específica en la que se generó la información.

- Par formado por la clave pública del nodo y la marca de tiempo (KU, Tiempo) cifrado con la clave privada (KR) del nodo, que es utilizado por los nodos que ya lo han autenticado, cuando se cambia el Pseu.

3.2.3. Generación y Certificación de Claves Pública/Privada

Dentro de esta propuesta, el dispositivo asociado a cada vehículo de la red debe ser capaz de generar su par de claves pública/privada y también de firmar las claves públicas de otros vehículos que deseen formar parte de la red y sean confiables.

Con el fin de facilitar la implementación de la ZKP para el circuito hamiltoniano en el proceso de autenticación de vehículos descrito a continuación, se calcula la clave pública de cada vehículo KU_{ID} a partir del valor decimal de la representación binaria de la submatriz triangular superior de la matriz de adyacencia simétrica que contiene los elementos correspondientes al circuito hamiltoniano en un grafo (ver Fig. 3.2).

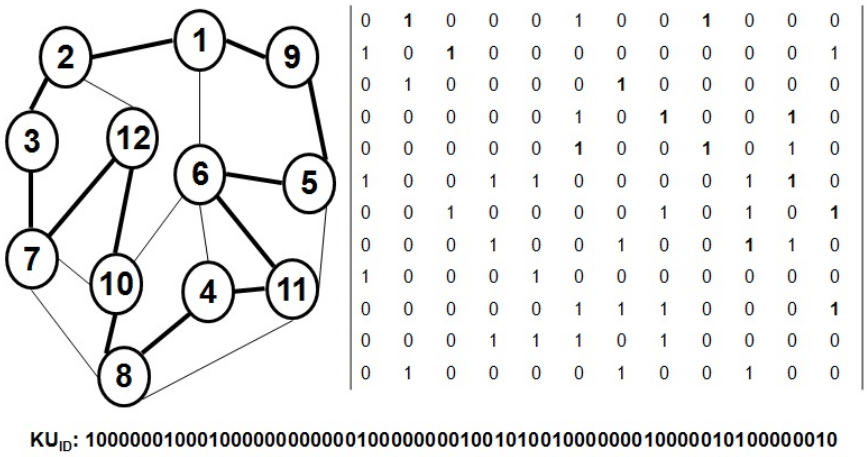


Figura 3.2: Ejemplo de Clave Pública Basada en Circuito Hamiltoniano

El número decimal que corresponde a la representación binaria se utiliza en la propuesta como el exponente de la clave pública en el cifrado RSA usado por el dispositivo para cifrar y descifrar mensajes y firmar certificados de clave pública.

En la Fig. 3.3 podemos ver una traza de una elección de un par de claves pública/privada mediante el circuito hamiltoniano para la generación de la clave pública. Después de elegir los números primos p y q, el exponente público e se genera a partir de un circuito hamiltoniano al azar, por lo que es menor que y coprimo con $(p-1)(q-1)$. Posteriormente se genera el exponente privado.

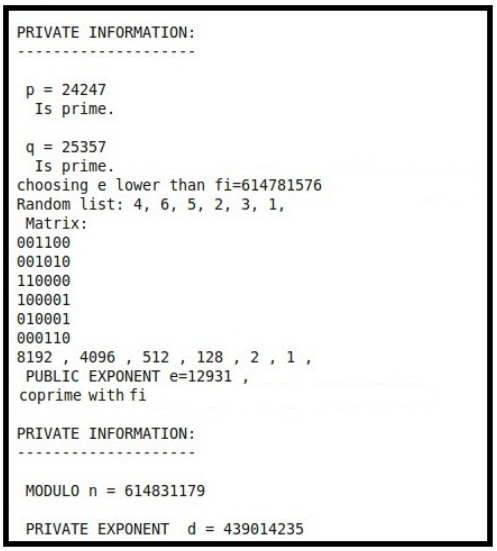


Figura 3.3: Implementación del RSA con Clave Basada en el HCP

Con el fin de ser capaz de autenticar su clave pública, todos los nodos deben intercambiar las firmas con un número de nodos legítimos de la red, que depende del tamaño de la VANET. Al principio, en la implementación real realizada y expuesta en el siguiente capítulo estimamos que dos firmas serán suficientes para probar que el usuario es confiable y que no puede auto-firmarse los certificados para comprometer la seguridad de la red, pero en general el número de firmas requeridas debe crecer a la vez que la VANET crece.

La certificación auto-organizada de las claves públicas hace posible autenticar la clave pública de un nodo sin conocerlo y sin la necesidad de ninguna tercera parte de confianza. Dicha certificación se basa en la confianza de los vecinos de sus vecinos mediante la formación de un grafo certificado. En este trabajo se entiende que un certificado entre A y B es siempre bidireccional ya que consiste siempre en dos firmas: la firma de la clave pública de B con la clave privada de A, y viceversa.

Cuando un vehículo A quiere comprobar la validez de la clave pública de otro vehículo B, A tiene que encontrar una cadena de certificados desde A hasta B en el grafo certificado que resulta de la fusión de los subgrafos G_A y G_B correspondientes a KS_A y KS_B respectivamente.

3.2.4. Esquema Basado en ZKP

Los nodos se intercambian paquetes especiales para autenticarse mutuamente. Entre otros datos, dichos paquetes contienen los datos FC, Pseu del nodo origen y Pseu del nodo destino. La Fig. 3.4 muestra de forma esquemática las tres fases de interacción incluidas en el protocolo auto-gestionado propuesto para la autenticación entre dos nodos A y B.

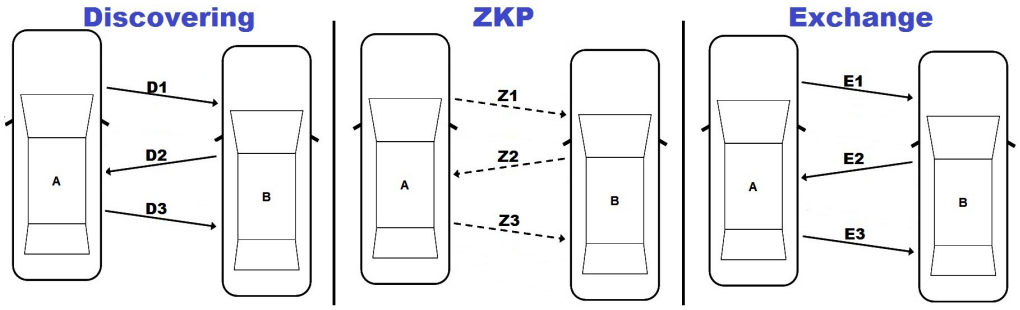


Figura 3.4: Protocolo de Autenticación Propuesto

Las tres fases se describen con detalle a continuación. La primera fase es el proceso de descubrimiento, que incluye parte de los beacons enviados por el nodo A, y en particular los resúmenes de los identificadores almacenados en KS_A. En esa fase de descubrimiento, si un nodo B quiere comunicarse con otro nodo A, B comprueba si tienen alguna clave en común X que pueda ser encontrada en la intersección de sus almacenes de claves, en cuyo caso genera un grafo $G_B(X)$ en el que X es solución al HCP, y lo envía a A junto con los resúmenes de los ID de su almacén de claves. Si A confirma la existencia de X, devuelve a B también un grafo $G_A(X)$. Entonces ambos ejecutan en la segunda fase una ZKP mutua sobre el conocimiento de X. Tras ella en la última fase ambos nodos usan X para intercambiarse sus claves públicas y establecer una clave secreta compartida que usan para intercambiarse sus almacenes de claves.

Algoritmo Esquema de Autenticación

función $EsquemaAutenticacion()$ (...)

D1. $A \rightarrow B$: beacon con $\{h(ID_i) : ID_i \in KS_A\}$

D2. $B \rightarrow A$: $\{h(ID_i) : ID_i \in KS_B\}$ y un grafo $G_B(X)$, si $\exists X \in KS_A \cap KS_B$

D3. $A \rightarrow B$: un grafo $G_A(X)$ si $\exists X \in KS_A \cap KS_B$

Z1. $A \rightarrow B(B \rightarrow A)$: un grafo $GI_A(X)$ $(GI_B(X))$ isomorfo con $G_A(X)$ $(G_B(X))$

Z2. $B \rightarrow A(A \rightarrow B)$: un reto binario aleatorio b (a)

Z3. $A \rightarrow B(B \rightarrow A)$: si $b = 0$ $(a = 0)$ $GI_A(X) \approx G_A(X)$ $(GI_B(X) \approx G_B(X))$

Z3. en otro caso un circuito hamiltoniano en $GI_A(X)$ $(GI_B(X))$

E1. $A \rightarrow B(B \rightarrow A)$: E_X $(KU_A)(E_X(KU_B))$

E2. $B \rightarrow A(A \rightarrow B)$: $KU_A(K_B)$ $(KU_B(K_A))$

E3. $A \rightarrow B(B \rightarrow A)$: $E_{KB}(KS_A)$ $(E_{KA}(KS_B))$

fin función

El algoritmo anterior permite que dos nodos se autentiquen uno frente al otro, así como que establezcan una clave secreta compartida e intercambien sus almacenes de claves.

La Fig. 3.5 muestra una ejecución en el esquema de autenticación propuesto e implementado utilizando Microsoft Visual Studio en *C#*.

Un cliente-servidor capaz de múltiples conexiones al mismo tiempo se implementa en cada dispositivo. Todas las señales acerca de la autenticación y beacons se realizan con paquetes UDP. Cada cliente transmite beacons periódicamente a todos los dispositivos conectados en la red. Cada beacon está formado por los siguientes datos:

"01," + thisIpAddr + "," + PSEU + "," + Ek1(ID1,KUid1,TimeStamp)

Antes de comenzar a usar el dispositivo, el nodo necesita información para comunicarse con otros dispositivos, y en particular, una base de datos con tres tablas cargadas. Estas tablas tienen datos para mantener un mínimo número de usuarios necesario para autenticar al resto y cuyos datos (certificados y claves públicas) se generan con el generador mostrado en la Fig. 3.3:

certificateStore (idcolumn INT PRIMARY KEY, idA NTEXT, idB NTEXT, certAB BIGINT, certBA BIGINT, date DATETIME);

KS (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, module BIGINT, publicKey BIGINT, secretKey BIGINT, degree INT);

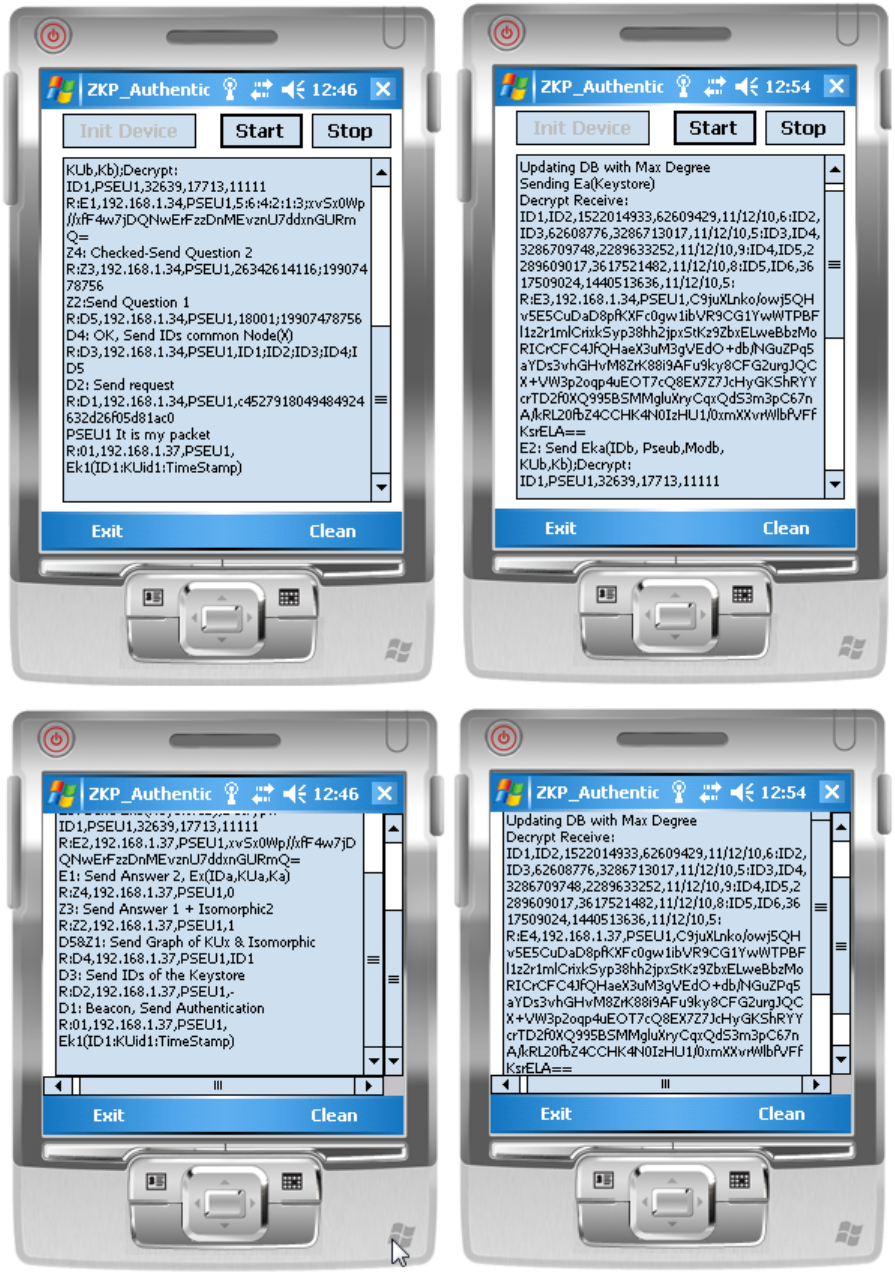


Figura 3.5: Implementación del Esquema de Autenticación Propuesto

myStore (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, modulo BIGINT, publicKey BIGINT, privateKey BIGINT, secretKey BIGINT, degree INT);

Las conexiones entrantes se gestionan en el servidor para que cuando se reciba una, el servidor compruebe la identidad del nodo que envió el paquete. Después de eso, comprueba si el nodo ya está autenticado en la red, y si no, se inicia el protocolo de autenticación. El procedimiento que los nodos usan entonces para enviar y recibir información, se indica en el algoritmo que describe el Esquema de Autenticación.

3.2.5. Actualización del Almacén de Claves

Para las VANETs autogestionadas aquí propuestas se requiere que cada nodo tenga su propio almacén de claves para autenticar otros nodos. En estas redes el número potencial de usuarios es enorme, por lo que se propone un esquema para el almacenamiento de certificados de clave pública que explote el mencionado principio de los seis grados de separación. Gracias a esta propiedad, no es necesario que cada usuario almacene los certificados de todos los nodos para ser capaz de autenticar al resto de nodos de la red. En lugar de eso, sólo necesita almacenar el menor número de certificados tales que mediante la combinación de su almacén de claves con el de cualquier otro usuario con el que desee autenticarse, la probabilidad de encontrar al menos una cadena de certificados en el grafo certificado resultante sea alta.

En consecuencia, la actualización óptima de los almacenes de claves es una parte importante de la propuesta, ya que permite limitar el número de claves almacenadas a un valor aquí denotado *lim*. Tal valor es generalmente menor que el número de usuarios que forman la red, e igual al número mínimo que permite que cualquier nodo pueda conectarse a cualquier otro nodo de la red.

Tal como se vió en el capítulo anterior, se podrían usar diferentes ideas para actualizar los almacenes de claves con el fin de maximizar la probabilidad de que cualquier nodo sea capaz de autenticar a cualquier otro nodo de la red y al mismo tiempo permita limitar el tamaño de los almacenes de claves. Un posible algoritmo basado en una construcción en amplitud de un árbol en el grafo certificado se describe a continuación.

Para actualizar su almacén de claves cada nodo elige los certificados de clave públi-

ca que se corresponden con los nodos que han emitido o recibido más certificados válidos, lo que se representa con los grados de los vértices en el grafo certificado correspondiente. Esta opción maximiza la probabilidad de la intersección entre los almacenes de claves, lo que es necesario para el proceso de autenticación. En particular este algoritmo lo ejecuta un nodo B cada vez que fusiona su almacén con el de otro nodo A, una vez se han autenticado mutuamente gracias a que la intersección entre sus almacenes de claves no es vacía. El objetivo es escoger los mejores certificados para almacenar en el KS_B.

Algoritmo Actualización del Almacén de Claves de B

01: **función** Actualizacion_KeyStore() (...)
02: Inicializa estructura de datos:
03: Union:= $KS_A \cup KS_B$;
04: $KS_B = \{B\}$
05: **para** cada $i \in KS_B$
06: **para** cada $j \notin KS_B : (i,j) \in Union$
07: **si** ((grado $(j) = maximo(grado(vecino\ de\ i\ en\ Union))$)
y $(cardinal(KS_B) < lim)$)
08: $(i,j) \in KS_B$
09: **fin si**
10: **fin para**
11: **fin para**
12: **fin función**

Se ha realizado una implementación inicial del esquema de actualización del almacén de claves propuesto usando la herramienta de simulación Network Simulator NS-2, si bien la simulación del algoritmo en VANETs se presentará en la última sección de este capítulo.

En la simulación preliminar realizada, una red inalámbrica inicial donde los nodos se encuentran colocados al azar (véase Fig. 3.6) produce el primer grafo certificado (véase Fig. 3.7). En este momento, cada nodo guarda en su almacén local de claves los certificados de los nodos que están a distancia 1. Entonces, los nodos comienzan a moverse al azar, y cada vez que dos nodos están a una distancia de 1 salto, verifican si pueden confiar el uno en el otro e inician un intercambio de almacenes para la actualización del almacén de claves.

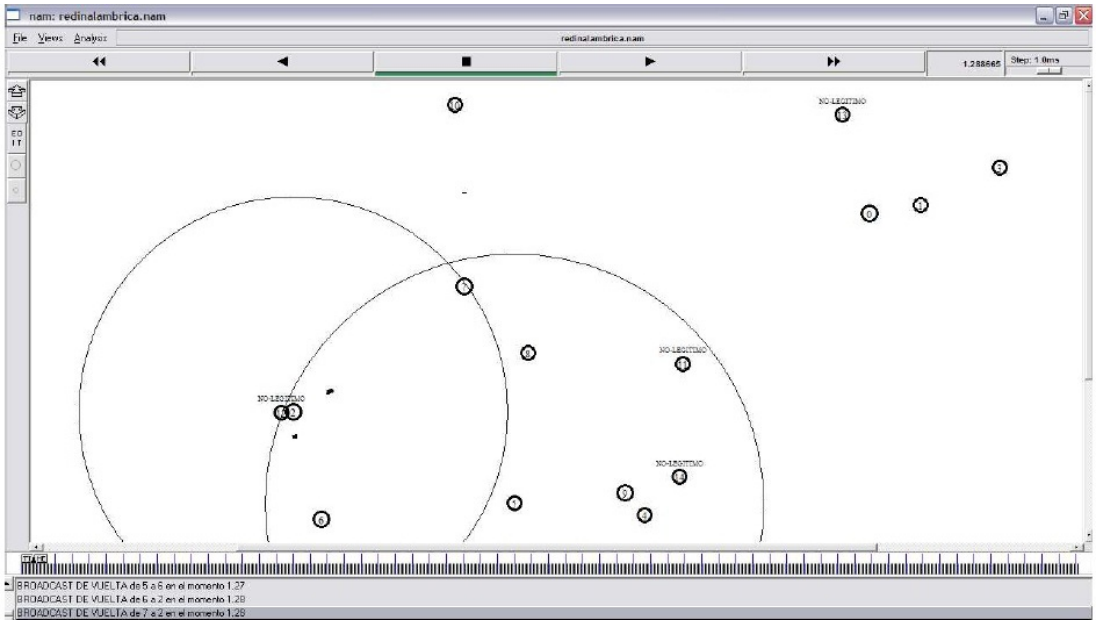


Figura 3.6: Red Inicial

En la Fig. 3.7 vemos que si en ese momento el nodo 8 quiere comunicarse con el nodo 0, no pueden confiar el uno en el otro porque no existe intersección entre sus grafos certificados. Sin embargo, si los nodos 8 y 5 quieren comunicarse con el nodo 4, sí pueden.

En la Fig. 3.8 podemos ver una parte del grafo certificado tras un periodo de vida de la red. En la segunda parte de la figura se observa la actualización del almacén de claves del nodo 10 tras haberse autenticado con el nodo 4. En este ejemplo, el nodo 10 une su almacén de claves con el del nodo 4 y aplicando el algoritmo decide construir su almacén de claves con los certificados (10,7),(7,5),(7,8),(5,1),(5,9),(8,2) y (2,6).

Después de realizar 25 simulaciones con distintos parámetros de movilidad para 20, 30 y 60 nodos, respectivamente, los resultados promedio de las ejecuciones con diferentes tipos de redes mostrados en la siguiente tabla demuestran que el rendimiento puede ser considerado aceptable en general. De acuerdo con las simulaciones también podemos concluir que el esquema se ve afectado por la movilidad de los nodos porque una mayor movilidad conduce a un aumento más rápido y equilibrado de los almacenes de claves. Por lo tanto, este es un argumento convincente para utilizarlo en redes vehiculares, ya que son redes de alta movilidad.

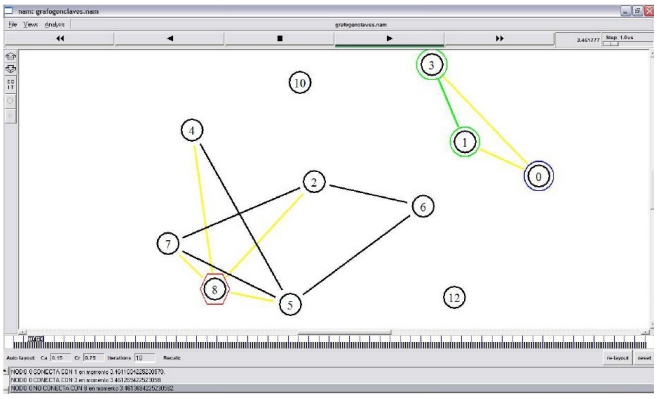


Figura 3.7: Verificación de Confianza

	20 nodos	30 nodos	60 nodos
Nº Paquetes Enviados	1011,52	2764,7	5309,0
Nº Paquetes Recibidos	985,4	2749,8	5216,5
Nº Certificados Añadidos	56,4	80,51	191,9
Nº Actualizaciones de Almacenes	420,2	690,24	1475,9

3.3. Arquitectura de Clústers Auto-Organizados

En esta sección se propone el uso de clústers para reducir la sobrecarga de comunicación que se generan en los escenarios de VANETs donde hay tráfico denso. En particular, se presenta una arquitectura distribuida basada en la agrupación de nodos para crear subconjuntos virtuales en la red, formados por líderes de grupos y nodos pasarela entre grupos, de modo que esos nodos son responsables de la propagación eficiente de los mensajes en la VANET. El principal objetivo de la propuesta es establecer un equilibrio entre la estabilidad de las conexiones de red y la relación coste/eficiencia de la gestión de los clústers. Al mismo tiempo, el uso de clústers facilita la combinación de criptografía de clave pública y de clave secreta, lo que también ayuda a mejorar el rendimiento y la seguridad de las comunicaciones. Los clústers tienen otras ventajas como la escalabilidad de la red y la ampliación del ancho de banda de las comunicaciones. En esta sección se describen las definiciones completas de todos los procedimientos que forman parte de la arquitectura de clústering aquí propuesta, incluyendo el algoritmo de selección de líder del grupo sobre la base de una versión del problema del conjunto independiente de grafos, y un esquema de elección de clave secreta

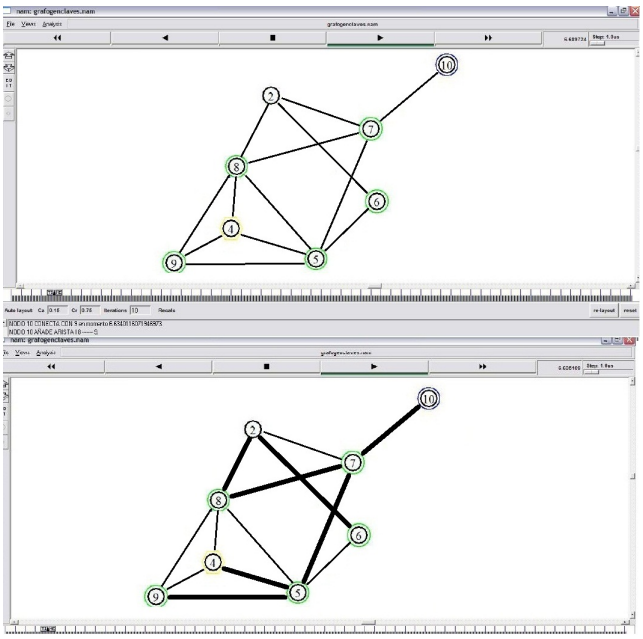


Figura 3.8: Actualización del Almacén del Nodo 10

basado en una generalización del protocolo Diffie-Hellman. Las simulaciones presentadas al final del capítulo muestran que nuestra propuesta mejora el rendimiento de las VANETs, garantizando al mismo tiempo la entrega en tiempo real y la seguridad de los mensajes enviados.

3.3.1. Introducción

La arquitectura propuesta implica que uno de los nodos de cada clúster actúa como Líder del Clúster (CH, Cluster Head), y los nodos que se encuentran en los límites de los clústers reenvían la información a nodos pertenecientes a otros clústers. Además, ya que los vehículos pueden producir información altamente redundante en la VANET, con el fin de evitar este problema conocido como tormentas de difusión, los paquetes idénticos de diferentes fuentes en un clúster pueden ser agregados a través de diferentes funciones como la eliminación de duplicados, la minimización y/o la media y otras funciones que cada nodo puede realizar por sí mismo.

En esta sección se propone un conjunto de protocolos que permiten la formación

de una cadena virtual de vehículos para hacer posible la rápida propagación de mensajes de difusión. La formación de la columna vertebral y la gestión de mensajes se realiza mediante la explotación de algunas características específicas de las VANETs, como es la persistencia de la agrupación en los escenarios más comunes.

Los clústers son aquí definidos como una estructura conceptual donde los vehículos que viajan cerca y en el mismo sentido se auto-organizan en torno a su representante elegido llamado CH. Este nodo especial asume el rol de gerente de las comunicaciones entre clústers y entre los miembros de su clúster, que deben estar dentro de su rango de emisión.

En nuestro esquema el papel de las vías de acceso para las comunicaciones entre-clústers se delegan a otros miembros, en función de su proximidad a otros clústers. Esto se muestra en la Fig. 3.9, donde se identifican cuatro estados básicos de los nodos: líder del clúster (CH), Nodo miembro (MN, Member Node), Nodo enlace (GW, GateWay) y No Definido (ND).

Los clústers son especialmente útiles en condiciones de tráfico denso, cuando el número de vehículos en una zona geográfica cercana es alto, como las horas punta o los atascos de tráfico, ya que en estos casos el número de comunicaciones V2V es mucho mayor. Bajo estas circunstancias, la topología altamente dinámica de las VANET puede perturbar la formación del clúster y su reorganización, aumentando la inestabilidad del clúster. Por lo tanto, los algoritmos de agrupamiento deben ser diseñados para mantener la estructura del clúster lo más estable posible con el fin de proteger el rendimiento de las comunicaciones.

El propósito de los algoritmos propuestos en esta sección es la gestión del clúster, donde el CH está directamente conectado con todos los nodos de su clúster. En particular, los clústers se definen aquí de acuerdo con células dinámicas donde entre otras características, el CH es el nodo con el mayor número de vecinos potenciales dentro del clúster. En particular, la regla de decisión para la selección del CH tiene en cuenta factores tales como la velocidad media, la posición y la dirección de los vehículos. También el estado de los vecinos se tiene en cuenta para que no haya dos CHs que puedan ser vecinos, y cada nodo debe tener al menos un CH dentro de sus vecinos. Por lo tanto, la definición autónoma de clústers implica que los vehículos que circulan en la misma dirección y a la velocidad media del clúster tienen una baja probabilidad de cambiar de clúster durante parte de su recorrido.

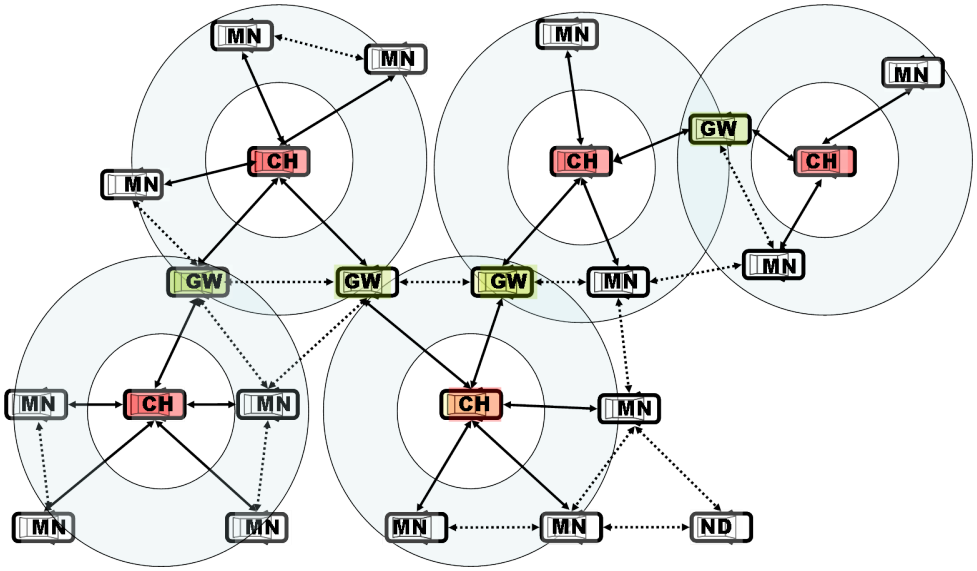


Figura 3.9: Estado Básico de los Nodos

La gestión de los clústers deben cumplir dos requisitos importantes. En primer lugar, debe reducir al mínimo el consumo de recursos y el intercambio de mensajes. En segundo lugar, se debe tener en cuenta la topología y el gran dinamismo de la red. Nuestra propuesta implica una reducción significativa en el número de retransmisiones debidas a la difusión de mensajes. En particular, si n denota el número de nodos en las cercanías de un vehículo, sin el uso de clústers el vehículo envía aproximadamente n paquetes por cada dato recibido y cada vecino recibe la misma información duplicada, la cual tiene que retransmitir de nuevo por lo que el número total de comunicaciones entre los n nodos en el vecindario es de $n(n-1)$. Sin embargo, cuando se utiliza el esquema de clústering aquí propuesto, sólo se generan n conexiones por clúster para cada retransmisión de datos. El primer paquete conecta el nodo miembro que recibe o produce la información por primera vez con el CH de su clúster, entonces el CH envía una difusión a los restantes $n-1$ miembros de su clúster, incluyendo los nodos enlace, que son los responsables de enviar la información a los clústers vecinos.

En nuestra propuesta se supone que los nodos difunden periódicamente mensajes *beacon* que pueden contener la siguiente información acerca del remitente:

$$< Pseu, loc, velocidad, dir, estado >$$

donde

- *Pseu* es un pseudónimo utilizado por el remitente a fin de que los otros nodos puedan detectarlo y enviarle los mensajes, pero protegiendo su anonimato.
- *loc* denota las coordenadas GPS de la ubicación del remitente.
- *velocidad* es la velocidad del remitente.
- *dir* es la dirección del remitente.
- *estado* indica si el remitente es CH, MN, GW o ND.

Las coordenadas GPS de los vecinos permiten comprobar al menos parcialmente la información sobre los vecinos de los vecinos, que se envía durante la fase de creación del clúster y que se explica más adelante. La *velocidad* de los vecinos no sólo se usa para decidir quién será el CH, sino también para excluir a los vehículos cuyas velocidades son atípicas con respecto a las velocidades del resto de vecinos. El parámetro *dir* se utiliza aquí para identificar los nodos que pueden formar parte de un clúster por ser nodos que viajan en la misma dirección a una velocidad similar. Estos datos también son útiles para determinar el destino de los mensajes, ya que por ejemplo, algunos mensajes deben propagarse sólo en una sola dirección, mientras que otros, como los avisos de congestión debido a un accidente, deben propagarse en ambas direcciones, hacia delante y hacia atrás. Por último, con respecto al parámetro *estado*, ya que en nuestro esquema todos los nodos pertenecen a algún clúster, por lo menos formado por sí mismo, el estado ND sólo puede ser utilizado para el estado inicial del nodo antes de ejecutar los protocolos descritos más adelante. Además, cuando un nodo pertenece a más de un clúster, se convierte en GW para la comunicación entre clústers, formando parte de la columna vertebral para la propagación de mensajes en la VANET.

3.3.2. Notación y Descripción General

Esta sección contiene la notación y descripción general de los procedimientos que forman parte de la arquitectura de clúster del sistema propuesto, incluyendo todas las etapas

posibles en la gestión de clústers, en función de la situación específica de los vehículos.

La notación básica utilizada a lo largo de los algoritmos que forman parte de la arquitectura propuesta es como sigue:

- x denota el nodo ejecutor del algoritmo.
- VecinosCH(x) es el conjunto de CHs vecinos de x.
- vecino(i) denota el i-ésimo vecino del nodo ejecutor, que es un potencial miembro del clúster.
- esCH(i) es una función booleana que indica si cierto nodo i es CH o no.
- PeticionCreacion(x) representa un mensaje enviado por x conteniendo la petición de creación de clúster.
- Recibe(Mensaje,i,x) indica que el mensaje del nodo i es recibido por x.
- ListaCluster[] contiene los miembros del clúster.
- CHNom es un mensaje que indica que el nodo emisor está nominado para ser el CH del clúster que se está formando.
- Peso(i) es el valor asociado al nodo i utilizado para determinar su candidatura al rol de CH de acuerdo con parámetros como su número de vecinos, localización, velocidad, etc.
- PeticionClave(x) representa un mensaje enviado a x conteniendo una petición de clave compartida.
- p es un número primo.
- g denota un elemento generador de Z_p.
- S_i es un entero en $[0, p-2]$ aleatoriamente elegido por el nodo i.
- g^{S_i} denota el compromiso público de i con el entero S_i.
- h representa una función hash.

- K_x es la clave secreta del clúster con líder x.
- Espera(T) implica esperar durante un tiempo T antes de proceder con el siguiente paso.

El esquema de clústers propuesto en este trabajo está formado por las siguientes etapas. Cuando un nodo no es miembro de ningún grupo, se inicia la fase de inicialización donde se implementa un proceso de descubrimiento de grupo. Después de esto, el nodo puede ejecutar ya sea el procedimiento de unión a clúster o de la fase de creación de clúster, dependiendo de si hay un CH cerca o no. Durante la creación del clúster se llevan a cabo tanto la selección del CH como el establecimiento de la clave secreta del clúster. Después de toda la fase de creación de clúster, y también después de unirse a un grupo, los nodos miembros del clúster colaboran en el procedimiento de mantenimiento que periódicamente revisa la validez del clúster.

3.3.3. Inicialización de Vehículo

Esta es la primera etapa que se inicia cuando un vehículo arranca y su estado es ND porque todavía no pertenece a ningún clúster. Esta etapa se describe en el algoritmo de inicialización, que el nodo ejecuta con el fin de descubrir si existe un CH cerca o no.

Cada vehículo que se encuentra en estado ND tiene que comprobar periódicamente si entre los vecinos que viajan dentro de su rango y al mismo ritmo, existe algún CH. Si hay al menos un vecino que es candidato a ser CH, el nodo inicia el procedimiento de unión a clúster explicado más adelante. De lo contrario, se procede a la etapa de creación del clúster definida a continuación. Hay que tener en cuenta que esta etapa de inicialización no genera ningún tráfico de control adicional debido a que toda la información necesaria para su ejecución está contenida en los beacons que los nodos difunden periódicamente para anunciar su presencia a los demás.

Algoritmo Inicialización de Vehículo

01: **función** InicializacionVehiculo (...)

02: $i = 1$;

03: cardinal(VecinosCH(x)) = 0;

```
04:  mientras (existe vecino(i)) hacer
05:     si esCH(vecino(i)) entonces
06:        cardinal(VecinosCH(x))++;
07:     fin si
08:     i++;
09:  fin mientras
10:  si (cardinal(VecinosCH(x)) == 0) entonces
11:     CreacionCluster();
12:  si no
13:     para (j=1; j ≤ cardinal(VecinosCH(x));j++) hacer
14:        UnionaCluster();
15:     fin para
16:  fin si
17: fin función
```

3.3.4. Creación del Clúster

El algoritmo de creación del clúster se ejecuta cada vez que un nodo que está en el estado ND y que ya ha ejecutado la fase de inicialización, ha descubierto que no está cerca de ningún CH. Con el fin de iniciar el proceso de creación de clúster, el nodo ejecutor difunde una solicitud de creación de clúster hacia todos los vecinos que viajan en la misma dirección, con una distancia igual a 1 salto y velocidad dentro del rango de velocidad del CH. Los nodos que reciben esta petición pueden responder aceptando la invitación e indicando cuántos de sus vecinos son candidatos a ser miembros de un nuevo clúster donde él sería el CH. Después de esto, la etapa de selección del CH se pondrá en marcha por los nodos que respondieron a la invitación, y la clave secreta compartida se establecerá de acuerdo con el protocolo de establecimiento de clave secreta del clúster. Después de esto, el nuevo clúster puede considerarse como ya establecido.

Algoritmo Creación de Clúster

```
01: función CreacionCluster (...)
02:    l = 1; i = 1;
```

```
03:  Retransmite(PeticionCreacion(x));
04:  mientras (existe vecino(i))hacer
05:    si Recibe(Respuesta,i,x) entonces
06:      ListaCluster[l] = vecino[i];
07:      l++;
08:    fin si
09:    i++;
10:  fin para
11:  si (l ≥ 1) entonces
12:    SeleccionCH(ListaCluster[]);
13:  si no
14:    estado=CH;
15:  fin si
16: fin función
```

En conclusión, esta etapa de creación de clúster requiere básicamente una transmisión de invitación para unirse al nuevo clúster y las respuestas unidireccionales de n nodos candidatos, lo que significa un total de $2n$ paquetes. En consecuencia, la gestión de los paquetes generados en esta etapa no disminuyen el rendimiento de las comunicaciones de forma importante.

3.3.5. Selección de Líder del Clúster

En esta sección se propone un algoritmo para seleccionar un nodo como CH del clúster que se está formando. La idea principal del algoritmo de selección del CH es permitir que un nodo evalúe su potencial como CH antes de tomar ese papel y renuncie a ello si concluye que no es el mejor candidato a CH en ese momento. Cuando un nodo decide convertirse en CH debe transmitir un mensaje de invitación para reclutar a sus vecinos. Después de recibir la invitación del nuevo candidato a CH, los nodos vecinos deciden si unirse al nuevo clúster. Cada CH comprueba periódicamente si los miembros de su clúster tienen más capacidad para ser CH que él mismo y, si uno de estos vecinos es mejor candidato que él, dimite y propone a ese nodo para convertirse en el nuevo CH del clúster. Este proceso

de renovación también se ejecuta automáticamente si el CH sale de su clúster de forma repentina.

Los criterios que se utilizan para la selección de CH son múltiples. Por un lado, el CH tiene la menor probabilidad (en comparación con otros dentro del mismo clúster) de salir del clúster virtual actual, porque su velocidad es cercana a la velocidad promedio en el clúster. Esto asegura que un nodo que se mueve a una velocidad media diferente a la de sus vecinos no sea elegido como CH. Al mismo tiempo, la eficiencia de las comunicaciones entre clústers se maximiza con la elección del CH propuesto porque en nuestro esquema el CH tiene la distancia mínima desde el centro virtual del clúster correspondiente.

El problema de la organización de la VANET en clústers se entiende como el problema de encontrar el conjunto independiente de máximo peso en el grafo correspondiente a la red. Se introduce un algoritmo distribuido para la determinación eficiente de ese conjunto independiente en el grafo que representa la VANET, que sólo requiere que cada nodo tenga cierto conocimiento de su entorno. El problema del conjunto independiente del máximo peso es un conocido problema NP-duro [91], pero en este trabajo sólo estamos interesados en el problema para la clase específica de grafos que representan la topología de una VANET.

De acuerdo con nuestra definición de clúster, no hay dos CHs que puedan ser vecinos en el momento de la formación de alguno de ellos. Además, la red tiene que ser cubierta con una columna vertebral de clústers donde cada clúster tiene un CH y posibles GWs compartidos con otros clústers, lo que implica que cada nodo debe tener al menos un CH en su vecindario. En consecuencia, el problema de detección del CH se puede reducir al problema del conjunto dominante, que está estrechamente relacionado con el problema de encontrar un conjunto independiente maximal de nodos en el grafo de la red ya que todo conjunto independiente maximal es un conjunto dominante minimal. En particular, en nuestra solución asociamos un peso a cada nodo para indicar su nivel de adecuación para el papel del CH en función de parámetros como el número de vecinos, la ubicación, velocidad, etc. Por lo tanto, el algoritmo de selección de CH es equivalente al problema de encontrar un conjunto independiente de máximo peso en el grafo de la red y los nodos en el conjunto independiente serán los CHs. Con el fin de ejecutar el algoritmo, cada miembro del clúster sólo tiene que saber el peso de sus vecinos. Inicialmente, sólo los nodos con mayor peso con

respecto a sus vecinos difunden un mensaje a sus vecinos candidatos indicando que ellos son los candidatos a CH. En una segunda ronda, si un nodo no recibe ninguno de estos mensajes, retransmite uno de ellos. De lo contrario, se comprueba si su papel será como MN o como GW.

Algoritmo Selección de Líder del Clúster

```
01: función SeleccionCH (...)
02:   CHNom = 1;
03:   para (i=1; i ≤ Cardinal(ListaCluster);i++) hacer
04:     si peso(i) > peso(x) entonces
05:       CHNom = 0;
06:     fin si
07:     si CHNom == 1 entonces
08:       estado=CH;
09:       Retransmite(CHNom);
10:       EstablecimientoClaveSecreta(ListaCluster[]);
11:     si no
12:       si Cardinal(Recibe(CHNom, ListaCluster[],x))==0 entonces
13:         estado=CH;
14:         Retransmite(CHNom);
15:         EstablecimientoClaveSecreta(ListaCluster[]);
16:       si no
17:         si Cardinal(Recibe(CHNom,ListaCluster[],x))==1 entonces
18:           estado=MN;
19:         si no
20:           estado=GW;
21:         fin si
22:       fin si
23:     fin si
24:   fin para
25: fin función
```

3.3.6. Establecimiento de Clave Secreta del Clúster

La mayoría de las referencias acerca de las comunicaciones secretas en VANET sugieren el uso de criptografía de clave pública basada en una infraestructura de clave pública con certificados emitidos por una autoridad de certificación. Esta solución implica que un par de claves pública/privada se le asigne a cada nodo y se almacene en su dispositivo a prueba de falsificaciones y los certificados de clave pública sean autenticados ya sea por una CA centralizada, o distribuida tal y como hemos propuesto antes en este trabajo.

Nuestra propuesta de esquema basado en clústers permite combinar PKIs con el uso de criptografía de clave secreta para reducir la sobrecarga que supone el uso de criptografía de clave pública en exclusiva. En particular, aquí se propone el establecimiento y uso de claves secretas compartidas en clústers, porque la criptografía de clave secreta es en general más eficiente que la criptografía de clave pública. El gran tamaño de las VANETs no permite la precarga de claves compartidas en los vehículos, por lo que el establecimiento de la clave secreta debe ser dinámico y distribuido. Hay que tener en cuenta que la comunicación mediante clave secreta compartida y la proximidad de los miembros del clúster que se comunican en modo promiscuo, permiten a los nodos del clúster controlar que tanto el CH como otros nodos en el clúster actúen correctamente, lo que tiene la utilidad adicional de servir para el fomento de la cooperación correcta.

A fin de preservar la igualdad de roles de las OBUs en VANET, nos aprovechamos de la naturaleza distribuida de los clústers propuestos para definir un proceso para establecer claves de forma consensuada. Se pueden utilizar varios métodos, pero el que proponemos aquí implica que el CH envíe cierta información a todos los miembros, que les permita calcular de forma independiente la misma clave secreta compartida.

El protocolo de establecimiento de claves propuesto establece una clave secreta para todos los miembros de un clúster, basado en una contribución que cada nodo intercambia abiertamente en un medio inalámbrico no seguro. La clave secreta se obtiene con el algoritmo de establecimiento de clave secreta y se puede utilizar para establecer un canal seguro entre todos los miembros del clúster.

En particular, en el esquema descrito a continuación, los nodos que forman un

nuevo clúster generán una clave secreta compartida a través de un esquema basado en la dificultad del problema del logaritmo discreto, que consiste en calcular el valor de un entero positivo S, conocidos los enteros positivos $g^S(mod\ p)$, g y p. Este problema es la base del conocido método de Diffie-Hellman para el establecimiento de un secreto compartido entre dos partes mediante comunicaciones inseguras. En consecuencia, el algoritmo propuesto supone el uso de una generalización del protocolo de Diffie-Hellman para más de dos usuarios.

El siguiente algoritmo se basa en un esquema de compromiso de bits para que cada nodo i envíe al CH su contribución a la clave secreta compartida. De esta manera, el CH, el cual en el algoritmo se denota como nodo ejecutor x, no puede cambiar esta contribución, ni leerla. El uso de un esquema de compromiso de bits hace posible el intercambio de información pública para permitir la generación de la clave secreta compartida por cada nodo, sin poner en situación de riesgo ni la clave secreta compartida ni las diferentes contribuciones a ella.

Algoritmo Establecimiento de Clave Secreta

01: **función** EstablecimientoClaveSecreta(...)
02: Retransmite(PeticionClave(x));
03: **para** (i=1; $i \leq$ Cardinal(ListaCluster);i++) **hacer**
04: Recibe($g^{S_i}\ (mod\ p)$, i, x);
05: **fin para**
06: Retransmite($\{h(g^{S_i}), g^{S_i S_x}(\ mod\ p)\}\forall_{i \neq x}$);
07: $K_x = g^{S_x(1+\sum_{i \neq x} S_i)}(mod\ p)$;
08: **fin función**

Téngase en cuenta que la retransmisión en el paso 6 del algoritmo anterior no representa una amenaza para el secreto de la clave del clúster, ya que es inútil para cualquier nodo que no ha contribuido al secreto. También es importante destacar que si bien el algoritmo anterior es ejecutado por el CH, cada miembro i del clúster puede comprobar si su contribución se ha incluido correctamente en el mensaje enviado por el CH. En tal caso, se puede calcular de forma independiente la clave secreta del clúster a partir del mensaje recibido del CH, mediante la eliminación de su aportación a $g^{S_i S_{CH}}$ para obtener $g^{S_{CH}}$ y luego calcular la clave secreta de acuerdo con la expresión:

$$K_{CH} = g^{S_{CH}} \cdot \prod_{i \neq CH} g^{S_i S_{CH}} = g^{S_{CH}\left(1+\sum_{i \neq CH} S_i\right)} (mod\ p)$$

De acuerdo con el algoritmo anterior, la clave del clúster se genera con las aportaciones de los primeros miembros del clúster. Mientras el clúster exista, cada posible nuevo nodo que se una a él recibirá del CH la clave secreta del clúster, cifrada con la clave pública del nuevo nodo.

Se debe tener en cuenta que el uso propuesto de clústers reduce la sobrecarga de las comunicaciones, pero en general no permite definir diferentes niveles de seguridad entre los miembros de un mismo clúster. Por el contrario, protege principalmente la red de los potenciales atacantes externos. Por lo tanto, la entrega de las claves secretas compartidas existentes en un clúster durante la unión de un nuevo miembro es necesaria, pero los miembros que dejan de formar parte del clúster no podrán ver ninguna actualización de la clave del clúster y por consiguiente, ninguna nueva comunicación que se realize perteneciente al clúster.

El cifrado de clave secreta es en general más eficiente que el cifrado de clave pública, por lo que gracias al proceso de compartir secretos de establecimiento de claves propuesto más arriba, cualquier tipo de cifrado de clave secreta puede ser utilizado en VANETs para proteger la confidencialidad mediante el cifrado de clave secreta. Aparte de las aplicaciones relacionadas con la seguridad vial, existen otros escenarios donde son necesarias las comunicaciones confidenciales. Este es el caso por ejemplo de algunas aplicaciones comerciales.

Por otro lado, es conveniente usar un método efectivo para controlar el acceso a los mensajes de multidifusión en VANETs con objeto de evitar comportamientos pasivos. Si los nodos receptores comparten una clave secreta, es posible usarla para limitar el acceso a los mensajes difundidos a través del cifrado. Por lo tanto, después de que se haya definido la clave secreta del clúster mediante el algoritmo propuesto, la multidifusión cifrada dentro del clúster ya es posible.

3.3.7. Unión a Clúster

Esta etapa se inicia cuando un vehículo encuentra entre sus vecinos al menos un nodo que es CH. El algoritmo de unión a clúster descrito a continuación muestra un escenario

en el cual un nodo se une a todos los clústers correspondientes a CHs en su vecindario.

Con el fin de llevar a cabo esta etapa, primero el nodo tiene que enviar una solicitud de inicio de sesión cifrada con la clave pública de cada CH vecino. Después de la autenticación, el CH envía la correspondiente clave secreta del clúster cifrada con la clave pública del nodo ejecutor. De esta manera, el nodo pasa a formar parte del clúster y procede a la fase de mantenimiento del clúster.

Algoritmo Unión a Clúster

```
01: función UnionaCluster(...)
02:    para (j=1; j ≤ Cardinal(VecinosCH(x)); j++) hacer
03:       PeticionClave(j);
04:       Recibe(K_j,j,x);
05:       MantenimientoCluster();
06:    fin para
07: fin función
```

3.3.8. Mantenimiento de Clúster

La movilidad en VANETs suele ser muy dinámica. Por ejemplo, normalmente existe un conjunto de vehículos que conducen cerca de otros durante varios kilómetros, mientras otros vehículos pasan rápidamente. Esta es la razón principal por la que es necesario estar constantemente ejecutando la fase de mantenimiento del clúster, que consiste en comprobar la validez de los clústers.

El algoritmo de mantenimiento de clúster muestra el proceso que un MN o GW ha de llevar a cabo mientras pertenece al clúster. Los nodos comprueban que no han perdido contacto con su CH cada T unidades de tiempo. El nodo considera que ha perdido el contacto con su CH si no ha recibido ningún mensaje de su CH durante más de dos periodos T. En tal caso, cambia su estado a ND y comienza la fase de inicialización.

Algoritmo Mantenimiento de Clúster

```
01: función MantenimientoCluster (...)
02:    mientras (Recibe(Mensaje,CH,x) hacer
03:       Espera(T);
```

```
04:   fin mientras
05:   Espera(T);
06:   si (Recibe(Mensaje,CH,x) entonces
07:      MantenimientoCluster();
08:   si no
09:      estado=ND;
10:      InicializacionVehiculo();
11:   fin si
12: fin función
```

3.3.9. Gestión de Mensaje

Gracias a la utilización de clústers, el número de comunicaciones puede disminuir notablemente sin perder ningún tipo de información útil. El algoritmo de gestión de mensaje muestra los pasos que un nodo debe ejecutar después de recibir o generar un mensaje.

Si el nodo ejecutor es el destino final del mensaje, simplemente procesa la información. De lo contrario, su reacción depende de si el nodo es CH o no. En este último caso, se lo envía a través de unicast cifrado al CH. Si el nodo ejecutor es CH, su acción depende de si el mensaje es para ser reenviado o no. En el primer caso, se envía a través de multicast cifrado con la clave del clúster a todos los miembros. De lo contrario, si el mensaje tiene un solo destino y este destino pertenece a su clúster, el mensaje se envía cifrado mediante unicast al destino que se encuentra dentro del clúster. De lo contrario, si el único destino no pertenece a su clúster, envía a través de unicast cifrando el mensaje a todos los GWs, que lo retransmitirán a otros CHs a través de una comunicación cifrada entre clústers.

Téngase en cuenta que la existencia de GWs es de gran importancia, ya que proporcionan un cierto grado de solapamiento entre clústers lo que facilita la comunicación entre clústers y no sólo para los mensajes de retransmisión, sino también para otras aplicaciones como el descubrimiento de la topología de la red y la localización de nodos.

También es notable que, sobre todo cuando un mensaje se propaga, y cuando un mensaje requiere comunicación entre clústers, los mecanismos de cooperación pueden ser

útiles porque sin ellos, los vehículos intermedios pueden no tener los incentivos necesarios para retransmitir comunicaciones de otros.

Algoritmo Gestión de Mensaje

```
01: función GestionMensaje (...)
02:   si (Destino(Mensaje)==x) entonces
03:     Procesar(Mensaje);
04:   si no
05:     si (estado==CH) entonces
06:       si (Cardinal(Destino(Mensaje))==1)) entonces
07:         si (Destino(Mensaje) en ListaCluster[]) entonces
08:           Unicast(Mensaje, x, Destino(Mensaje));
09:         si no
10:           Multicast(Mensaje, x, GWs);
11:         fin si
12:       si no
13:           Multicast(Mensaje, ListaCluster[]);
14:       fin si
15:     si no
16:       Unicast(Mensaje, x, CH);
17:     fin si
18:   fin si
19: fin función
```

3.4. Simulación de las Propuestas

Tanto la viabilidad como la eficacia de nuestro enfoque se pueden ver a través de las siguientes figuras que reflejan las simulaciones realizadas. En la primera, Fig. 3.10, una pantalla de NS-2 y SUMO muestra el estado de la VANET cuando los clústers están en funcionamiento. En esta simulación se compararon tres modelos: una VANET sin clústers, nuestra propuesta basada en clústers y finalmente el enfoque CARAVAN mencionado en

el estado del arte, en el que cada nodo está en el rango de transmisión de todos los nodos del clúster. Clústers estáticos o clústers con más de un salto no se han tenido en cuenta debido a que las diferencias en el número de comunicaciones son evidentes. Por un lado, con clústers estáticos cada nodo cambia muchas veces de clúster. Por otro lado, mediante clústers de más de un salto, el número de paquetes de control es mayor que con un solo salto debido a que el CH tiene que conocer todos los nodos a dos o más saltos de distancia lo que dificulta mucho la gestión.

Las opciones más relevantes para las simulaciones realizadas han sido:

- número total de vehículos: 80,
- número de vehículos con OBUs: 80,
- número de carriles para cada sentido: 3,
- tiempo de simulación: 100 segundos,
- momento en el que comienzan las retransmisiones: segundo 40,
- periodo de retransmisión: 10 segundos,
- distancia de retransmisión de nodos: 75 metros,
- distancia recorrida antes de que suceda el atasco de tráfico: 800 metros.

Las simulaciones realizadas con clústers se hicieron considerando cuatro niveles de desarrollo:

1. La capa de movilidad del vehículo gestiona el movimiento de los nodos del modelo de simulación, que define las carreteras, los carriles, los límites de velocidad, posibles atascos, etc.
2. La capa de energía del nodo se utiliza para distinguir entre vehículos con y sin OBU ya que los vehículos sin OBU que están presentes en el camino hay que tenerlos en cuenta por el espacio físico que ocupan, pero no contribuyen en las comunicaciones.

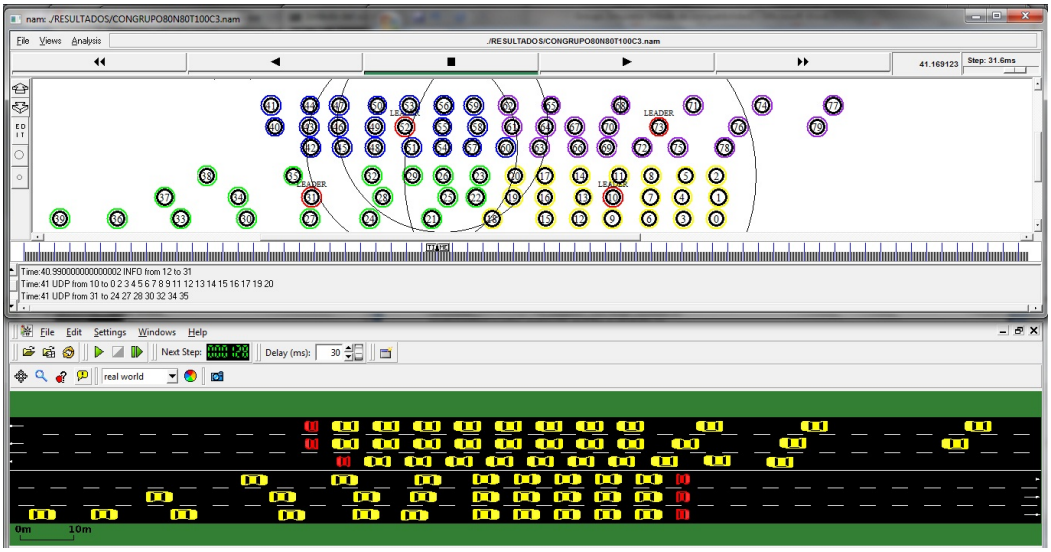


Figura 3.10: Captura de Pantalla de una Simulación

3. La capa de formación de clústers define qué vehículos pertenecen a cada clúster y sus funciones, es decir, quién es el CH de cada clúster, quiénes generan la información de tráfico y quiénes son los GWs que retransmiten la información a otros clústers.

4. La capa de comunicaciones P2P es responsable de la definición de los nodos que están en el rango de retransmisión del nodo en cada momento.

Las simulaciones permiten obtener estadísticas esenciales, tales como número de paquetes generados y perdidos. Estos datos estadísticos básicos son útiles para realizar simulaciones eficientes en escenarios a gran escala.

Tanto las simulaciones implementadas usando nuestra propuesta como los clústers usando CARAVAN pueden ser comparados con los resultados obtenidos de la simulación sin el uso de clústers, todos usando la misma topología de carretera (ver Fig. 3.11). Esto ayuda a ilustrar la evaluación del uso de clústers. Las VANETs simuladas se suponen formadas por vehículos conectados por Wi-Fi 802.11b/g, que se estima que tiene un alcance entre 50 y 300 metros. Las simulaciones realizadas no tienen en cuenta el efecto Doppler, ya que en condiciones de tráfico denso este factor se minimiza debido a la baja velocidad de los vehículos. Tampoco tienen en cuenta posibles interferencias con otros dispositivos en el

mismo rango de retransmisión u otras WLAN.

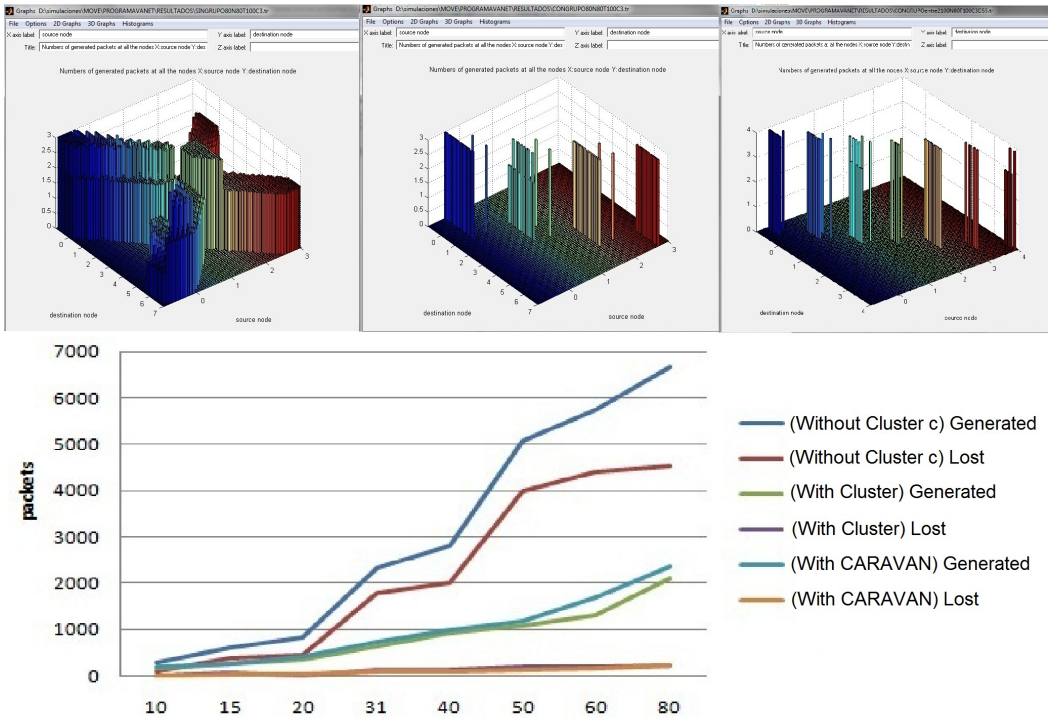


Figura 3.11: Simulaciones Con y Sin Clústers, y Con CARAVAN

Entre la información obtenida de las simulaciones se tiene el número de paquetes y bytes generados, enviados, transmitidos, recibidos, perdidos, etc., para cada nodo.

La Tabla 3.1 muestra algunos resultados de la simulación. Además, otra información que se obtiene mediante las simulaciones es el número de paquetes generados y perdidos en toda la red, el número de clústers formados, qué nodos son los CHs de cada clúster, qué nodos generan los paquetes y cuáles los retransmiten, etc. Además de toda esta información, las simulaciones proporcionan detalles de lo que ocurre en cada momento en la VANET gracias al uso del visor de simulaciones NS-2, el NAM. También se muestra el modelo de tráfico a través de la herramienta SUMO, mientras que la información se representa mediante TraceGraph.

En la Fig. 3.12 podemos ver una comparación entre el promedio de paquetes generados y perdidos. Se deduce que sin el uso de clústers en VANET, el número de paquetes

Tabla 3.1: Resultados de Simulación

	Sin Clústers		Con Clústers		Con CARAVAN	
n. de nodos	paquetes enviados	paquetes perdidos	paquetes enviados	paquetes perdidos	paquetes enviados	paquetes perdidos
10	278	107	167	12	187	20
15	598	402	277	43	271	48
20	825	443	351	0	427	52
31	2343	1804	638	79	749	88
40	2805	2014	932	95	1009	100
50	5077	3981	1101	132	1190	137
60	5732	4415	1314	159	1693	168
80	6675	4529	2120	215	2357	232

generados crece mucho más rápido que usando clústers, y también crece el número de paquetes perdidos. La principal razón es la sobrecarga de tráfico que generan las VANETs en condiciones de tráfico denso. De hecho, el modelo básico sin clústers supone un uso masivo de las operaciones de retransmisión. Está claro que los clústers ayudan a disminuir el porcentaje de paquetes perdidos y a realizar las operaciones de gestión de las VANETs. En la comparación entre nuestra propuesta y la de CARAVAN, tanto el número de paquetes que se genera como el número de paquetes perdidos son más pequeños en nuestra propuesta. Además, el tiempo de retransmisión en CARAVAN es más grande que en nuestra propuesta, porque los clústers que se forman son más pequeños y el número de nodos a través de los cuales pasa la información es mayor.

La Fig. 3.12 muestra el tamaño promedio de clústers de diferentes densidades considerando diferentes rangos de transmisión usando nuestro enfoque y en el enfoque de CARAVAN. Los clústers mayores son los que tienen la mayor densidad de vehículos. Las propuesta fueron simuladas en un entorno donde se producía un atasco de tráfico realista, con una autopista de tres carriles en cada sentido y 100 vehículos. Se puede observar que el número de vehículos perteneciente a cada clúster aumenta de forma lineal. El segundo gráfico muestra el tamaño promedio de los clústers con el enfoque de CARAVAN. En este caso podemos ver que el número de nodos que pertenecen a cada clúster es aproximadamente la

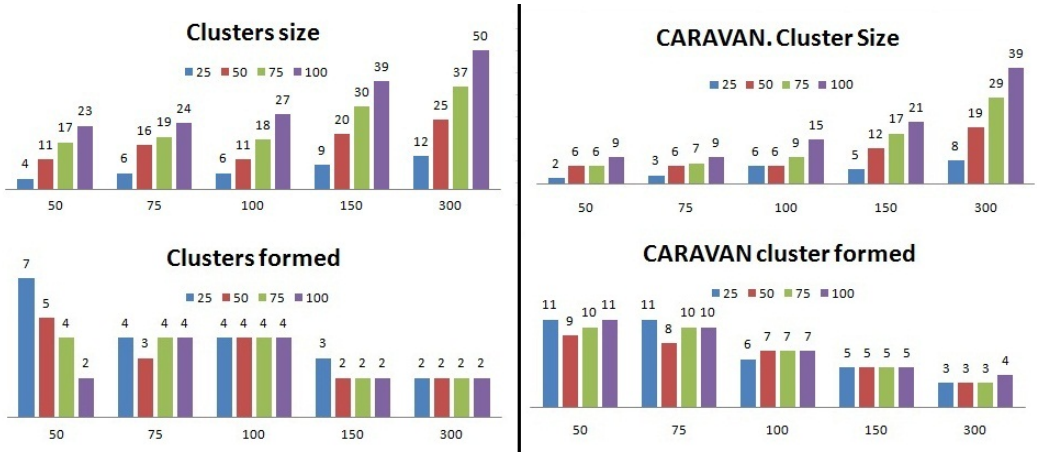


Figura 3.12: Tamaño y Número de Clústers en Ambas Propuestas

mitad de los nodos en nuestro enfoque. La Fig. 3.12 muestra también el número de clústers formados. En este caso podemos ver que el aumento del rango de transmisión reduce el número de clústers. Éste es un resultado normal porque el tamaño de los clústers es mayor a medida que el rango de transmisión es mayor. En el enfoque de CARAVAN podemos ver lo mismo, pero en este caso, el número de clústers es mayor que en nuestro enfoque. Esto se debe a que todos los nodos deben tener visión directa con todos los nodos del clúster y eso hace que los clústers en CARAVAN sean más pequeños.

Capítulo 4

Aplicación Móvil para la Asistencia a la Conducción (VAiPho)

Este capítulo contiene los principales detalles de la implementación real realizada de algunos algoritmos diseñados en esta tesis como parte de una aplicación desarrollada para teléfonos móviles, llamada VAiPho (VANET in Phones) con objeto de desplegar una VANET real con utilidad inmediata en detección de atascos y plazas de aparcamiento, entre otras opciones. En particular, el capítulo se divide en dos secciones en las que se especifican las interfaces y esquemas usados para la implementación básica, así como las ideas para resolver el problema práctico de la fusión de subredes.

4.1. Estado del Arte

Cuando se diseña una herramienta para crear una red de vehículos auto-organizada con el objetivo de aumentar la seguridad vial, el primer requisito que se debe tener en cuenta es la exactitud y fiabilidad de la información retransmitida. Por lo tanto, la seguridad es el tema más importante a tener en cuenta a la hora de diseñar un sistema de comunicaciones para VANETs [172]. Las redes ad-hoc vehiculares son susceptibles a diferentes tipos de ataques. En la bibliografía se encuentran algunos esquemas propuestos para VANETs auto-organizadas [1] [54] [163] [212]. Los autores de esos documentos tratan de resolver todos

o parte de los problemas de seguridad presentes en este tipo de redes. Sin embargo, en este capítulo se presenta un enfoque diferente. El trabajo [72] tiene el mismo objetivo que esta sección, pero no aborda el principal aspecto de la seguridad de las conexiones. Un aspecto importante de seguridad es el anonimato del usuario. Por ello en VAiPho se utilizan pseudónimos variables para garantizar una alta probabilidad de que un atacante no pueda realizar un seguimiento del vehículo. En [16] se propone un sistema de pseudónimos, usado cuando las coordenadas y la velocidad de los vehículos son enviados en los beacons. En nuestra propuesta no se envían coordenadas ni velocidad en los beacons.

Nuestra propuesta tiene en cuenta que la integración de las VANETs será gradual, por lo que al principio, no habrá RSUs, y las primeras VANETs comenzarán con unos pocos vehículos, que después se irán incrementando. Este crecimiento del número de nodos en la VANET será más rápido o más lento dependiendo de la popularidad, aceptación y facilidad de uso de la herramienta. Por esta razón, la interfaz de usuario es una característica muy importante de la aplicación propuesta. Un complemento ideal para VAiPho se describe en [14] donde las aplicaciones que se ejecutan en el dispositivo móvil se muestran en un terminal más grande y por lo tanto, más fácil de usar para el conductor. VAiPho también puede ser combinado con [2] para que el usuario pueda interactuar con el dispositivo utilizando la voz para acceder a funcionalidades tales como *"muestra plazas de aparcamiento"* o *"muestra atascos hasta destino X"*, donde el destino debe ser previamente almacenada. VAiPho también proporciona una solución más realista que en la propuesta [182] para la búsqueda del vehículo estacionado, a pesar de que sólo funciona en aparcamientos al aire libre.

En cuanto a la solución para la búsqueda de aparcamientos existen menos soluciones propuestas y ninguna de ellas está implantada en la realidad. En [142] se presenta una solución a la búsqueda de aparcamientos mediante un dispositivo instalado en la puerta del copiloto que detecta las plazas libres de aparcamientos y las comunica a un servidor a través del cual los usuarios que buscan aparcamiento pueden localizar dichas plazas libres. Esta solución aparte del hardware de detección requiere del uso de 3G o GPRS para poder funcionar. En [162] los autores muestran una solución donde los usuarios pueden encontrar plazas de aparcamiento en una red segura sin el apoyo de extensas infraestructuras, no obstante el esquema tiene problemas de seguridad puesto que es fácil para un atacante indicar

qué aparcamientos libres están ocupados o viceversa. [60] presenta una solución para encontrar un aparcamiento en función de su distancia al usuario. Permite conocer la información exacta de la posición del aparcamiento o el número de plazas que hay dentro de un parking. Por último, la solución para encontrar el coche aparcado es la más fácil de implementar y de la que más aplicaciones se pueden encontrar para las principales plataformas móviles, [15], [55], [56], [90], [127], [133].

4.2. Diseño e Implementación de VAiPho

Entre los diferentes tipos de información que pueden ponerse a disposición de los conductores a través de las VANETs, la información acerca de las condiciones del tráfico para reducir las congestiones es una de las más importantes.

Esta sección muestra las interfaces y el modo de funcionamiento de VAiPho, una nueva herramienta que ofrece soluciones para garantizar el funcionamiento de las VANETs para la detección de eventos de tráfico y el intercambio de información utilizando sólo teléfonos móviles y sin necesidad de ningun tipo de autoridad centralizada. Como veremos más adelante, las interfaces de VAiPho han sido diseñadas para evitar distracciones durante la conducción, ya que operan de forma automática e independiente del conductor a través de mensajes de voz. Además, en la implementación de VAiPho se tuvieron en cuenta las características de seguridad que son esenciales en las redes, tales como la autenticidad, confidencialidad, integridad y no repudio gracias a la incorporación a su diseño de varios de los algoritmos expuestos anteriormente en esta tesis. Todo esto permite el despliegue de una VANET real inmediata sin aumentar el precio de los vehículos por la necesidad de integrar nuevos dispositivos como OBUs, simplemente mediante el uso de teléfonos móviles equipados con conexión inalámbrica y GPS.

4.2.1. Utilidad de VAiPho

El objetivo principal de este trabajo es definir un modelo simple y escalable para VANETs donde los usuarios pueden colaborar a través de sus dispositivos móviles y obtener información de interés actualizada sobre el tráfico en la zona con el fin de elegir la mejor

ruta hasta su destino, optimizada gracias a la información en tiempo real de la carretera.

Cada año hay más y más atascos en las carreteras debido a que el número de vehículos sigue aumentando muy rápidamente, de modo que en 2011 hay cerca de 1000 millones de automóviles en el mundo. Debido a los atascos en [203], los autores estimaron que en Estados Unidos hubo una pérdida de 78 mil millones de dólares sólo en 2007, en forma de 4.2 billones de horas perdidas y 2,9 billones de litros de gasolina consumidos.

Los atascos de tráfico producen altos niveles de estrés en las personas. Además, son la principal causa de contaminación del aire. Se ha demostrado que las personas atrapadas en atascos de tráfico tienen tres veces más probabilidades de sufrir un ataque al corazón que aquellos que no están atrapados en ningún atasco, aunque no esté claro si los ataques cardíacos están relacionados con el estrés o con la exposición a altos niveles de contaminación provocados por el tráfico. De todos modos, las comunicaciones entre los vehículos podrían ayudar a prevenir estos problemas mediante la reducción de los atascos de tráfico, lo que evitaría también los gastos enormes de tiempo y dinero de los usuarios, y de las reservas de petróleo.

Hoy en día, muchas aplicaciones de software basadas en GPS ofrecen servicios de información sobre el tráfico, que obtienen a partir de las autoridades de tráfico locales, departamentos de policía o/y sistemas que vigilan el flujo del tráfico. Sin embargo, los datos que manejan esos sistemas, o bien no son datos en tiempo real por lo que no reflejan eventos que se acaban de producir, o bien si lo hacen pero sin respetar la privacidad de los usuarios, lo que provoca un rechazo por parte de ellos a usarlos.

Actualmente existen varias investigaciones e implementaciones para los mismos servicios que VAiPho resuelve, pero desde enfoques distintos. En cuanto a la solución de detección de atascos, Google Traffic [96], TomTom [202], Sygic [195] o Waze [209] ofrecen distintas soluciones para la detección de atascos. Las principales diferencias con VAiPho son que todas ellas necesitan estar conectadas por 3G o GPRS para funcionar, y que además los usuarios pierden completamente su privacidad puesto que tienen que proporcionar continuamente información propia a las empresas que dan soporte al servicio. De hecho, esa información luego podría ser usada incluso en contra del usuario para por ejemplo, multarlo si excede los límites de velocidad.

El resultado principal de esta tesis es VAiPho, un sistema de comunicaciones seguras para redes espontáneas y auto-organizada basadas en teléfonos móviles inteligentes con GPS y comunicación inalámbrica Wi-Fi, que funcionan sin necesidad de ningún tipo de infraestructura instalada en los vehículos ni en las carreteras. El modo de funcionamiento es totalmente distribuido y descentralizado. VAiPho tiene en cuenta la protección de la privacidad e integridad frente a diferentes ataques. Su objetivo principal es aumentar la seguridad de los pasajeros y el confort gracias al intercambio de mensajes de advertencia entre los vehículos. También ayuda a reducir las emisiones de CO_2, aumenta la eficiencia de la conducción evitando el malgasto de tiempo y combustible en los atascos de tráfico, y aumenta el confort del usuario al reducir las horas que pasa en la carretera para llegar a su destino, además del número de desaceleraciones bruscas. La estructura de VAiPho permite aprovechar otros servicios secundarios como la detección de plazas de estacionamiento libre, así como recordatorio de dónde está el coche aparcado.

En esta sección nos centramos en la primera fase del despliegue de las VANETs, cuando el número de dispositivos que cooperan en la carretera es bajo. Tan pronto como las VANETs se amplian, el modelo propuesto debería ser revisado para evitar comunicaciones innecesarias a fin de que el elevado número de comunicaciones que puede llegar a haber, no degrade la red. Este problema también se podría atajar mediante alguna de las soluciones basadas en clústers expuestas en esta tesis.

4.2.2. Requerimientos

La implementación de la aplicación para smartphones ha sido multiplataforma. En particular, VAiPho ha sido desarrollado en una primera versión beta a modo de prueba para las plataformas Windows Mobile, Symbian y Android. En un futuro próximo está previsto el desarrollo para los sistemas operativos iOS de i-Phone y RIM de BlackBerry. En todas ellas, las especificaciones mínimas del sistema mostradas en la Fig. 4.1 son necesarias para el funcionamiento óptimo de VAiPho. Las características mínimas que debe tener el smartphone para ejecutar VAiPho son:

- Bluetooth®: Permite la conexión del dispositivo con el vehículo, para que VAiPho se

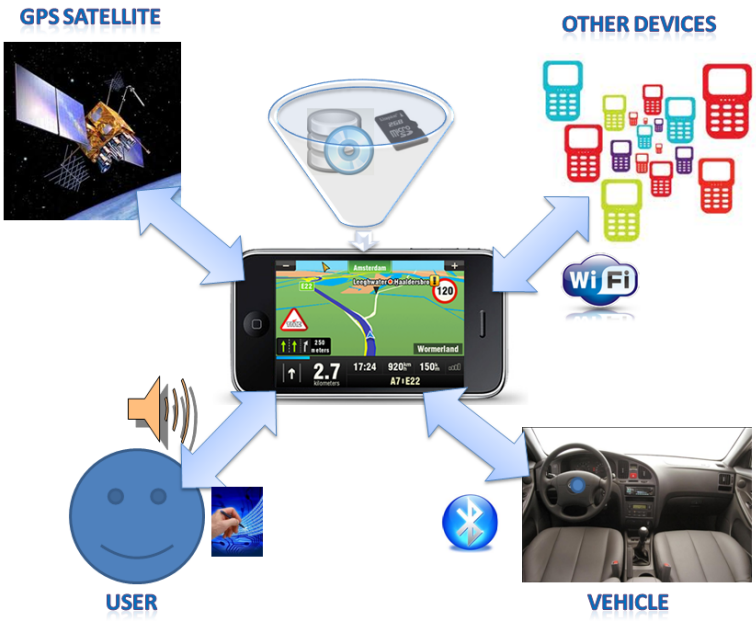


Figura 4.1: Requerimientos mínimos de VAiPho

active automáticamente sin requerir la atención del usuario.

- Wi-Fi® IEEE 802.11 b/g: Permite el libre intercambio de información entre los diferentes dispositivos que conforman la red.
- Base de Datos: Permite el almacenamiento de diferentes eventos, como pueden ser información sobre atascos o posibles plazas de aparcamiento, recordatorio de dónde está el vehículo estacionado o información publicitaria. También almacena información de otros usuarios, necesaria para la autenticación.
- Antena GPS: Permite obtener las coordenadas de los diferentes eventos que ocurren, así como la velocidad y dirección en la que el vehículo está circulando.
- Espacio de almacenamiento: Proporciona capacidad para instalar los programas necesarios.
- Capacidad para ejecutar programas: El dispositivo debe ser lo suficientemente potente para, en tiempo real, ejecutar programas que manejen y realizen operaciones complejas usando bastante información.

Como podemos ver, VAiPho utiliza la capacidad de los smartphones habituales hoy en día. Dado su precio asequible ya existe un gran número de usuarios que tienen este tipo de teléfonos, que permiten ejecutar una potente aplicación como VAiPho. Además, los usuarios de smartphones están acostumbrados a manejar estos dispositivos, por lo que la interfaz de usuario de VAiPho les resultará familiar, lo que evitará las dificultades habituales de aprender a utilizar un nuevo dispositivo.

Las comunicaciones inalámbricas en VAiPho se realizan usando el estándar para comunicaciones inalámbricas *IEEE 802.11b/g* a pesar de que no es el más adecuado para aplicaciones de seguridad vial debido a que estas comunicaciones, al estar en el rango de emisión de 2.4GHz, pueden tener interferencias con otros dispositivos Bluetooth, Zigbee y otras redes WLAN que pueden ocupar el mismo canal de emisión que la red VAiPho.

Añadido a éste hay otros problemas, como el retardo en la propagación de la señal, la retransmisión limitada debido a que la antena que tiene el móvil es pequeña, el uso de comunicaciones inalámbricas, que produce consumo de batería, y por último, el estándar *IEEE 802.11b/g*, que no está diseñado para trabajar con diferentes velocidades de los vehículos, por lo que las comunicaciones pueden fallar.

Para solucionar todos estos problemas se diseñó en teoría el estándar de comunicaciones inalámbricas *IEEE 802.11p*, que es más apropiado para este tipo de comunicaciones. No obstante, actualmente no existe ningún dispositivo con capacidad de retransmisión por la frecuencia de envío del estándar *IEEE 802.11p*. Sin embargo, la mayoria de los smartphones que se fabrican hoy en día sí tienen capacidad de comunicación mediante el estándar *IEEE 802.11b/g*. En cualquier caso, la herramienta VAiPho está diseñada en módulos, por lo que el módulo de comunicaciones puede ser fácilmente adaptado al estándar *IEEE 802.11p* en cuanto este se haya difundido.

Hemos realizado numerosas pruebas para comprobar las comunicaciones usando el estándar *IEEE 802.11b/g*, con vehículos y estas pruebas han sido satisfactorias. Es cierto que mediante este estándar, los vehículos que circulan en direcciones opuestas no tienen tiempo suficiente para comunicarse, pero los vehículos que circulan en la misma dirección o en ciudad, donde la velocidad es más baja, sí tienen tiempo suficiente para establecer comunicaciones.

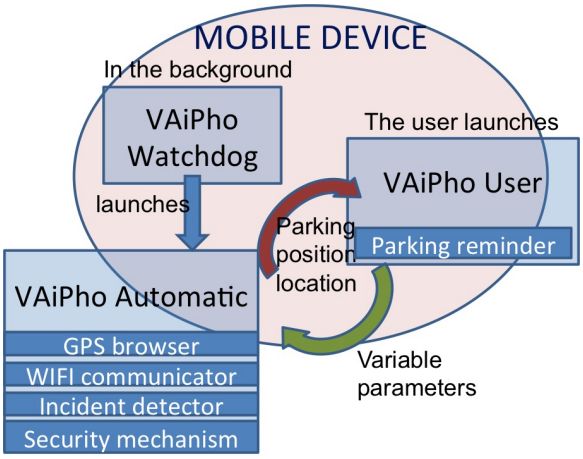


Figura 4.2: VAiPho - Funciones de las Diferentes Interfaces

4.2.3. Estructura de VAiPho

VAiPho está compuesto por tres módulos principales perfectamente diferenciados, como se muestra en la Fig. 4.2:

- *VAiPho WatchDog*, que se ejecuta en segundo plano, es un programa ligero que se encarga de detectar la conexión Bluetooth con el manos libres del coche para en ese caso, lanzar el programa VAiPho Automatic.

- *VAiPho Automatic*, lo inicia el módulo VAiPho WatchDog automáticamente para detectar los eventos que suceden en la carretera así como para mostrarlos al usuario y reenviarlos a otros usuarios.

- *VAiPho User*, este programa está especialmente diseñado para los usuarios ya que les permite interactuar con la aplicación y les ofrece diferentes funcionalidades interesantes.

En la siguiente subsección se explica la estructura interna de cada una de estas aplicaciones en profundidad.

VAiPho Watchdog

Uno de los temas más importantes en el diseño de aplicaciones para seguridad vial es que los usuarios deben mantener la atención en la conducción, porque esta es una de las principales causas de muertes en la carretera. Por este motivo proponemos VAiPho Watchdog, que es una aplicación ligera que se encarga de escuchar el registro del teléfono móvil y cuando este se conecta con el manos libres del coche, inicia automáticamente VAiPho Automatic por lo que no necesita ningún tipo de atención por parte del conductor para conectarlo y así evita distracciones. VAiPho Watchdog arranca automáticamente cuando el usuario enciende su teléfono móvil y se ejecuta en segundo plano. El programa consume muy poca batería porque su única función es escuchar el registro y lanzar *VAiPho Automatic* en caso de que el dispositivo se conecte por Bluetooth al manos libres del vehículo.

Un posible inconveniente de VAiPho es que utiliza múltiples interfaces de comunicación al mismo tiempo, lo que puede implicar un consumo de batería alto. Somos conscientes de que la prioridad de los teléfonos móviles son las llamadas telefónicas, y que el consumo de batería por parte de la aplicación VAiPho podría causar molestias a los usuarios. Para resolver este problema *VAiPho Watchdog* comprueba el nivel de la batería antes de iniciar *VAiPho Automatic*. Para llevar a cabo este proceso, estipulamos que la batería del teléfono puede estar en 5 estados: muy alta, alta, media, baja o muy baja. Si el estado de la batería alcanza un valor específico, no inicia *VAiPho Automatic*. Este valor puede ser modificado por los usuarios, que pueden elegir su valor preferido entre las 5 posibilidades mencionadas, si bien el valor por defecto es baja.

VAiPho Automatic

VAiPho Automatic es el programa principal de VAiPho. Además, es el más complejo porque lleva a cabo muchas tareas. VAiPho requiere conectividad con otros teléfonos a través de Wi-Fi®, Además, VAiPho utiliza la información de un software GPS, como la velocidad del carril, la velocidad del vehículo o las coordenadas con el fin de detectar posibles atascos o aparcamientos libres. Esta información tiene que ser procesada, almacenada y enviada a otros vehículos. Para ello el programa implementa las siguientes funciones:

- Inicia la interfaz inalámbrica
- Crea una red ad-hoc llamada *vaipho*
- Carga datos de la base de datos y del archivo de certificados
- Carga e inicia la retrasmisión de beacons por parte del cliente y la escucha de paquetes por parte del servidor. Cuando se encuentra con otros usuarios, implementa mecanismos de seguridad como la autenticación.
- Inicia el navegador GPS
- Inicia la detección de incidentes

Cuando *VAiPho Automatic* se pone en marcha, siempre sigue el mismo proceso. En primer lugar se inicia la interfaz inalámbrica móvil y se crea o se une, en caso de que ya exista, a una red ad-hoc llamada *vaipho*. A continuación se cargan de la base de datos, los datos del usuario, como los pares de claves pública y privada, clave secreta, pseudónimo, etc. Estos datos se toman de la base de datos a partir del fichero de certificados que el usuario descarga de la página web de VAiPho. Por último, tanto el cliente como el servidor inician los procesos para recibir y enviar beacons desde/hacia la red y ejecuta cuando hace falta los mecanismos de autenticación de nodos. En este punto, el sistema está listo para comunicarse e intercambiar información sobre los eventos con otros dispositivos autenticados, que están en el rango de transmisión. Todo este proceso es automático y transparente para el usuario, que sólo oye un mensaje de voz "Iniciando VAiPho", que indica que VAiPho ha comenzado y que se cargará la interfaz de *VAiPho Automatic*.

Después de establecer el sistema de comunicación, el navegador GPS se inicia y comienza a ejecutarse un sistema de detección de incidentes. El objetivo de este sistema es detectar situaciones anómalas y generar eventos con el fin de alertar a otros usuarios acerca de estas situaciones. Este proceso es automático y utiliza el software de GPS para obtener información útil para generar los eventos, como la velocidad y la posición del vehículo. Específicamente para este trabajo se ha utilizado el software de navegación GPS *Sygic Development Kit* (SDK) que permite a los desarrolladores agregar funciones de navegación

en cualquier solución software que se ejecute a través de dispositivos móviles Windows Mobile. El objetivo principal de VAiPho es detectar automáticamente los atascos en la carretera por lo que el SDK proporciona las funciones de obtención de la ubicación y la velocidad actual, así como el límite de velocidad de la carretera actual. Con esta información, nuestro detector de incidentes calcula a partir de los parámetros que recibe del SDK si el vehículo circula a una velocidad anormalmente reducida. En ese caso llega a la conclusión de que el vehículo puede estar en un atasco de tráfico. Una vez detectado el incidente, el proceso genera un evento con el nombre de la calle, la dirección del movimiento y la ubicación en la que se encuentra el incidente. Este evento se almacena en la base de datos y se transmite a otros vehículos. Gracias a la detección de estos eventos y mediante la cooperación entre dispositivos es posible saber más acerca de las condiciones de la carretera.

Cuando un vehículo recibe esta información, se consideran dos estados posibles del vehículo en relación con el incidente. El primero es el caso en el que el vehículo circula por la misma carretera y también detecta el mismo evento. Para este estado se utiliza un esquema de agregación de datos con el fin de evitar la sobrecarga de la red y confirmar o desechar la información proporcionada. El segundo estado corresponde a un vehículo que no detecta el evento, ya sea porque no está circulando en la misma calle donde se ha detectado el evento, o bien está en la misma carretera, pero a mucha distancia del evento. En este caso, podemos distinguir dos casos posibles:

- El vehículo no tiene el lugar del evento como parte de su recorrido: En este caso, el vehículo actúa como un simple retransmisor del paquete de información y se lo envía a todos los vehículos que estén en su rango de transmisión.

- El vehículo tiene el lugar del evento como parte de su recorrido: En este caso se necesita más capacidad de procesamiento. Tras determinar que el lugar del evento se encuentra en su ruta, el programa debe calcular si es mejor elegir una ruta alternativa o mantener la que tuviera. Si el sistema determina que es mejor usar una ruta alternativa, la nueva ruta tiene que ser calculada.

Para poner en práctica todos estos procesos se deben tener en cuenta muchos posibles problemas de seguridad debido a que es relativamente fácil generar eventos falsos, o

bien que algunos atacantes traten de alterar el contenido de los paquetes, o incluso denegar paquetes de retransmisión para atacar a la red. Por lo tanto, tanto las comunicaciones entre vehículos como la información sobre eventos detectados transmitidos en la red deben proporcionar evidencias de que son verdaderos. Para ello ha sido creado el módulo de seguridad, que se presentará en la subsección 4.2.5 donde explicaremos posibles ataques y cómo VAiPho resiste a cada uno de ellos.

Otra tarea importante de *VAiPho Automatic* es la detección de posibles plazas de aparcamiento. En todo tipo de ciudades y sobre todo en las grandes urbes, hay grandes problemas de falta de aparcamientos. La aplicación VAiPho puede ayudar a encontrar plazas de aparcamiento libres en tiempo real. El procedimiento es simple, cuando un usuario arranca su vehículo y comienza a circular, deja un estacionamiento libre. Así, tras arrancarse la aplicación VAiPho, el GPS del móvil se sincroniza y envía las coordenadas donde se encuentra estacionado su vehículo al resto de nodos de la red durante un corto periodo. Estas coordenadas se retransmiten como posible aparcamiento libre. Los eventos recibidos de plazas de aparcamiento tienen una fecha de caducidad rápida, entre 30 segundos y 4 minutos, configurable por el usuario. El valor predeterminado se establece en 1 minuto. El fraude o error no se puede controlar en los eventos de estacionamiento libre. De hecho hay varias situaciones identificadas en las que un vehículo puede salir de un lugar que realmente no es un estacionamiento real, tales como:

- aparcamiento privado al aire libre,
- estacionamiento en doble fila,
- estacionamiento en lugar prohibido.

Por esta razón, cuando la herramienta alerta sobre un aparcamiento, lo que indica es que se trata de una potencial plaza de aparcamiento, y no hay garantía de que esta plaza exista ni que, en caso de existir, siga estando libre cuando el vehículo interesado llegue a dicha plaza. Cuando VAiPho detecta una posible plaza de aparcamiento, lanza un mensaje de voz y la muestra en el mapa.

Por otro lado, cuando el usuario apaga el coche, se almacenan en la base de datos las coordenadas donde ha estacionado el vehículo con el fin de recordar dónde está el

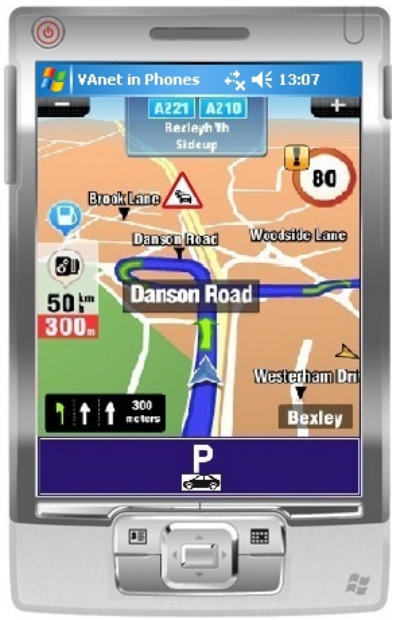


Figura 4.3: Interfaz de Conductor de VAiPho

vehiculo si el dueño se olvida. En muchas ocasiones los usuarios tienen dificultades para encontrar el lugar donde habían dejado sus vehículos estacionados, sobre todo si se mueven en grandes ciudades donde las calles le son desconocidas. En este sentido, con las coordenadas almacenadas en la base de datos, VAiPho dibujará en el mapa el lugar donde está estacionado el vehículo, lo que permitirá al usuario calcular la ruta para llegar a su vehículo con el software GPS.

VAiPho User

El diseño de la interfaz de usuario puede ser la diferencia entre la aceptación o el rechazo del producto en el mercado. Si los usuarios finales creen que la herramienta no es fácil de aprender o de usar, aunque fuese excelente podría no ser exitosa. El diseño de la interfaz de usuario puede significar que el producto sea fácil de entender y usar, lo que se traduciría en una mayor aceptación por parte de los usuarios, o bien todo lo contrario.

Se han diseñado dos interfaces simples. La interfaz que se muestra cuando el usuario no está conduciendo. (ver Fig. 4.3), permite al usuario configurar parámetros variables de

Figura 4.4: Interfaz de Peatón Sin y Con Vehículo Aparcado Almacenado

VAiPho como caducidad de las plazas de aparcamiento, nivel mínimo de batería, idioma de la aplicación, o carga de fichero de certificados generado desde la página web de VAiPho. Además, el usuario puede comprobar los eventos que se han producido y almacenado en su teléfono móvil. *VAiPho User* proporciona una práctica herramienta para ubicar el lugar donde el vehículo vinculado está estacionado. Esta utilidad es una de las más interesantes de VAiPho, pero sólo es válida si *VAiPho Automatic* estaba activado cuando el usuario estaba aparcando el vehículo y si tenía cobertura GPS. En caso contrario, la aplicación muestra un mensaje al usuario indicando que la información no está disponible. En el caso de que la aplicación tenga guardada la ubicación donde se estacionó, *VAiPho User* dispone de un botón de búsqueda de aparcamiento, *"encontrar"*, que el usuario puede utilizar para encontrar la ubicación de su vehículo aparcado. Esta aplicación inicia el navegador GPS en modo *a pie*, y un icono muestra el lugar donde se encuentra el vehículo aparcado y la mejor ruta a pie hasta el mismo.

La interfaz mostrada cuando el usuario está conduciendo es muy similar a la utilizada por el GPS convencional. Para evitar la distracción del conductor no sólo utiliza iconos

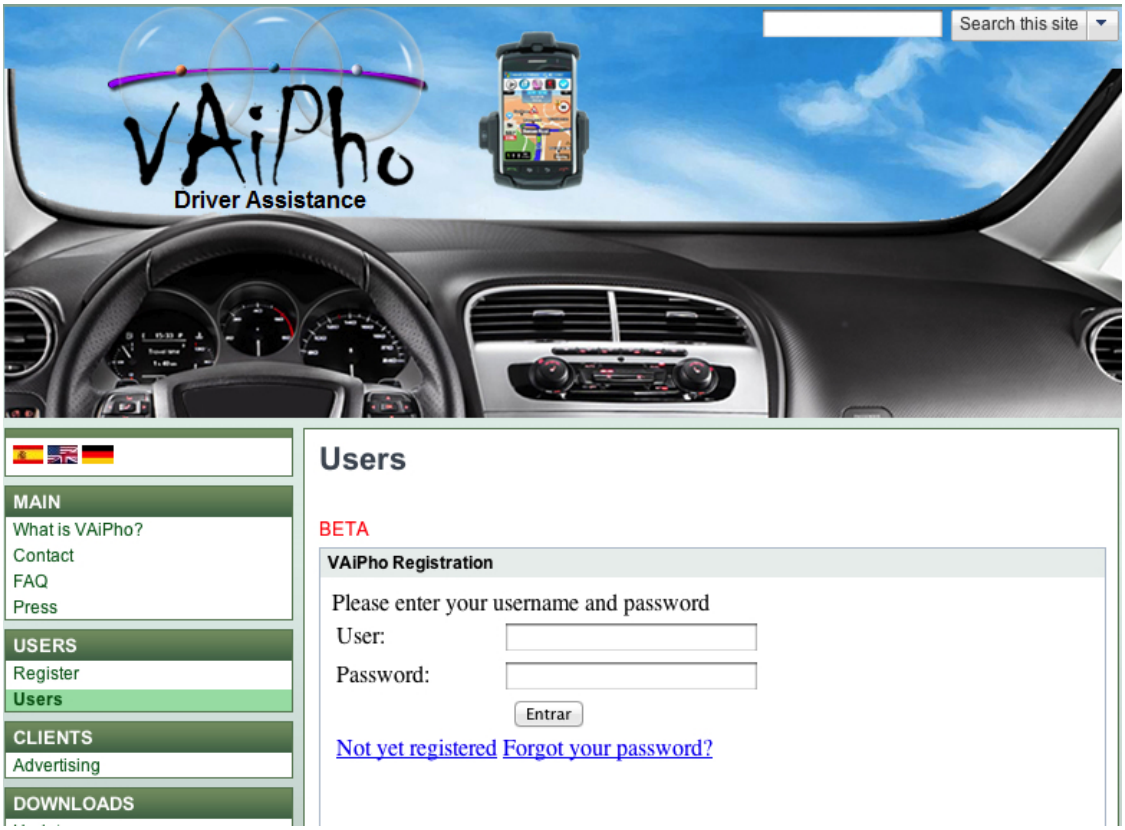


Figura 4.5: Página Web de VAiPho

en los mapas sino que también utiliza mensajes de voz. Por lo tanto, cuando se detecta un atasco de tráfico, VAiPho muestra un icono en el mapa y un mensaje de voz indica que existe un atasco en la ruta. Este mismo método se utiliza para mostrar los posibles aparcamientos cuando el usuario llega cerca del destino y presiona el botón de busqueda de aparcamiento.

4.2.4. Página Web de VAiPho

La página web [207], además de dar publicidad de VAiPho, permitirá que los usuarios descarguen la herramienta para instalarla en sus smartphones. En la Fig. 4.5 se observa que esta página tiene un menú donde se destacan los puntos de mayor interés tales como registro, descargas y FAQ (Frequently Asked Questions). Los usuarios podrán obtener VAiPho para las distintas plataformas móviles y descargar las actualizaciones de cualquier nueva funcionalidad o mejora que se desarrolle. El usuario debe descargar e instalar la

aplicación en su dispositivo y ubicar los archivos generados por la página web en la ruta que le indica dentro de su smartphone. Cuando el dispositivo inicie VAiPho, el programa actualizará la base de datos con la información contenida en esos archivos. En el menú de usuario, las personas interesadas en instalar VAiPho, en primer lugar deben registrarse. Esto les permitirá generar un archivo de claves públicas y certificados necesario para utilizar la herramienta. Además, se ofrece un servicio de apoyo donde los usuarios que quieran instalar la aplicación pueden enviar cualquier pregunta acerca de su funcionamiento. Finalmente, en la sección *clientes*, las empresas o los usuarios que estén interesados en anunciarse a través de la herramienta pueden rellenar los datos necesarios para generar el fichero de publicidad con la infomación a difundir. Ambas cuestiones, gestión de los certificados y de la información publicitaria, son comentadas a continuación.

Certificados

VAiPho es una herramienta que sirve para generar una red auto-organizada usando como base un modelo de confianza de los usuarios con sus amigos. Cuantos más certificados tenga un dispositivo firmado por otros usuarios, mayor será la probabilidad de conseguir la confianza requerida por otros nodos que se quieren comunicar con él. Por lo tanto, podemos encontrar una similitud entre nuestra red y redes sociales como Facebook, Twitter, etc. Proponemos que VAiPho utilice ideas de marketing viral para expandirse con mayor rapidez. Es decir, utilizará técnicas basadas en redes sociales pre-existentes.

Con esto en mente, cuando un usuario entra en la web para registrarse con el fin de empezar a usar VAiPho, obtiene su par de claves pública/privada y se le solicita información sobre sus contactos, al igual que lo hacen las redes sociales. Actualmente, la herramienta puede importar contactos desde una cuenta de correo. De esta manera, VAiPho busca otros usuarios registrados que sean amigos de este nuevo usuario. La lista con la intersección se muestra al usuario para que pueda confirmar si esos amigos son de confianza en la red VAiPho. Con la clave privada del usuario, el sitio web firmará en nombre del usuario las claves públicas de sus amigos y viceversa, emitiéndose los correspondientes certificados. Una vez autenticado, el usuario podrá descargar un archivo que contiene los certificados firmados por él/ella y los firmados por sus amigos para él/ella. Todos estos certificados servirán para

generar el grafo certificado de la red de confianza. Este proceso, así como la ubicación en la que el archivo debe ser almacenado se detalla en la página Web de VAiPho.

Publicidad

Todos los usuarios registrados o empresas podrán contratar con VAiPho publicidad geolocalizada. Para ello, la web de VAiPho tiene una sección llamada *clientes* en la que recogen los datos necesarios, que incluyen nombre de la empresa, mensaje, coordenadas X e Y, área de interés, fecha de vencimiento y logotipo de la empresa. El mensaje está limitado en número de caracteres para que sea fácilmente leído en la aplicación. Esta información será revisada y procesada por un agente responsable de publicidad para evitar publicidad engañosa o problemas en los mensajes. El coste del mensaje publicitario será en función del lugar (coordenadas X, Y), el área de interés y la fecha de vencimiento. Después del pago, la empresa recibirá un paquete que contiene la información y su certificado. El usuario tendrá que descargar la *aplicación de Publicidad de VAiPho* para instalarla en su dispositivo. El dispositivo será el que difunda la información publicitaria.

4.2.5. Análisis de Seguridad

La seguridad de las comunicaciones en VANETs representa un reto importante que hay que resolver ya que se espera que en el futuro estas redes supongan una revolución para la seguridad y la comodidad del transporte por carretera. En estas redes, los mensajes de advertencia pueden influir en las decisiones de conductor para que reduzca la velocidad y/o para que elija rutas alternativas basadas en la información recibida, por lo que es necesario un sistema para determinar si la información de tráfico que se pone a disposición del conductor es de confianza o no. Además, la calidad de las comunicaciones en VANETs se degrada cuando el número de vehículos que no cooperan es muy grande. Por esta razón, en VANETs la fiabilidad de las comunicaciones, junto con la privacidad y autenticidad de los usuarios son los principales objetivos a tener en cuenta. Con este fin, VAiPho integra los siguientes mecanismos de seguridad para evitar posibles ataques.

VAiPho es una red formada mediante conexiones inalámbricas limitadas por la cobertura Wi-Fi de los smartphones dentro de los vehículos. Por este motivo, no es posible

el direccionamiento. En consecuencia VAiPho estará formado por múltiples subredes que cambiarán continuamente, y los nodos que conforman estas subredes intercambiarán la información difundiéndola. Esta topología de red no asegura que la información pueda llegar a todos los puntos de la red, pero sí a los más poblados. Además este tipo de red no tiene límite en cuanto al número de nodos y distancia que soporta, puesto que siempre se pueden formar más subredes sin problema. A continuación, en sucesivas subsecciones comentamos diversos aspectos de seguridad desde los puntos de vista del usuario, la información y la red.

Seguridad de Usuario

Uno de los temas más importantes en materia de seguridad es el anonimato. Normalmente no es deseable que las partes que se comunican revelen sus identidades. Además, en una VANET un atacante podría usar la señal inalámbrica para seguir físicamente a alguien. En un sistema centralizado con 3G, la situación podría ser incluso peor debido a que un único atacante podría seguir a todos los vehículos a través de sus números de teléfono móvil de forma centralizada. Estamos hablando de una cuestión de derecho a la intimidad personal, por lo que las identidades de los usuarios nunca deben ser reveladas a fin de evitar posibles seguimientos. VAiPho nunca usa 3G por lo que el ataque descrito es imposible en nuestro esquema. Sin embargo, la señal inalámbrica sí puede ser escuchada por cualquier nodo que esté en rango. Para proporcionar anonimato de usuario se usan pseudónimos variables como identificadores.

Cada nodo cambia su pseudónimo en periodos de tiempo aleatorios y avisa de ello mediante los beacons al resto de usuarios vecinos con los que esté autenticado.

Otra importante característica de seguridad consiste en verificar que un dispositivo se corresponde realmente con un usuario legítimo de la red VAiPho. Comprobar esto en una red completamente distribuida, donde no existe ningún tipo de infraestructura para realizar un control de identidades, es muy complicado. En este trabajo se ha diseñado e implementado un mecanismo para verificar la autenticidad de los usuarios. El mecanismo de autenticación se basa en una demostración de conocimiento nulo, que es un método interactivo que un usuario realiza para demostrar a otro que sabe un secreto, sin revelar

nada al respecto. El secreto en nuestro caso es la clave pública de un nodo que ambos usuarios conocen. Por tanto, es necesario que los usuarios tengan algún nodo en común en su grafo certificado. Para hacer esto posible, los usuarios una vez autenticados intercambian sus repositorios para permitir el crecimiento y actualización de ambos grafos certificados. Además, este mecanismo de forma automática revoca los certificados de usuarios que han demostrado mal comportamiento, lo que les dejará fuera de la red en caso de mal comportamiento de forma reiterada frente a diferentes usuarios. La información sobre usuarios revocados también debe ser intercambiada después del proceso de autenticación. Los nodos dejan de estar mutuamente autenticados cuando expira un lapso de tiempo establecido sin contacto.

Seguridad de la Información

Por un lado, un atacante podría simular un atasco de tráfico inexistente con el fin de convencer a los demás usuarios de que no elijan una ruta para de esa manera, tener el camino libre de vehículos. Por esta razón, VAiPho utiliza un mecanismo de agregación de datos que evita este tipo de fraude. El primer dispositivo que detecta una situación anormal, como una velocidad mucho menor de lo esperado durante mucho tiempo, envía una advertencia de atasco de tráfico a sus nodos vecinos. Si los nodos vecinos también detectan que la velocidad es anormal, firman la información recibida lo que corrobora la información. Cuando el dispositivo promotor recibe un número mínimo de firmas, las agrega en un paquete con la información y lo envía a todos los vecinos para que difundan el mensaje.

Por lo tanto, los atascos de tráfico deben ser detectados por varios vehículos, que deberán firmar el evento de tráfico con su clave privada con el fin de agregar todos ellos en un solo paquete. De esa forma nos aseguramos de que no ha sido sólo un vehículo el que ha detectado el incidente, sino que ha sido corroborado por varios vehículos. Este mecanismo elimina la posibilidad de difundir falsos atascos de tráfico creados por un solo atacante y también evita la posible confusión generada por el sistema cuando un vehículo se detiene a un lado de la carretera debido a un pinchazo o porque el vehículo se ha roto por ejemplo.

El número de firmas necesarias dependerá de la expansión de la herramienta, es decir, cuanto mayor sea el número de vehículos con VAiPho, mayor es el número de

firmas requeridas. Esto significa que cuanto mayor sea el número de vehículos que forman la red VAiPho, menor será la posibilidad de un posible ataque. Con el fin de calcular el número de firmas requeridas, VAiPho comprueba el tiempo de su primera autenticación con otros usuarios en cada viaje y calcula el número promedio de usuarios por minuto. En la implementación actual, este promedio es menor que uno por minuto, por lo que el número de firmas requerido es de dos, que es el mínimo. Si el número está entre uno y cuatro nodos autenticados por minuto, el número de firmas requeridas será de tres. Si el promedio de autenticaciones es superior a cuatro por minuto, el número de firmas requeridas será de cinco, etc.

Seguridad de la Red

Hoy en día la conectividad inalámbrica utilizada por VAiPho es a través de *IEEE 802.11b/g*, pero en el futuro se espera que sea mediante *Wi-Fi direct* o bien usando el estándar *IEEE 802.11p*. Los paquetes de información se intercambian mediante dispositivos en los vehículos. Un atacante podría intentar que la comunicación inalámbrica fallase, lo que podría causar que la VANET no funcionase y no pudiese proporcionar servicios tales como el reenvío de paquetes. En este sentido, los atacantes intentarían provocar un ataque por denegación de servicio pasivo. Por este motivo, VAiPho debe asegurar que sólo los dispositivos registrados y no revocados de la red, que ayuden en su correcto funcionamiento se beneficien de la información transmitida por la misma y tengan acceso a la retransmisión de mensajes. VAiPho evita que los usuarios que pretendan realizar un ataque pasivo, degraden la funcionalidad de la herramienta y la conectividad de la red impidiéndoles el acceso a la red.

Además del procedimiento de autenticación mencionado, VAiPho dispone de un mecanismo específico basado en el intercambio de datos cifrados como método para captar usuarios cooperativos para la red. Este procedimiento impide que los nodos pasivos ajenos a la red puedan beneficiarse de la información transmitida, lo que mejora la eficacia y seguridad de las comunicaciones en las VANETs.

4.2.6. Simulación e Implementación

En esta sección se presenta tanto la simulación software como la configuración de la implementación real de la aplicación realizada.

El objetivo principal de las pruebas realizadas ha sido evaluar si el desarrollo de las VANETs a través de los teléfonos móviles podría funcionar en la vida real, contando con características específicas de las VANETs tales como arquitectura híbrida, alta movilidad, topología dinámica, problemas de escalabilidad, y comunicaciones intermitentes e impredecibles. La primera prueba realizada fue comprobar que las comunicaciones entre los teléfonos inteligentes con Wi-Fi usando el estándar IEEE 802.11b/g eran viables mediante una sencilla aplicación cliente-servidor entre dispositivos, al circular con vehículos en entornos urbanos o autopistas a diferentes velocidades y con diferente número de dispositivos. Estas pruebas fueron satisfactorias. Después de esto, creamos varias simulaciones mediante el uso de SUMO y NS-2. Finalmente se implementó el prototipo de VAiPho usando smartphones y poniéndolo a prueba con vehículos en atascos y otras situaciones reales. A continuación en dos subsecciones mencionamos algunos aspectos relevantes de la simulación y de la implementación real.

Simulación

Tanto la viabilidad teórica como la eficacia del enfoque de VAiPho se muestran a través de la Fig. 4.6 en una simulación del ejemplo de su funcionamiento. En la primera parte, una imagen producida con el simulador NS-2 [159] y SUMO [194] muestra el estado de la VANET en un momento en el que VAiPho está en funcionamiento. Las opciones más relevantes seleccionadas para la simulación han sido: número total de vehículos: 600 - 15000, número de vehículos con dispositivos VAiPho: 1 % -100 %, tiempo de simulación: 100-216000 segundos, período de autenticación: 20 segundos, y distancia de transmisión entre nodos: 75 metros.

Las simulaciones implementadas permiten saber el número de conexiones generadas en la red dependiendo del porcentaje de vehículos que tienen el dispositivo, de forma que de esos datos se puede extraer cuál es el número mínimo de vehículos con dispositivo

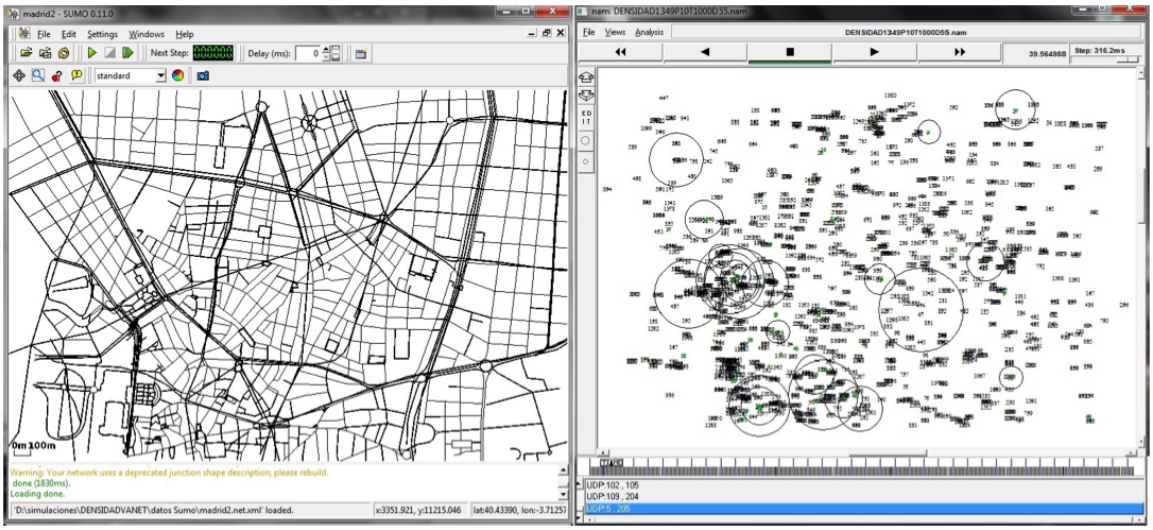


Figura 4.6: Simulación VAiPho con SUMO y NS-2 en Madrid

necesarios para que la red empieze a intercambiar información de forma fluida.

Por otro lado, las simulaciones permiten conocer a partir de qué porcentaje de vehículos con dispositivo empieza a decaer la calidad de las comunicaciones por interferencias en el canal. Las simulaciones realizadas no tienen en cuenta posibles interferencias con otros dispositivos u otras WLAN que pudiesen afectar a las comunicaciones inalámbricas por lo que es posible que este umbral obtenido en simulaciones se reduzca en entornos reales. En la Fig. 4.7 se observa que en general el incremento de porcentaje de vehículos con VAiPho mejora la eficacia de la red en la retransmisión de mensajes ya que proporcionalmente son muy pocos los paquetes perdidos en relación con los enviados.

Implementación en Smartphones

Para la evaluación práctica hemos implementado completamente la aplicación VAiPho en smartphones con Windows Mobile usando *C#*.NET. Un vídeo del prototipo, que se puede encontrar en la página Web de VAiPho [207], muestra cómo funciona la herramienta en diferentes circunstancias.

La herramienta ha demostrado ser eficaz en el reconocimiento de atascos de tráfico, la detección de aparcamientos y la búsqueda del vehículo aparcado. También se ve en

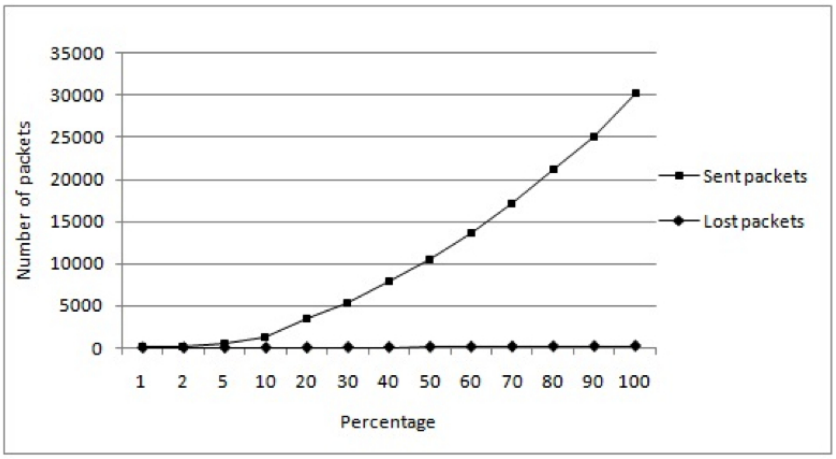


Figura 4.7: Número de Paquetes Generados y Perdidos en una Simulación VAiPho

el video cómo se autentican los dispositivos, protegiendo la privacidad del usuario usando pseudónimos variables e intercambiando información secreta con otros dispositivos. Los dispositivos que ejecutan VAiPho reciben información, que muestran por la pantalla y mediante voz al conductor, y la retransmiten al encontrarse con otros nodos, tras autenticarse con ellos.

En particular los ensayos mostraron un buen rendimiento en:

- Atasco de tráfico. Un dispositivo lo detecta automáticamente y envía un mensaje de advertencia. Otro dispositivo que esté cerca en la misma ruta y dirección, detecta el mismo atasco de tráfico y recibe la señal emitida por el primer dispositivo. En ese momento firma el mensaje recibido y lo devuelve al dispositivo que lo generó, que agrega el mensaje con ambas firmas y lo vuelve a retransmitir. Un tercer dispositivo recibe el mensaje firmado, comprueba la validez de las firmas y muestra el atasco de tráfico por la pantalla junto a la posibilidad de elegir una ruta alternativa. Este tercer dispositivo guarda la información en la base de datos y la retransmite cuando se encuentra con otros nodos.

- Detección de aparcamiento. En la simulación, un vehículo con VAiPho, que sale de una plaza de aparcamiento envía esta información a otros vehículos con VAiPho, los cuales la muestran por pantalla y retransmiten la información a otros. En particular

cuando se arranca uno de los vehículos, el dispositivo se sincroniza con el GPS y guarda las coordenadas del vehículo en la base de datos. A continuación, se firman las coordenadas con su clave privada y las envía a otros dispositivos con los que se vaya autenticando, siempre que no haya expirado el tiempo de validez del aparcamiento que ha liberado. Los nodos que reciben esta información, que hayan iniciado la búsqueda de aparcamiento, verán un icono con el posible aparcamiento dibujado en su mapa.

- Localización del vehículo estacionado. Un usuario con la herramienta VAiPho aparca su vehículo y se aleja caminando del vehículo aparcado. A continuación, el usuario inicia la interfaz de usuario de VAiPho y presiona el botón que le recuerda la posición en la que está estacionado su vehículo. A continuación la herramienta muestra la ruta a pie hasta el vehículo.

- Publicidad geolocalizada. Un nodo que recibe información publicitaria en el ámbito de validez muestra el icono en el mapa y su respectivo mensaje de voz.

4.3. Fusión de Subredes

Actualmente existen infinidad de artículos que tratan sobre redes móviles, ya sean MANETs o VANETs. La mayoría de estos artículos son teóricos, algunos se apoyan en diversas herramientas de simulación y unos pocos muestran desarrollos en dispositivos reales. La tecnología actual más expandida y conveniente para crear una red móvil inalámbrica es la tecnología Wi-Fi basada en el protocolo IEEE 802.11xx. No obstante, como ya se ha mencionado, este protocolo no está diseñado para trabajar en VANETs y genera numerosos problemas al usarlo en este ámbito. Entre esos problemas está el que surge cuando se intentan unir varias subredes. Los dos problemas principales a la hora de unir subredes son la incapacidad de comunicación si las subredes se encuentran en distintos canales y la duplicación de IPs. La invención descrita en este capítulo denominada VAiPho resuelve ambos problemas mediante las técnicas descritas a continuación.

Hay muchos trabajos publicados que tratan de resolver el problema de las diferencias en los canales de las subredes analizado en esta sección como es el caso de [211], donde

los autores tratan de utilizar varios canales simultáneamente, o de [5], que muestra cómo calcular el ruido en el canal y cómo evitarlo mediante el uso de lógica difusa y el algoritmo de Viterbi. También está el trabajo [129], que analiza el impacto de los errores de estimación del canal debido al ruido y la interferencia por la existencia de multiples usuarios. Por otra parte, los autores de [217] proponen un esquema de cooperación eficiente entre múltiples canales de retransmisión múltiple y algoritmos heurísticos de tiempo polinomial para eliminar las interferencias cuando la sincronización a nivel de transporte no esté disponible y todos los nodos utilicen un esquema de decodificación avanzado. La referencia [215] propone un marco de análisis para redes inalámbricas cognitivas multi-radio y multi-canal. Bajo este marco, el enrutamiento de datos, la asignación de recursos y la planificación de tareas fueron conjuntamente diseñados para maximizar una función de utilidad de la red que define su rendimiento.

Otros trabajos tratan de resolver diferentes problemas que tiene el protocolo IEEE 802.11. Tal es el caso de [51], donde los autores proponen una aplicación que mediante la tecnología Bluetooth permite resolver el problema de configuración de los terminales en una red ad-hoc basada en el protocolo IEEE 802.11 mediante la ampliación de la conectividad de los nodos a través del reenvío de paquetes, para evitar la necesidad y el uso de una infraestructura fija. En [124] los autores comprueban el rendimiento del protocolo 802.11b en este tipo de configuraciones utilizando técnicas de modelado analítico. En la referencia [137] se propone un esquema de protección para los ataques de denegación de servicio en los protocolos de seguridad WEP, WPA y WPA2 mediante la resolución de un problema con los marcos de control. Los autores de [131] hacen un estudio de la relación del reloj con el ahorro de energía usando IEEE 802.11 para redes ad hoc y proponen un protocolo de comunicación para mejorar los resultados. Finalmente, [158] propone y simula la estimación de los parámetros de la señal utilizando técnicas de invariancia rotacional con el fin de mitigar la interferencia del Bluetooth con la fase de estimación de canal en el estándar IEEE 802.11g.

Todos estos trabajos intentan resolver diferentes problemas del protocolo IEEE 802.11, pero ninguno de ellos analiza el problema que aquí se intenta resolver. El trabajo más cercano al tema que aquí se analiza es [198], donde se propone el uso de redes punto

a punto multi-salto a través del protocolo convencional IEEE 802.11. En ese trabajo los autores comprueban el tiempo de resincronización de estados transitorios que se generan cuando se fusionan múltiples redes ad-hoc y proponen una disminución de los tiempos de resincronización para reducir el consumo de energía. Sin embargo, dicho trabajo no incluye ningún tipo de análisis de implementación en dispositivos reales.

4.3.1. Planteamiento del Problema

Un requisito fundamental para que VAiPho funcione es que cada dispositivo sea a la vez cliente y servidor en la misma red inalámbrica. En los actuales equipos móviles que pueden conectarse a través de Wi-Fi se utiliza normalmente el protocolo IEEE 802.11. Gracias a ello, los dispositivos tienen la capacidad de comunicación libre con otros dispositivos de corto y mediano alcance (hasta 400 m.). Por lo tanto, para realizar esta tarea cada dispositivo debe comprobar si la red inalámbrica ya está creada o no, y luego, si existe la red, el dispositivo debe conectarse a ella. De lo contrario, el dispositivo crea una nueva red inalámbrica. La creación de estas nuevas redes inalámbricas VAiPho en entornos reales puede causar muchos problemas debido principalmente a la formación de subredes desconectadas, que habría que unir llegado el caso. Los dos problemas más importantes que surgen son: que dos o más subredes no puedan ni siquiera verse una a la otra porque están en canales diferentes, o que dos o más subredes se encuentren en el mismo canal, pero haya dispositivos con la misma dirección IP.

La solución implica que las subredes tienen que estar en el mismo canal y sin conflictos de IP para poder combinarse sin ningún problema. Sin embargo, este caso no es el más habitual. La situación más usual es la primera, porque la elección del canal es al azar. Por otra parte, el canal con el número más bajo utiliza una frecuencia más baja, y con una frecuencia más baja, la propagación es mejor porque hay menos atenuación. Sin embargo, el problema de la diferencia en la atenuación no es comparable con la pérdida de paquetes en el canal debido a la interferencia entre redes en canales adyacentes. Hay canales que están más saturados que otros. Tal es el caso de los canales 6 y 11. También existen interferencias con Bluetooth y otros sistemas que utilizan los 2,4 GHz de la banda ISM (Industrial, Scientific and Medical). Los canales de retransmisión que son más comunes en

el protocolo IEEE 802.11xx son los del tipo *a, b, g* y *n*. Su rango va desde el canal 1 al 11 en los Estados Unidos, del 1 al 13 en Europa y del 1 al 14 en Japón. Los canales comparten parte de su ancho de banda.

Figura 4.8: Canales en Europa

La Fig. 4.8 muestra los canales que usan las redes inalámbricas en Europa. Los dispositivos pueden optar por crear la red inalámbrica en cualquiera de estos canales. La figura también muestra el número máximo de redes que puede haber en distintos canales sin interferencias. En particular, cinco es el número máximo de redes inalámbricas sin interferencias con redes vecinas.

Las distintas ramas del protocolo IEEE 802.11xx ofrecen la ventaja de ser compatibles entre sí, de modo que el usuario no necesita nada más que su adaptador Wi-Fi integrado para conectarse a la red. El exceso de saturación en un canal es causa de colisiones de paquetes en la transmisión de datos, lo que se corresponde con una baja velocidad de transferencia.

En esta sección se propone una solución para el problema de la combinación de subredes cuando dos instancias de la misma red se encuentran en diferentes canales. La solución ideal implica que las subredes más pequeñas se integren en la más grande, pero el problema es que los nodos no conocen el tamaño de las otras subredes. Esta sección describe un algoritmo para calcular el tiempo que los dispositivos de una subred necesitan para restablecer su interfaz y combinarse con otras subredes, en función del tamaño de la subred. También, al final de la sección se utilizan reglas de lógica difusa para definir cómo afecta la interferencia entre canales.

4.3.2. Propuesta Determinística

Si un nodo detecta un conflicto de IP, restablece su interfaz de red y esto es suficiente para resolver el problema. Sin embargo resolver el otro problema mencionado, de la fusión de subredes, es mucho más complejo.

Para resolver dicho problema se propone aquí restablecer la interfaz inalámbrica de los dispositivos que pertenecen a una de las subredes inalámbricas para que se unan a otra de las subredes. La forma óptima para restablecer la subred podría ser mediante el ajuste de las subredes que tienen menor número de dispositivos, pero esta información no es transparente para los dispositivos de cada subred, que sólo pueden saber cuántos dispositivos hay en su subred, pero nunca pueden conocer cuantos dispositivos hay en otras subredes.

A continuación se describen dos aproximaciones de solución al problema, en primer lugar con un algoritmo básico y luego con uno optimizado.

Propuesta Básica

La primera aproximación de solución propuesta se basa en la elección de la subred en el canal más adecuado para minimizar la interferencia que generan otras redes en los canales adyacentes. Para ello, en el algoritmo propuesto, el dispositivo comprueba regularmente las redes existentes en busca de nodos que estén a su alcance de transmisión. Se consideran nodos de la VANET a los vehículos con OBU en su interior, que puede ser cualquier dispositivo móvil que ejecute la aplicación *VAiPho*. Si hay dos instancias de la red *vaipho*, todos los dispositivos pertenecientes a las subredes con mayor interferencia deben restablecer su interfaz inalámbrica para unirse a la otra instancia. El primer nodo que detecta esta situación y que pretenece a una de esas subredes envía un mensaje de aviso al resto de nodos de la red con el objetivo de que reinicien su interfaz inalámbrica. Por lo tanto, a la recepción de este aviso, los nodos de su subred después de retransmitir el mensaje al resto de nodos, apagan su interfaz de red durante un segundo y luego vuelven a activarla. Entonces, como la red *vaipho* ya existe cuando los nodos reinicien dicha interfaz, se conectarán a esta red y el problema será resuelto. Los nodos se mantendrán autenticados con los nodos con los que estaban autenticados en la anterior subred, ya que los datos de autenticación no varían con el cambio de subred, que tiene exactamente el mismo nombre, *vaipho*.

Algoritmo Elección de subred con menos interferencia

```
00: ...
01: //Hay redes != propia que tienen menos interferencia
02: int[] channelsDifference = new int[nNetworks];
03: //inicia channelsDifference para ver cual tiene menos diferencia
04: for (int i = 0; i < VaiphoInstances; i++)
05:   channelsDifference[i] = 50;
06: for (int i = 0; i < nNetworks; i++)
07:   for (int j = 0; j < nNetworks; j++)
08:     if (network[i].Equals(vaipho))//compara unas redes vaipho con otras
09:       if ((i != j) &&
09:       (Abs(channels[i] - channels[j])) < channelsDifference[i])
10:         channelsDifference[i] = Abs(channels[i] - channels[j]);
11:   //diferencia de canales con la red vaipho
11:   //las que no tienen la mayor diferencia deben reiniciarse.
12:   for (int i = 0; i < nNetworks; i++)
13:     if ((biggestDifference < channelsDifference[i])
13:     &&(channelsDifference[i]!=50)) //guarda la mayor diferencia
14:       biggestDifference = channelsDifference[i];
15:       nodePositionBiggestDifference = i;
16:     if (nodePositionBiggestDifference != numberOwnNetwork)
16:     //si no tiene la mayor diferencia
17:       networkDetachProtocol(vaipho);
18:       return true;
19: ...
```

Propuesta Optimizada

Los dispositivos pueden saber exactamente el número de nodos que se encuentran en su subred, el canal donde se encuentra, en qué canales hay otras instancias de la red

vaipho, y qué canales están siendo utilizados por otras redes que puedan interferir con la subred *vaipho*. Con esos datos deberían escoger la subred más apropiada, que depende principalmente del número de nodos que contenga, y de la interferencia entre canales utilizados por otras redes. Si el número de dispositivos de dos subredes coincide aproximadamente, la interferencia en los canales será lo que determine qué subred se reiniciará.

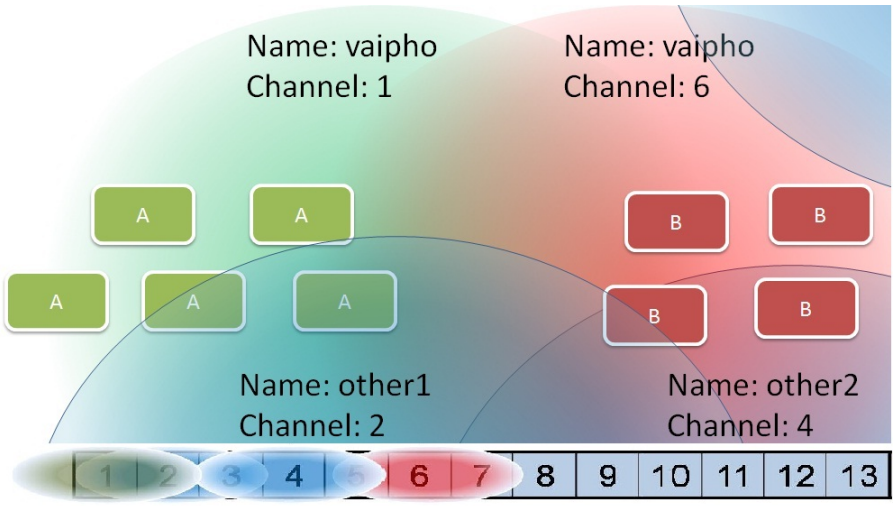


Figura 4.9: Ejemplo de Situación Habitual

La Fig. 4.9 muestra un ejemplo típico que puede darse en una VANET donde los nodos son vehículos con unidades a bordo que se están moviendo a través de escenarios urbanos. En este ejemplo, dos subredes A y B formadas en la misma vía pero en diferentes canales, entran en el mismo rango de emisión. Por esta razón, los dispositivos de una subred no pueden ver a los dispositivos que están en la otra subred. Además, hay interferencias de otras redes inalámbricas en esos canales. Sin embargo, aunque no puedan ver la composición de otras subredes, cada nodo de una subred sí puede ver a otros nodos de otras subredes por lo que puede comprobar las interferencias que hay en su canal y en los canales en los que se encuentran los otros nodos y compararlos. La Fig. 4.9 muestra un ejemplo de la idea anterior, donde la subred A está en el canal 1, la subred B está en el canal 6, y existen otras dos subredes cercanas en los canales 2 y 4. Por lo tanto, la subred A está en el canal 1, que está a una distancia 1 de la subred más cercana (canal 2), y la subred B está en el canal 6, que está a una distancia 2 de la subred más cercana (canal 4). De ahí se concluye que la

Tabla 4.1: Dispositivos Móviles

Nombre del Teléfono	Plataforma	Velocidad de CPU	RAM	ROM	Capacidad de Batería
HTC HD Mini	WM 6.5	600 MHz	384 MB	512 MB	1200 mAh
HTC P3300	WM 5.0	201 MHz	64 MB	128 MB	1250 mAh
HTC Touch2	WM 6.5	528 MHz	256 MB	512 MB	1100 mAh
hp IPAQ 614C	WM 6.0	520 MHz	128 MB	256 MB	1590 mAh

subred A tiene más posibilidades de tener interferencia que la subred B, luego ésta deberia ser la que se reinicie.

Sin embargo, como se mencionó anteriormente, la solución óptima consistiría en restablecer las subredes con menos dispositivos pero dada la falta de información disponible para los nodos, su única opción es tomar decisiones heurísticas en base a estimaciones y lógica difusa. El esquema concreto será descrito más adelante.

4.3.3. Implementación con Dispositivos Reales

Para comprobar realmente la eficacia del proceso básico descrito para fusionar subredes, la mejor opción sería contar con dos subredes a gran escala con dispositivos reales. Sin embargo, hacer esto con un gran número de dispositivos es imposible dado que no contamos con tantos recursos. Por lo tanto, la alternativa de análisis elegida ha sido tomar datos obtenidos a partir de dispositivos reales para luego realizar simulaciones con NS-2 utilizando estos datos reales.

La implementación en dispositivos reales se ha desarrollado mediante el uso de teléfonos móviles inteligentes con Windows Mobile 5 y 6. La plataforma utilizada para el desarrollo es Microsoft Visual Studio 2008. El pseudocódigo correspondiente al algoritmo básico implementado para la detección de subredes vaipho se muestra a continuación.

Algoritmo Detección de Subredes VAiPho

```
00: function CheckingSubnetworks (...)
01:    ...
02:    foreach (AccessPoint access in NearbyAccessPoints)
03:       network[nNetworks] = access.Name.ToString();
```

```
04:     channels[nNetworks] = access.Channel;
05:     MacDirection[nNetworks] = access.PhysicalAddress.ToString();
06:     if (network[nNetworks].Equals(vaipho))
07:       VaiphoInstances++;
08:       //guarda el nº de canal con la mayor subred
09:       if (channels[nNetworks] > biggestChannel)
10:         biggestChannel = channels[nNetworks];
11:       //guarda la subred
12:       if (MacDirection[nNetworks].ToString().Equals(MacOwnDirection))
13:         numberOwnNetwork = nNetworks;
14:       nNetworks++;
15: ...
```

La Fig. 4.10 muestra una captura de pantalla de dos emuladores que utilizan la misma subred *vaipho*, mientras existen otras subredes *vaipho* en otros canales. En este ejemplo, el dispositivo de la derecha es el primero en detectar la presencia de varias instancias de la red *vaipho*. Después de comprobar que hay otras subredes *vaipho*, este nodo inicia el protocolo de fusión de subredes. En el protocolo de fusión, el primer nodo que detecta la otra subred *vaipho* envía el mensaje de advertencia al resto de nodos de su subred con el fin de reiniciar su interfaz de red. Después de esto, cada nodo se desconecta de su interfaz durante 1 segundo y cuando vuelve a activar su interfaz de red, se conecta a la subred *vaipho* que ya existe.

4.3.4. Análisis de Rendimiento

La Fig. 4.11 muestra varias redes inalámbricas en el rango de transmisión en una de las implementaciones reales realizadas con objeto de analizar el rendimiento de nuestra propuesta. En particular, se puede ver que hay dos instancias de la red *vaipho*. La primera está en el canal 1, mientras que la segunda está en el canal 11. Se puede ver también que hay otras subredes en los canales 1 y 11. Esta situación implica la necesidad de fusionar ambas subredes.

Figura 4.10: Implementación en Dispositivos Reales

Con el fin de comprobar el tiempo medio que cualquier subred tardaría en integrarse en otra subred, se llevaron a cabo numerosas simulaciones con diferentes tamaños de subredes. En estos casos, una de las subredes existentes se fusionaba con otra, y para conseguirlo, siempre el primer nodo que detectaba que había dos instancias de la misma red vaipho, comenzaba a transmitir información al resto de nodos de su subred para reiniciar sus interfaces de red y de esa forma, conectarse a la red *vaipho* existente.

El tiempo que los nodos tardan en reiniciar su interfaz de red y conectarse a la otra instancia de la red en el canal correcto se obtuvo de ejecuciones con dispositivos reales, mediante la media del tiempo que tardaban. Los tiempos obtenidos van desde *2,94* segundos registrado como el mejor de los casos para los dispositivos *HTC Mini HD*, y *9.15* segundos registrado como el peor de los casos y que se obtuvo con el dispositivo *HTC P3300*. Por lo tanto, la aplicación tarda entre *2,94* y *9,15* segundos en reiniciar su interfaz de red y conectarse nuevamente. Los tiempos que dependen del protocolo de autenticación utilizado, no fueron incluidos en estas medidas. Por lo tanto, el tiempo estimado sólo incluye el tiempo que se tarda en apagar la interfaz de red más un segundo de espera más el tiempo en volver a conectar el dispositivo con la red *vaipho* existente. Después de estas ejecuciones con

Figura 4.11: Instancias de la Red Vaipho

dispositivos reales, los datos obtenidos fueron utilizados para una simulación con NS-2 a gran escala.

La Fig. 4.12 muestra la simulación de una carretera con tres carriles. En esta simulación, dos subredes se señalan con distintos colores. De color rojo está el nodo que detecta las dos instancias de la misma red, y envía el mensaje de advertencia para restablecer la interfaz de red a todos los nodos de su subred.

La Fig. 4.13 muestra resultados numéricos obtenidos de las simulaciones en NS-2. Estas simulaciones se realizaron con subredes de tamaños entre 10 y 140 nodos. Se calculó el tiempo medio que los nodos tardan en volver a conectarse a la red con menos interferencias. Se distinguen diferentes niveles en función del tamaño de la subred. En esta figura, se utiliza "nivel" para referirse a un subconjunto de nodos que se encuentran en el rango de transmisión del mismo nodo transmisor, por lo que el primer nodo que detecta que hay varias instancias de la red *vaipho* envía un mensaje de advertencia. Cuando la advertencia llega al subconjunto de nodos que están en su rango de transmisión, retransmiten la información y luego reinician su interfaz de red.

El grupo de nodos a los que les llega entonces la información serán de otro nivel, y así sucesivamente. La Fig. 4.13 muestra el tiempo medio que tardan los nodos de cada nivel en volver a conectar con la otra subred *vaipho*. Las simulaciones muestran un resultado obvio: cuanto mayor sea el tamaño de la subred, mayor es el tiempo que tarda la subred en volverse a conectar. Sin embargo, si se distinguen los nodos en diferentes niveles de

Figura 4.12: Simulación NS-2 de la Propuesta

retransmisión se puede observar que aproximadamente tardan el mismo tiempo desde el nodo origen, independientemente del número de nodos que tenga la subred.

4.3.5. Propuesta Basada en Lógica Difusa

El número de paquetes generados y perdidos siempre será menor si la subred cuyos dispositivos reinician su interfaz de red es la subred que tiene menor número de nodos. Esto queda confirmado en la Fig. 4.14, que muestra el número de paquetes generados y perdidos tras la reinicialización de subredes con diferente número de nodos, obtenidos en las simulaciones descritas en la sección anterior. Partiendo de esta premisa trabajamos para diseñar un método que permita, sin saber el número de nodos que hay en cada subred, hacer que sea la subred con menos dispositivos la que se reinicie. Además de esto, como ya se ha mencionado, otro factor importante que se debe tener en cuenta es la interferencia que tienen los canales utilizados. A partir de la interferencia en cada canal se logra calcular un

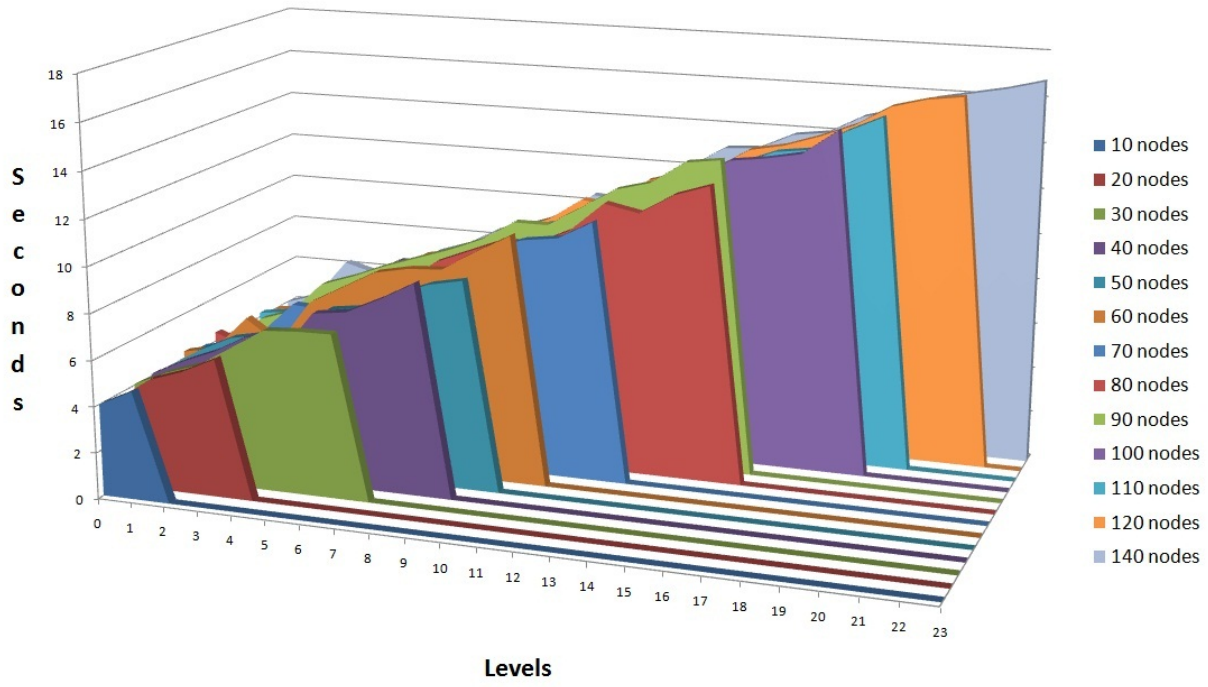


Figura 4.13: Retardo al Combinar Subredes

factor de corrección para el tiempo que cada subred va a esperar para iniciar el protocolo de fusión con otra subred. Se debe minimizar este tiempo para el nodo que va a insertarse en la otra subred, pero procurando que los nodos pertenecientes a las distintas subredes no reinicien sus interfaces de red al mismo tiempo.

Tomando los valores reales del tiempo que tarda cada nodo de una subred en reinsertarse en otra subred (véase Fig. 4.15), se ha calculado el tiempo que tarda la subred completa en insertarse en otra, dependiendo del número de nodos de la subred. El tiempo que cada nodo necesita para desvincularse de su subred y unirse a otra subred depende del número de nodos que haya en la red. Concretamente, dicho tiempo $W(x)$ puede ser aproximado linealmente por la siguiente fórmula, donde S_r es la desviación estándar residual que mide la variación promedio de los datos de tiempo sobre la línea de regresión, y se denota el número de nodos de la subred:

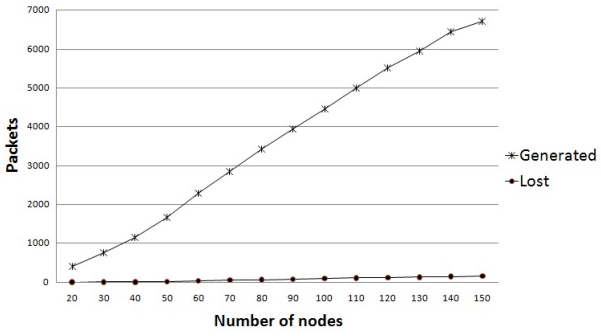


Figura 4.14: Número de Paquetes Generados y Perdidos

$$W(x) \approx 0.107x + 3.63 \pm S_r \; si \; x > 0 \tag{4.1}$$

A partir de esta fórmula es posible estimar el momento óptimo antes de repetir el procedimiento de verificación sobre la existencia de varias instancias *vaipho*.

Se propone el uso de una expresión lineal mediante la eliminación de la constante de tiempo inicial de $3,63$ en la fórmula 4.1 como punto de partida para la toma de decisiones. De esta forma, un nodo que detecta que hay varias instancias de la red *vaipho* esperará un tiempo determinado dependiendo del número de nodos que haya en su subred que está caracterizado por la fórmula $W(x) = 0.107x$, donde x es el número de nodos de su subred.

Por ejemplo, un nodo que detecta varias instancias de la red *vaipho* y que pertenece a una de ellas, compuesta de dos nodos, esperará *0.214* segundos antes de volver a comprobar si existen varias instancias de la red *vaipho*. Otro nodo que detecta las mismas subredes, pero que pertenece a una subred de 20 nodos esperará *2,14* segundos antes de volver a comprobar. De esta forma, los nodos de la subred menor tienen tiempo suficiente para reiniciarse y unirse a la red mayor antes de que los dispositivos de esta subred reinicien su interfaz.

Además de tener en cuenta el número de nodos que hay en cada subred, en casos de subredes de similar tamaño se puede considerar además la interferencia entre canales para que subredes que se encuentran en un lugar con menos interferencias prevalezcan sobre otras

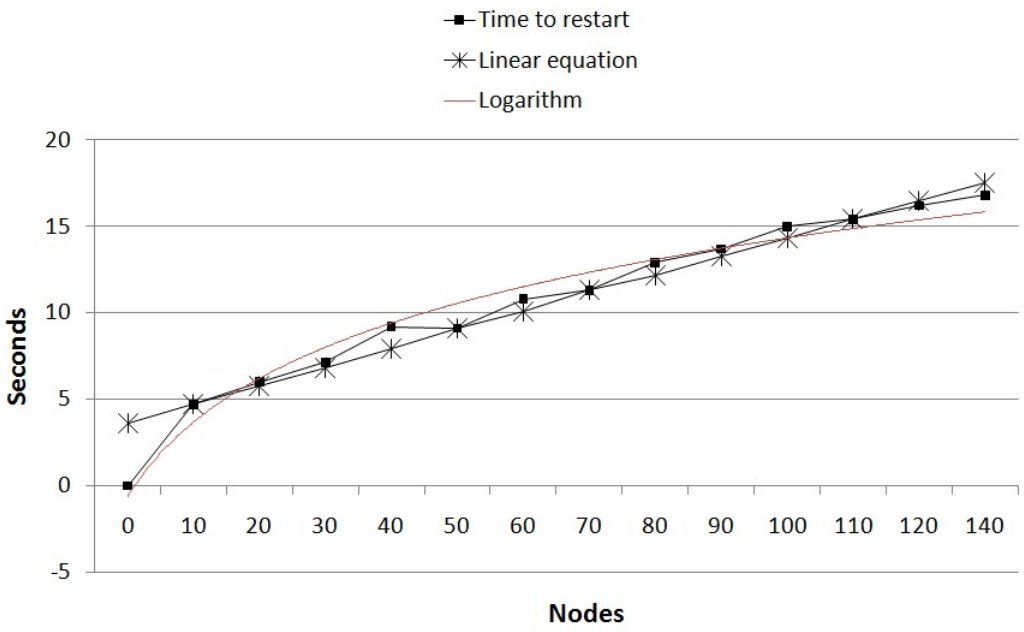


Figura 4.15: Tiempo de Retardo para Combinar una Subred con Otra

subredes con más interferencia.

Una subred operando en un canal m no tiene interferencia si no hay otras subredes que estén funcionando en el mismo canal o en canales a menos de dos canales de distancia. La interferencia de un canal se mide en dBm, que es una unidad de medida utilizada en telecomunicaciones para expresar la potencia absoluta de una señal de red con nivel de potencia P que utiliza el canal c en un momento dado t a través de la siguiente relación logarítmica, que toma valores negativos:

$$dBm(c(t)) = 10 * log\frac{P}{1mW} \tag{4.2}$$

Mediante el uso de esta medida, la interferencia de un canal c en un momento dado t se puede estimar mediante la siguiente expresión 4.3, donde N denota el número de subredes que utilizan el mismo canal c (aquí denotados c_i), y M y Q denotan respectivamente el número de subredes dentro de 1 o 2 canales de distancia al canal c (aquí denotados c_j y c_k,

respectivamente), y a se obtiene de la expresión $\frac{2*(-a)}{dBc_{alta}} = 1$ donde dBc_{alta} corresponde al canal con más interferencia.

$$dBm(c(t)) \approx \sum_{i=1}^{N} \frac{-a}{dBm(c_i(t))} + \sum_{j=1}^{M} \frac{2*(-a)}{3*dBm(c_j(t))} + \sum_{k=1}^{Q} \frac{-a}{3*dBm(c_k(t))} \tag{4.3}$$

Esta expresión proviene de que: si dos subredes operan en el mismo canal, la coincidencia es total, si se encuentran en dos canales adyacentes la coincidencia es de $\frac{2}{3}$ de la frecuencia que utilizan, y si están a dos canales de distancia la coincidencia es de $\frac{1}{3}$ de la frecuencia que utilizan. Téngase en cuenta que el valor de a podría variar dependiendo del lugar donde los vehículos estén, y que si se encuentran en un lugar donde hay muchas redes inalámbricas, la posibilidad de tener más de dos subredes que utilicen el mismo canal es mayor.

En este punto se propone la utilización de una aproximación basada en lógica difusa para utilizar el grado exacto de diferencia de interferencia existente en dos canales en los que se encuentran dos subredes *vaipho* para estimar de forma difusa si se considera que hay o no interferencia suficiente como para incrementar o no el tiempo de espera necesario para que cada nodo reinicie su interfaz de red.

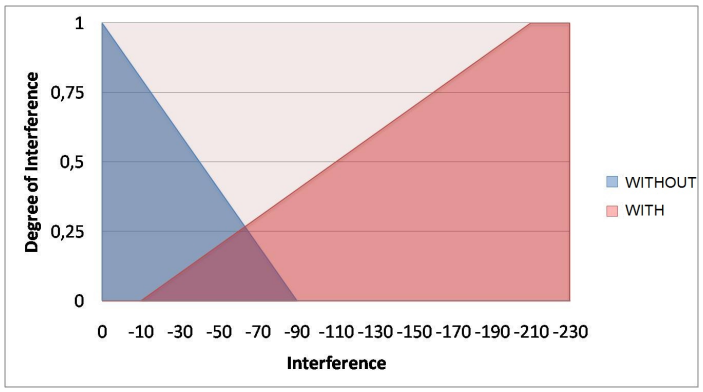


Figura 4.16: Función de Fusificación para la Interferencia

En un esquema típico basado en lógica difusa el primer paso es tomar una variable de entrada numérica y convertirla en una variable linguística (fusificación), que luego se

usa para evaluar unas reglas de control de acciones cuyas salidas se defusifican en forma de valores reales de salida. En el caso que nos ocupa, la variable de entrada es la interferencia en el canal y la variable linguística toma los valores "Con" y "Sin Interferencia" con probabilidad dada por la coordenada y en la Fig. 4.16.

La ecuación que define el grado de interferencia y en una subred *vaipho* a partir de su valor de interferencia x viene dada por la ecuación 4.4.

$$y = \begin{cases} \not\exists & si & x > 0 \\ 1 + 0.0\hat{1}x & si & x \in (0, -90) \\ 1 & si & x < -90 \end{cases} \tag{4.4}$$

Por otra parte, la ecuación que define el grado de interferencia producido por otras subredes *vaipho* con valor de interferencia x viene dada por la ecuación 4.5

$$y = \begin{cases} \not\exists & si & x > 0 \\ 0 & si & x \in (0, -10) \\ -0,005(x + 10) & si & x \in (-10, -230) \\ 1 & si & x < -210 \end{cases} \tag{4.5}$$

Este método permite tener en cuenta el grado de interferencia que encontramos en una instancia de la red *vaipho* para estimar el tiempo de espera para la reconexión. Por lo tanto, la Fig. 4.16 muestra la influencia del valor de la interferencia, que depende de la potencia de la red y de la existencia de interferencias con otras redes, sobre la probabilidad de que se considere con o sin interferencia, lo que influye en la correspondiente acción según las reglas de control que se explican más adelante.

En el diseño de las reglas de control, denotadas más abajo como Regla 1 y Regla 2, DI denota el grado de interferencia de una subred analizada, la red del nodo que ejecuta el sistema se denota como $vaipho(\alpha)$, y las otras subredes se denotan mediante $vaipho(\beta(l))$ donde $l \geq 1$. $F(x)$ se utiliza para almacenar el tiempo que un nodo de la subred $vaipho(\alpha)$ con x nodos debe esperar antes de comprobar de nuevo si existen otras instancias de *vaipho*.

En el caso de que otra instancia *vaipho* siga existiendo después del tiempo de espera, el nodo inicia el protocolo de fusión de subredes, en otro caso, el nodo no inicia ningún protocolo sino que continúa normalmente.

La Regla 2 siempre se evalúa después de la Regla 1 para determinar la cantidad de tiempo que se debe sumar o restar al tiempo de espera.

Regla 1 Propia instancia de la red *vaipho*

si ($DI(vaipho(\alpha))$ es $WITHOUT$) **entonces**

$F(x) = W(x) + \frac{DI(vaipho(\alpha))}{2}$

si no

$F(x) = W(x) - \frac{DI(vaipho(\alpha))}{2}$

Regla 2 Otra instancia de la red *vaipho*

si ($DI(vaipho(\beta(l)))$ es $WITHOUT$) **entonces**

$F(x) = F(x) - \frac{DI(vaipho(\beta(l)))}{2}$

si no

$F(x) = F(x) + \frac{DI(vaipho(\beta(l)))}{2}$

De acuerdo con la propuesta anterior, cada nodo de la subred $vaipho(\alpha)$ será capaz de estimar su propio tiempo de espera $F(x)$ teniendo en cuenta no sólo sus propios datos como el número x de nodos en su subred, sino también los datos de otras subredes vecinas $vaipho(\beta(l))$.

La Fig. 4.17 muestra el tiempo de espera total que un nodo debe esperar antes de volver a comprobar las interfaces de *vaipho*. Este tiempo depende de la cantidad de nodos que haya en su subred y de la interferencia en el canal donde $vaipho(\alpha)$ y $vaipho(\beta(l))$ están.

4.3.6. Análisis de Seguridad

Se pueden distinguir dos tipos de ataques a redes inalámbricas, los ataques producidos por nodos que no han sido autenticados, por lo que se les conoce como ataques externos, y los producidos por nodos que están autenticados en la red, por lo tanto llamados ataques internos. En este documento se ha propuesto un protocolo de autenticación fuerte y por lo tanto, se asume de antemano que dicho protocolo de autenticación impide

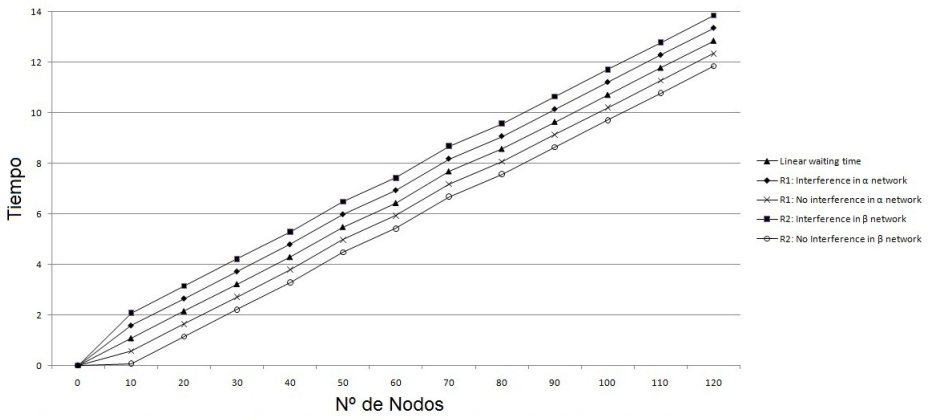


Figura 4.17: Tiempo de Espera Según Interferencia y Nº de Nodos

los ataques internos. Sin embargo, la topología de las VANETs o de las MANETs no permite garantizar la seguridad absoluta porque el canal está abierto a todo el mundo. Hay varios tipos de ataques que pueden realizar los nodos no autenticados mediante el uso de las debilidades de las redes inalámbricas distribuidas y autogestionadas.

Los ataques llevados a cabo contra el protocolo IEEE 802.11 se pueden clasificar como ataques pasivos y activos. Entre los ataques pasivos más importantes se pueden destacar el *Sniffing* y el *análisis de tráfico*. En cuanto a los ataques activos podemos nombrar *suplantación*, *reproducción*, *modificación*, y de *negación de servicio*. La mayor parte de los ataques pasivos y activos mencionados se pueden evitar con un buen protocolo de autenticación, pero hay otros posibles ataques que se pueden realizar con el esquema propuesto.

Cualquier usuario malintencionado puede realizar un ataques DoS simplemente introduciendo ruido en el canal para que, de esta manera, los dispositivos no puedan comunicarse unos con otros en esos rangos de emisión. La única solución posible a este ataque es cambiar el canal por el que se retransmite pero sin embargo, un nuevo canal tendría las mismas vulnerabilidades que el anterior.

El esquema propuesto en este documento tiene un gran inconveniente. Un usuario malintencionado podría crear una instancia *vaipho* en un canal diferente de la subred *vaipho* original y no reiniciar su interfaz. De esta manera los dispositivos de otras subredes reiniciarían sus interfaces inalámbricas y se conectarían a la subred falsa creada. En ese

momento, el usuario malintencionado podría cambiar de nuevo el canal creando otra subred *vaipho* para que los usuarios se vuelvan a conectar. Si el usuario malicioso hace esto continuamente, los usuarios estarían reconectándose todo el tiempo, lo que les impediría hacer uso de la red.

Para evitar parcialmente este ataque se debe definir un periodo de tiempo tras una reconexión durante el cual los nodos no pueden volver a reconectarse con otras subredes. Esta medida no resuelve completamente el problema, pero al menos la red funcionará normalmente hasta que algún nodo detecte y denuncie el intento de ataque DoS, puesto que los nodos tienen tiempo más que suficiente para autenticarse, generar y retransmitir los distintos eventos que ocurren en la carretera y que se almacenen en las bases de datos locales de los dispositivos hasta que caduquen.

Capítulo 5

Conclusiones y Líneas Futuras

Las redes móviles ad-hoc o MANETs representan actualmente, dada su naturaleza inalámbrica, descentralizada y móvil, una de las más prometedoras áreas en el campo de las comunicaciones. Sin embargo, para que su despliegue se realice con garantías de éxito es necesario afrontar las amenazas de seguridad que las acechan. De especial importancia es la resolución de la problemática de la autenticidad de los nodos y de las claves criptográficas, que en dichas redes es más difícil de abordar que en las tradicionales redes cableadas y redes inalámbricas con punto de acceso.

Un tipo especial de redes móviles ad-hoc son las redes ad-hoc vehiculares o VANETs, cuyas futuras aplicaciones supondrán grandes beneficios tanto en seguridad vial como en economía, medioambiente y calidad de vida de los usuarios. En la investigación de estas redes, a la problemática de la seguridad de las comunicaciones se le suma la inexistencia de VANETs reales desplegadas.

En esta Tesis se han propuesto varias soluciones a los problemas mencionados en MANETs y VANETs. En particular se han presentado nuevos esquemas de autenticación de nodos y de gestión de ciclo de vida, de claves públicas y de subredes, diseñados especialmente para dichas redes, aportando de esta forma un granito de arena en el área de la seguridad que tan necesaria es para su desarrollo. Además se ha propuesto e implementado una novedosa y práctica forma de implantar una VANET con los medios actuales.

Los principales resultados y contribuciones de la Tesis se resumen a continuación.

En el capítulo 2 se ha presentado en detalle el diseño y simulación de un nuevo y completo sistema de gestión del ciclo de vida auto-organizada para MANETs llamado SLCM. En particular se han descrito todas y cada una de las fases del nuevo esquema. El sistema SLCM es capaz de reaccionar y adaptarse a cambios en la topología de la red sin necesidad de autoridad de certificación ni infraestructura centralizada. La propuesta puede considerarse equilibrada porque los procedimientos que los miembros legítimos de la red han de llevar a cabo cuando la red se actualiza (inserción o eliminación de nodos) implica un trabajo idéntico para todos los nodos legítimos y tiene un coste inferior que otros métodos usados ya que minimiza el número de paquetes generados para el conjunto de los nodos.

Como parte del esquema SLCM se ha propuesto para el control de accesos un nuevo protocolo de autenticación de nodos en MANETs diseñado para su ejecución en entornos sin servidores y basado en el conocimiento de cada miembro de la red acerca de una pieza de información común y variable. Su técnica se apoya en las demostraciones de conocimiento nulo, paradigma criptográfico que evita la transferencia de información relevante y que permite definir un esquema de autenticación fuerte basado en el problema del circuito hamiltoniano en grafos.

El desarrollo y la evaluación del esquema SLCM se ha basado en simulaciones con el simulador de redes NS-2, que es una parte importante del trabajo realizado. Los resultados obtenidos con las simulaciones muestran la escalabilidad y la robustez de la propuesta.

También en el capítulo 2 se ha incluido una propuesta para la gestión de claves públicas en MANETs, basada en cadenas de confianza. En particular se han propuesto dos nuevos algoritmos para la actualización de repositorios de claves públicas que mejoran la eficiencia y el grado de éxito frente a la propuesta original en distintas situaciones, maximizando la probabilidad de que se pueda establecer una comunicación segura entre cualquier par de nodos, tal y como ha quedado demostrado en el análisis comparativo que se desprende de las simulaciones de los diversos esquemas realizadas en tcl y C++.

El capítulo 2 finaliza con la descripción de un nuevo esquema de autenticación mutua ligero para lectores y etiquetas que cumple con el estándar EPC Gen2 para la tecnología RFID y que es inmune contra los ataques más conocidos. El esquema propuesto puede ser usado para la gestión centralizada de MANETs mediante la adhesión de etiquetas a los

nodos móviles y la incorporación de nuevos nodos especiales constituidos por lectores, lo que permite un control inequívoco de su topología variable, alternativo al esquema SLCM descentralizado.

En el capítulo 3 se ha propuesto un método de autenticación de vehículos para VANETs autogestionadas y descentralizadas, que no requiere el despliegue de ningún tipo de infraestructura en la carretera ni de equipos especiales instalados en los vehículos. El protocolo propuesto se basa en pares de claves pública/privada y grafos certificados, y en un esquema de descubrimiento de nodos usando pseudónimos variables, que protege la privacidad y previene a los vehículos de posibles seguimientos. Nuestra propuesta permite el uso de un esquema de cifrado híbrido que combina criptografía de clave pública y criptografía de clave secreta, lo que puede ser utilizado para optimizar los recursos.

Al final del capítulo 3 se ha incluido una propuesta de uso de clústers como solución para reducir el número de comunicaciones producidas en VANETs en condiciones de tráfico denso, cuando la sobrecarga de los datos transmitidos provoca un descenso considerable en la calidad de las comunicaciones. En particular, se ha detallado una descripción completa del esquema propuesto para la gestión de clústers autónomos en VANETs, que incluye la diferenciación entre los posibles estados en los que pueden estar los nodos, desde el estado inicial en el que el nodo no pertenece a ningún clúster, a la elección de un clúster existente para unirse a él, o la creación de un nuevo clúster. También se ha definido cómo proceder con las comunicaciones entre clústers y dentro del clúster. Por otra parte, se han presentado un algoritmo de selección del líder de clúster basado en una versión del problema de conjunto independiente, y un algoritmo para el establecimiento de clave secreta compartida en el clúster basado en una generalización del protocolo Diffie-Hellman.

El análisis de las propuestas recogidas en el capítulo 3 se ha realizado a través de simulaciones con el simulador de tráfico de código abierto SUMO para definir los modelos de movilidad de los nodos y el simulador NS-2 para definir las comunicaciones. Los resultados obtenidos demuestran que nuestras propuestas mejoran el rendimiento y seguridad de las VANETs, garantizando al mismo tiempo la entrega de mensajes en tiempo real.

En el capítulo 4 se han recogido algunas de las cuestiones más destacables de una novedosa propuesta realizada en esta Tesis, consistente en una aplicación de teléfonos móvi-

les para la asistencia a la conducción, denominada VAiPho (VANET in Phones). VAiPho es una herramienta que permitiría crear VANETs auto-organizadas, principalmente para evitar atascos de tráfico, pero también para otras utilidades como la detección de plazas de aparcamiento libres, la localización del vehículo aparcado, o la publicidad geolocalizada. VAiPho no requiere el despliegue de ningún tipo de infraestructura en la carretera, lo que permite una introducción gradual de las VANETs sin ningún tipo de inversión por parte de los usuarios ni de los gobiernos. El objetivo principal en esta parte del trabajo ha sido implementar los diferentes algoritmos de seguridad propuestos en el capítulo anterior y desarrollar el software correspondiente para dispositivos tales como teléfonos móviles equipados con conexión inalámbrica y GPS en varias plataformas móviles. En consecuencia, los dispositivos con VAiPho pueden comunicarse de forma segura unos con otros mediante el envío de información de interés tras haberse autenticado mutuamente. VAiPho ha sido diseñado teniendo en cuenta la seguridad tanto de la información, que tiene que ser actualizada y fiable, como de los usuarios, protegiendo su privacidad. Por este motivo, VAiPho incluye varios algoritmos para prevenir el fraude y la transmisión de información incorrecta generada en situaciones anormales o por nodos ilegítimos. También se ha prestado especial atención a que la interfaz de VAiPho fuera sencilla para garantizar que el programa no cause distracción en el conductor. Por este motivo se utilizan mensajes de aviso por voz e iconos de los eventos en la pantalla. La forma de funcionar de VAiPho requiere que haya muchos usuarios que hagan uso de la herramienta para generar y retransmitir la información. Por esa razón es conveniente que la herramienta sea gratuita para el usuario, de forma que se ha propuesto la obtención de ingresos mediante el soporte a publicidad geolocalizada que diferentes empresas podrían contratar.

Al final del capítulo 4 se ha propuesto el desarrollo de varias soluciones prácticas para un problema descubierto tras la implementación de VAiPho con dispositivos reales, consistente en la necesidad de fusionar varias subredes inalámbricas que se forman con los dispositivos móviles equipados con Wi-Fi cuando se utiliza el protocolo IEEE 802.11b/g. El problema principal analizado se produce cuando las subredes se crean en canales diferentes de manera que nodos de diferentes subredes no son visibles unos para otros. La solución óptima para resolver este problema sería que las subredes más pequeñas fuesen absorbidas

por las subredes mayores, pero no hay ninguna posibilidad de que los nodos de cada subred sepan el número de dispositivos que conforman otras subredes. Por lo tanto, la primera solución propuesta ha sido un algoritmo simple y determinista en función del número de nodos de la subred. Además, al final del capítulo se ha esbozado una solución más compleja que completa la anterior mediante el uso de lógica difusa para estimar el tiempo que un nodo debe esperar antes de volver a reiniciar su interfaz de red, basándose tanto en el número de nodos de su subred como en los datos sobre posibles interferencias con otras subredes. VAiPho ha sido comprobado en numerosas pruebas en entornos reales y los resultados obtenidos son muy prometedores. En particular, con respecto a la fusión de subredes la propuesta determinística ha sido analizada a través de la implementación en dispositivos reales a pequeña escala para a continuación, usar los datos obtenidos en simulaciones NS-2 a gran escala, obteniéndose que en pocos segundos, todos los nodos de una subred se combinan con la otra subred.

Entre los problemas abiertos que serán afrontados en un futuro próximo destacan el estudio de aplicaciones concretas y limitaciones prácticas de los esquemas propuestos para autenticación de nodos, y para gestión de ciclo de vida, de claves públicas, y de subredes tanto en MANETs como en VANETs, así como su puesta en práctica en entornos reales a gran escala. También la implementación de la propuesta basada en lógica difusa es un trabajo en progreso. Además, en el futuro se pretende ampliar las propuestas actuales a las llamadas redes de próxima generación en la conocida como Internet de las Cosas o IoT (Internet of Things), que ofrecerá conectividad ubicua a usuarios móviles a través de redes inalámbricas heterogéneas. Dichas redes darán soporte en muchos escenarios diferentes de aplicación, incluyendo militares, ambientales, comerciales, y de emergencia, salud, vigilancia, educación, negocios, etc., porque se podrán construir rápidamente debido a que no necesitarán ningún tipo de infraestructura fija o centralizada.

SOLUTIONS FOR AUTHENTICATION AND SUBNETWORK MANAGEMENT IN MANETS AND VANETS

EXTENDED ABSTRACT

Appendix A

Introduction

A.1. Acknowledgments

This dissertation would not have been possible without the valuable guidance and help of several people who in one way or another, have contributed to its completion.

First of all, I would like to thank my supervisor, Pino Caballero, who, with her priceless and generous support, and her endless encouragement helped me to overcome all the problems that have arisen, working tirelessly and guiding me from the beginning to achieve the objective.

I also want to thank Candelaria Hernández, for her advice and kindness at all times, which has facilitated the development of this work.

Alexis Quesada also deserves my thanks for helping me in my early research.

Furthermore, thanks to Wladimir Bodrow and Otokar Grosek for the great opportunity to stay with them in Berlin and Bratislava, where their hospitality allowed me to find a supportive and friendly working environment.

I would also like to thank Miguel Soriano, who during his visit gave me great advice and suggestions that have provided more value to this doctoral thesis.

Special thanks goes also to all my friends in the department, especially to Pepe Moreno, Belén Melián and Marcos Moreno for sharing good times.

Besides, I thank my mother, family and friends because they have been there to

support and encourage me when I needed it. In particular, thanks to my sister Lydia for proofreading some parts of this thesis in English.

Last, but not least, I am grateful to my wife and travel companion, Jezabel Molina, for everything.

Without all of you, I would have never fulfilled this dream.

A.2. Preface

Over the last years, wireless networks are gaining increasing popularity due to their growing performance and their variety of applications. These networks allow users to access information and resources in real time without being physically connected. They also offer great flexibility at low cost because there is no need for wired installations, what means that they are readily deployable. This is the main reason why they are very useful in environments where the installation of fixed infrastructures is very expensive, such as military and agriculture environments, emergency situations, etc.

Mobile Ad-hoc NETworks or MANETs are a type of wireless and distributed network without central authority where the nodes are mobile. The behavior of a MANET is in many aspects similar to a Peer-to-Peer or P2P network because in both cases the nodes have to receive and send information in a decentralized manner. Management of MANETs involves many difficulties because, for instance, their topology is highly changing due to the mobility of nodes and to the absence of fixed infrastructure.

Vehicular Ad-hoc NETworks or VANETs can be considered a subset of MANETs where mobile nodes are vehicles. In their classic definition, VANETs allow not only communicating information between On Board Units or OBUs located in vehicles, but also with the Road Side Units or RSUs. The main objective of these systems is to provide a better understanding of road conditions to drivers in order to reduce accidents and to make driving more comfortable and traffic more fluid, thereby reducing the amount of CO_2 expelled by the vehicles into the atmosphere.

Ad-hoc networks are particularly vulnerable to several types of attacks, both active and passive. For example, an attacker may try to emulate a legitimate node and capture data and control packets, destroy routing tables, etc. In particular, the effects of attacks on VANETs can be very destructive as they may even cause deaths. For this reason, the primary purpose of this thesis is the proposal of new tools to protect mobile ad-hoc networks from different attacks, ensuring as far as possible that the generation of information, as well as its retransmission are performed correctly. To this end, we propose and analyze here new schemes of authentication and subnetwork management for MANETs and VANETs.

Note that simulations play a key role in this work as they allow analyzing and evaluating the performance of the proposals on a large scale and in different conditions. In particular, many of the algorithms presented in this thesis have been simulated with the NS-2 network simulator and the SUMO traffic simulator. The implementations of some of the proposals on real devices are also of great interest in this thesis because they allow not only assessing their behavior in real environments, but also discovering problems that the simulations do not detect, and obtaining real data to feed larger scale simulations. In particular, the implementations on real devices have been carried out in the Windows Mobile platform using Visual Studio 2008.

A practical result of this work, in collaboration with other research, is VAiPho (VANET in Phones), which is a tool for driving assistance. VAiPho allows creating a real vehicular network using only smartphones, without installing any infrastructure neither in vehicles or on roads. VAiPho already offers several applications in urban environments, such as the detection of traffic jams, free parking spaces and parked vehicles. This tool is the product of the implementation of a patent of the University of the Laguna as a result of the research described in this thesis.

This report is divided into four chapters. In the first one, the main concepts used in the rest of the document are introduced. The second chapter on MANETs describes new schemes for the management of life cycle, public keys, and topology of such networks. The third chapter is oriented towards VANETs and contains several new proposals for authentication and management of clusters. The fourth chapter presents some issues on the design, implementation and results of VAiPho tool, and several proposed solutions to a problem of subnetwork fusion that was detected after the implementation of VAiPho.

The present research is in progress and its next goal is to develop a full and functional implementation of VAiPho in the major mobile platforms iOS and Android, with the aim of analyzing its large-scale behavior in real environments.

A.3. Contributions of the Thesis

This thesis includes the results of several research papers that have been published in journals and conferences related to security and management of MANETs and VANETs. In most cases implementations in simulations with NS-2 and/or SUMO, as well as in real devices, have played a main role.

As a first contribution contained in this report, Chapter 2, which deals with MANETs, proposes a new system for Self-organizing Life Cycle Management called SLCM. An important part of this system is a new protocol for authentication of nodes in MANETs. Both schemes are part of a paper accepted for publication in an indexed journal [17], and another paper collected in a volume of the LNCS [33]. In addition, successive drafts of these proposals were presented in two indexed conferences [38] [39] and two unindexed conferences [21] [26].

Chapter 2 also includes a new scheme for public key management in MANETs based on certificate graphs, and two novel algorithms for updating repositories. Such algorithms are included in a LNCS volume [40] and an non-indexed conference [103], respectively.

Chapter 2 is finishes with a proposal describing an RFID-based system for the management of MANET topology where each node has a tag to locate it. The main contribution is a lightweight mutual authentication scheme between reader and tag, designed to work with the strict restrictions on passive tags. This proposal has been accepted for publication in a journal with impact factor [34]. Drafts and ideas related to various aspects of the scheme have been the subject of two publications in LNCS volumes [28] [155], and presented at a non-indexed conference [34].

Chapter 3, devoted to VANETs, describes a distributed authentication scheme for nodes, specially designed for operating without any external support such as centralized authorities or road side units. The first contribution related to this topic is the description of schemes to generate certificates between nodes and to update local repositories in order to store the minimum information needed to communicate with any other node in the network. These proposals have been presented at two indexed conferences [47] [37], and an enhanced version has been sent to a journal with impact [20].

In addition, Chapter 3 also describes an architecture for managing groups of nodes in VANETs, here called clusters, which aims to reduce the number of communications in heavy traffic conditions, where a large amount of information packages are produced in a limited area, which degrades the quality of communications. Different components of the proposed architecture have been published in two volumes of the LNCS [22] [149], and presented at two indexed conferences [23] [150], and three non-indexed conferences [27] [24] [25], the latter having received a best paper award. The overview of the architecture is under review in an indexed journal [19].

Chapter 4 presents the tool VAiPho, arised from the implementation of some schemes presented in this thesis in order to deploy the first real VANET using only mobile phones. The main contributions implemented in VAiPho are the proposed distributed schemes for node authentication and public key management, and the software design of the tool, the user interfaces for mobile phones, and various vehicular applications. We have carried out a full implementation of a beta test mode of VAiPho for Windows Mobile platform, as well as some demos in Android and Symbian. Numerous real demonstrations have been developed with mobile phones and vehicles in real environments, presented to different companies and entities. VAiPho can be considered the main contribution of this thesis, as has been reflected in the patent [35], whose license for marketing has been recently acquired by a domestic company. The media impact of VAiPho after being patented and presented a real demonstration with vehicles in front of companies, entities and media has been overwhelmingly positive, appearing on numerous news programs, digital diaries and blogs, as well as on many interviews on radio and television media and print media [206]. Several results related to VAiPho have been presented in three communications at two non-indexed conferences [31] [24] [27]. The most current version of VAiPho design has been sent to an indexed journal [30]. It is also noteworthy that the idea behind VAiPho has deserved the first prize of the entrepreneurs contest "Conocer es Valer" of the University of La Laguna [29] and a End-term Project codirected by the author of this thesis that obtained the highest score [139].

Finally, to end this memory and Chapter 4 we include two proposed solutions to a problem detected once designed and implemented the first version of VAiPho, consisting in

the existence of subnetworks needing to be merged. Specifically, we propose a deterministic solution and an improvement using fuzzy logic. The results of this last section have been submitted to a journal with impact factor, which is currently under review [18].

A.3.1. Indexed Journals and LNCS

- [17] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-Organized Life Cycle Management of MANETs. Accepted by Security and Communication Networks, 2012.
- [18] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Merging Subnetworks in VANETs by Using the IEEE 802.11xx Protocol. Enviado a Eurasip Journal of Wireless Communications and Networking, 2011.
- [19] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-Organized Clustering Architecture for Vehicular Ad-hoc Networks. Enviado a Journal on Cluster Computing, 2011.
- [20] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Zero-Knowledge Authentication in Self-Organized VANETs. Enviado a IETE Journal of Research, 2011.
- [22] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Group Formation Through Cooperating Nodes in VANETs. Cooperative Design, Visualization and Engineering, Lecture Notes in Computer Science, Vol. 6240, 105-108, 2010.
- [28] Caballero-Gil, C., Caballero-Gil, P., Peinado-Dominguez, A., Molina-Gil, J. Lightweight Authentication for RFID used in VANETs. Lecture Notes in Computer Science. 12th International Conference on Computer Aided Systems Theory EUROCAST 2011, Springer-Verlag, 2011.
- [30] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P. Design and Implementation of VAiPho, Tool for Deploying VANETs with Phones. Enviado a Computers & Electrical Engineering, 2011.
- [33] Caballero-Gil, P., Caballero-Gil, C. A Global Authentication Scheme for Mobile Ad-hoc Networks. Advances in Information and Computer Security, Lecture Notes in

Computer Science, Vol. 4752, pp. 105-120, 2007.

- [34] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. RFID Authentication Protocol Based on a Novel EPC Gen2 PRNG. Accepted by Information-An International Interdisciplinary Journal, 2012.

- [40] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Quesada-Arencibia, A. A Simulation Study of New Security Schemes in Mobile Ad-hoc Networks. Computer Aided Systems Theory EUROCAST 2007, Lecture Notes in Computer Science, Vol. 4739, 73-81, 2007.

- [45] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Data Aggregation Based on Fuzzy Logic for VANETs. Computational Intelligence for Security in Information Systems, Lecture Notes in Computer Science, Vol. 6694, 33-40, 2011.

- [47] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C., Hernández-Goya, C. Security in Commercial Applications of Vehicular Ad-Hoc Networks, Financial Cryptography and Data Security, Lecture Notes in Computer Science, Vol. 6052, 427, 2010.

- [103] Hernández-Goya, C., Caballero-Gil, P., Delgado-Mohatar, O., Molina-Gil, J., Caballero-Gil, C. Using New Tools for Certificate Repositories Generation in MANETs. Data and Applications Security XXII, Lecture Notes in Computer Science, Vol. 5094, 175-189, 2008.

- [104] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Cooperation Enforcement Schemes in Vehicular Ad-hoc Networks. Computer Aided Systems Theory EUROCAST, Lecture Notes in Computer Science Vol. 5717, 429-436, 2009.

- [106] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Extending OLSR Functionalities to PKI Management. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6928, 2011.

- [144] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Avoid Non-Cooperation in Fully Self-Organized VANETs. Enviado a IEICE Transactions on Communications, 2011.

- [145] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Prevent Misbehaviour in VANETs. En segunda ronda de Journal of Universal Computer Science, 2011.
- [146] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Probabilistic Aggregation for Data Authentication in VANETs. En tercera ronda de Transportation Research Part C, 2011.
- [149] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Enhancing Collaboration in Vehicular Networks. Cooperative Design, Visualization and Engineering, Lecture Notes in Computer Science, Vol. 6240, 77-80, 2010.
- [155] Molina-Gil, J., Caballero-Gil, P., Fuster-Sabater, A., Caballero-Gil, C. Pseudorandom Generator to Strengthen Cooperation in VANETs. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6927, 2011.

A.3.2. Indexed Conferences

- [23] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Knowledge Management Using Clusters in VANETs. Description, Simulation and Analysis. International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management IC3K-KMIS. 2010.
- [37] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Fúster-Sabater, A. On privacy and Integrity in Vehicular Ad-hoc Networks. International Conference on Wireless Networks ICWN. 2010.
- [38] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Hernández-Goya, C. Self-Organized Authentication Architecture for Mobile Ad-hoc Networks. 6th International. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. Wiopt 2008.
- [148] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Cooperative Approach to Self-Managed VANETs. International Conference on Wireless Information Networks and Systems WINSYS. 2010.

- [150] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Group Proposal to Secure Vehicular Ad-hoc Networks. International Conference on Security and Management SAM. 2010.

- [151] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. A Vision of Cooperation Tools for VANETs. IEEE International Workshop on Data Security and PrivAcy in wireless Networks DSPAN-IEEE WoWMoM. 2010.

- [152] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Enhancing Cooperation in Wireless Vehicular Networks. 8th International Workshop on Security in Information Systems WOSIS. 2011.

- [156] Molina-Gil, J., Caballero-Gil, P., Hernández-Goya, C., Caballero-Gil, C. Data Aggregation for Information Authentication in VANETs. Sixth International Conference on Information Assurance and Security IAS. 2010.

A.3.3. Other Conferences

- [21] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Solución Global para la Autenticación de Nodos en MANETs. II Simposio sobre Seguridad Informática - Congreso Español de Informática CEDI. 2007.

- [24] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Tool to Simulate Groups in Vehicular Networks Using NS-2 and Tracegraph. 5th European Conference on Circuits and Systems for Communications ECCSC. 2010.

- [25] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Using Groups to Reduce Communication Overhead in VANETs. Second International Conference on Advances in P2P Systems - AP2PS. 2010.

- [26] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organizing Life Cycle Management of Mobile Ad hoc Networks, FTRA International Symposium on Advances in Cryptography, Security and Applications for Future Computing. ACSA. 2011.

- [27] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J., Hernández-Goya, C., A. Fúster-Sabater. Gestión de Grupos en VANETs: Descripción de fases. XI Reunión Española sobre Criptología y Seguridad de la Información RECSI. 2010.

- [31] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P., Martín-Fernández, F., Yánes-García, D. Introducing Secure and Self Organized Vehicular Ad-hoc Networks. International Conference on Computer Systems and Technologies. CompSysTech. 2011.

- [36] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil. J. An EPC Gen2 Compliant Authentication Scheme Based on a New Pseudorandom Number Generator. The 2011 FTRA International Workshop on Strategic Security Management for Industrial Technology. SSMIT, 2011.

- [39] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Hernández-Goya, C. Flexible Authentication in Vehicular Ad-hoc Networks. 15th IEEE Asia-Pacific Conference Communications APCC, 576-879. 2009.

- [41] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Yánes-García, D., Martín-Fernández, F. Detecta Atascos y Aparcamiento en tu Móvil. Salón atlántico de logística y transporte. SALT2011, Las Palmas de Gran Canaria, 2011.

- [42] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J., Yánes-García, D., Martín-Fernández, F. Vaipho: Una Herramienta para la Asistencia a la Conducción. En VIII Foro de innovaciones tecnológicas para el transporte. TRANSNOVA2011, Las Palmas de Gran Canaria. 2011.

- [46] Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C., Hernández-Goya, C. Stimulating Cooperation in Self-Organized Vehicular Networks. 15th IEEE Asia-Pacific Conference on Communication APCC, 346-349. 2009.

- [105] Hernández-Goya, C., Caballero-Gil, P., Molina-Gil, J., Caballero-Gil, C. Cooperation Requirements for Packet Forwarding in Vehicular Ad-hoc NETworks (VANETs). International Conference on Computer Systems and Technologies. 2009.

- [140] Martín-Fernández, F., Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Implementación de Comunicaciones Sseguras en Plataformas Móviles para Asistencia a la Conducción. Enviado a X Congreso de Ingeniería del Transporte. 2012.

- [147] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Herramientas para la Seguridad Cooperativa en Redes Ad-hoc. II Simposio sobre Seguridad Informática - Congreso Español de Informática CEDI. 2007.

- [153] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Reputation Lists and Groups to Promote Cooperation. International Conference on Computer Systems and Technologies. CompSystech. 2011.

- [154] Molina-Gil, J.; Caballero-Gil, P.; Caballero-Gil, C., Hernández-Goya, C. Agregación de Datos para Autenticar Información en VANETs XI Reunión Española sobre Criptología y Seguridad de la Información. Vol. 6927, RECSI 2010, 2010

A.3.4. Other Contributions

- [29] Caballero-Gil, C., Molina-Gil, J. Primer Premio del Concurso de Emprendedores "Conocer es Valer". http://emprendeull.ning.com/profiles/blogs/entrega-de-premiosdel-concurso-conocer-es-valer. Universidad de La Laguna. Importe: 3.000 Euros. 2011.

- [35] Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Sistema de Comunicaciones Seguras en una Red Ad-hoc Vehicular Espontanea y Autogestionada. National Patent No. P201000865. 29 June 2010. International Patent No. PCT/ES 2011/000220. 29 June 2011. Universidad de La Laguna. Tenerife. Spain. Licencia de Comercialización Adquirida por Empresa DETECTOR, S.A., 2011

- [139] Martín-Fernández F. End-term Project Directed by Caballero-Gil P., Caballero-Gil C. Implementación de comunicaciones seguras en la plataforma Symbian para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), June 2011.

Appendix B

Mobile Ad-hoc NETworks (MANETs)

A Mobile Ad hoc NETwork (MANET) is a type of wireless network without any infrastructure, where nodes must adapt to the changing dynamic situations that result from their mobility. Because of the decentralization of nodes and the security needs of communications, management of MANETs must be self-organized, which is a major research challenge.

B.1. Self-organizing Life Cycle Management of Mobile Ad hoc Networks

In order to cope with the intrinsic properties of MANETs, a new decentralized management system for MANETs called Self-organizing Life Cycle Management (SLCM), is here fully described and evaluated. Regarding security, node authentication is the most critical component of access control in any network, and in particular in MANETs. Broadcasting is also a fundamental data dissemination mechanism in these networks. Both aspects have received special attention when defining the proposed SLCM system. In particular, both a strong access control algorithm based on the cryptographic paradigm of zero-knowledge proofs, and a three-step broadcast protocol, are here defined. This work includes the per-

formance evaluation of the scheme, and the obtained experimental results show that SLCM significantly improves both the quality and the security of life cycle management of self-organized MANETs.

B.1.1. Problem Statement

Unlike the architecture of other networks, in a MANET every node may work as a host and router at the same time, and each movement of a mobile node affects the topology of the network. Thus, routing in ad hoc networks is one of the most extensively studied problems [219, 112] In addition, the lack of central infrastructure also makes it difficult, if not impossible, the existence of an authority that manages the operation and security of the network. This paper has been prepared with the aim of solving these problems.

Resource constraints and, in particular, limitations in communication and computation, gradual deployment and need of scalability, lack of central or fixed infrastructure, and unreliability of the radio media are some of the main challenges that must be taken into account when designing any protocol for MANETs. [13] gives a useful state of the art on practical and global solutions for MANET deployment. Regarding security, four important aspects are authenticity, confidentiality, integrity and privacy. Among them, authentication can be considered the most critical one because it enables the proper identification of legitimate nodes, allowing the fulfillment of any other security service.

In this work, we focus on monitoring and configuration management in MANETs, that is to say, on the processes to control nodes and data in these networks in order to maximize their security. In particular here we describe every phase of the proposed life cycle management scheme. A basic requirement when configuring MANETs lies on the self-organizing ability of network nodes. Recently, several self-management mechanisms for MANETs have been proposed in the bibliography for different actions such as path discovery [119] or clustering [216].

Other authentication proposals [102] are based on public key cryptography, what leads to the problem of public key certification. In general, the typical approach to this issue is through the existence of a Certification Authority (CA) that guarantees the validity of all node identities. In the case of MANETs, such a role can be played by a distributed group

of nodes [134, 220]. This approach can degrade the CA availability. Another solution to the certification problem in MANETs is based on the chain of trust paradigm [53]. Its main problem is the danger that an attacker can control the signing process by compromising only a small number of nodes.

In the present work, a legitimate node presents its credentials to another legitimate node in an attempt to access the network according to an authentication process based on the established cryptographic paradigm of Zero-Knowledge Proofs (ZKPs), which were introduced in [95]. Until now a few publications have mentioned the proposal of authentication systems for MANETs using ZKPs [33], and none of them includes the authentication proposal in the context of a whole life cycle management dealing with the related problem of topology changes due to mobility, which is exactly the main objective of this work.

The rest of this section is organized as follows. Subsection B.1.2 discusses the security motivation of this research. Subsection B.1.4 presents preliminaries including general aspects and notation, and a new optimized way to perform broadcast in MANETs. Subsection B.1.5 shows a complete description of the operation of the proposed scheme called SLCM, providing specific details of every phase: network initialization, node insertion, access control, proofs of life and node deletion. The security of the proposed scheme is discussed in subsection B.1.6 while subsection B.1.7 illustrates its performance analysis.

B.1.2. Security Goals

The proposed scheme is thought for small and medium-sized MANETs where security for communications is required at the expense of sending control packets. Since MANETs do not have any fixed infrastructure, their capacity to support network routing is limited so the schema is not appropriate for large networks as it would increase the complexity and the number of control packets too much.

Efficiency, reliability and security are our main design goals for the self-organizing life cycle management that we propose in this work for MANETs. In order to describe the security objectives, we distinguish between outsider and insider nodes. An outsider node is a node that is not an authorized member of the MANET whereas an insider node is an authorized legitimate member of the network. The security goal of this research is to

develop mechanisms that protect a self-organized MANET without any central authority against malicious behavior from outsider nodes as well as from insider nodes.

Detecting attacks from insiders is one of the tasks of Intrusion Detection Systems (IDSs). Since insiders have access to the MANET, it is easy for them to launch sophisticated attacks. In this paper we propose a response system providing the capability to effectively cut off compromised insiders from the MANET. In addition, the scheme offers some level of protection against insiders who try to forge packets and impersonate other insiders.

We now describe our main security goals in defending the underlying network against outsider nodes.

Any packet transmitted by an outsider node should be immediately dropped by the receiving insider node at the first hop with a very high probability. In other words, packets sent by outsiders should not be allowed to be propagated through the MANET. By fulfilling this requirement, we can successfully guard against a myriad of attacks launched by outsiders, such as DoS (Denial of Service), wormhole attacks, man-in-the-middle, SYN flooding, etc. This is because in this way we are effectively disabling the outsider's ability to route any packet to any node that is not its neighbor. However, the aforementionated requirement dictates that every packet has to be authenticated at every hop, which in turn means that the authentication mechanism should be extremely efficient.

On the other hand, the outsider node is assumed to have the capability to spoof its identity, data such as its IP and MAC addresses, so these are not considered reliable in the schema.

The outsider is also assumed to have access to the wireless channel so it can eavesdrop on legitimate traffic. Thus, if the traffic is supposed to remain confidential, end-to-end encryption should be used to protect it, and, in any case, legitimate traffic should not be useful to launch attacks.

If an IDS is used to discover a compromised insider, the system must be able to exclude it from propagating any packet within the network. Regarding this issue, Certificate Revocation Lists (CRLs) might be used to revoke the certificates of compromised insiders. These CRLs have to be sent to the whole network when a group of legitimate nodes detect a malicious node. Then, its CRL receiving every node update. If some node does not update

its CRL because it was off-line, it can check it against the version of the CRL that is sent during the life cycle of the network. In order to check whether a node is compromised or not, nodes must verify the information sent by its neighbors. If multiple nodes receive incorrect or inconsistent information repeatedly from the same source, this group of nodes introduce the information of the suspicious node in the CRL. In this case there must be a minimum number of nodes that agree to sign the revoked node, which is a threshold that depends on the size of the MANET.

B.1.3. GRI Protocol for Optimized Broadcast

Due to the absence of fixed infrastructure, routing is a hard problem in MANETs. The proposed scheme allows to know which nodes are authenticated and on-line, without any fixed infrastructure. In this paper we do not propose new routing schemes because the simulations of our proposal show that existing schemes such as DSR or OLSR give good results without saturating communications.

The following sections describe, respectively, a new optimized protocol for broadcast in MANETs that is used in the SLCM scheme, an overview of the SLCM proposal and a description of the used notation.

The broadcast protocol called *GRI* (Go-Return-Information) is a new optimized broadcast scheme designed to solve some problems in wireless communications without centralized authority. The protocol consists of three simple phases called go, return and information.

In the first stage the node that initiates the GRI broadcast sends a signal through broadcast with a request-response to all nodes that are within the transmission range of the network and each node receiving this message, forwards it to its neighbors. In the return stage, the nodes that are farther from the node that initiated the GRI broadcast, that is to say, the nodes that do not have anyone else to send the message, start the return phase of the GRI broadcast. In this stage, nodes send their identifiers to the node that initiated the GRI broadcast. Nodes have a timer to wait for responses. So every node has to respond during that time to the emitter nodes. When the response goes through intermediate nodes, they add their identification to the response packet and forward it to the source node.

In the information phase the node that initiated the GRI broadcast gets all the information of the network and sends a broadcast again with all the information of the network to all nodes. With this simple protocol it is possible to control the entire network and to send relevant information to all nodes in the network by generating the least number of control packets in the network.

Notice that in special situations such as for example if nodes are placed in a line, the packet size might grow quite large because if the initiator is the node on one end-point of the line, the return packet will contain the information gathered from all the nodes in the network.

As a visual example of the usual situation, Figure B.1 shows a comparison between packets generated by the tool trace. For 20 nodes the number of generated packets from each node by a normal broadcast and by the GRI broadcast are compared. Note that the number of broadcast packets generated by the GRI protocol is between 40 % and 60 % of the number of packets generated by the normal broadcast. Thus, the graph shows clearly that the results are better with the GRI broadcast than with the normal broadcast.

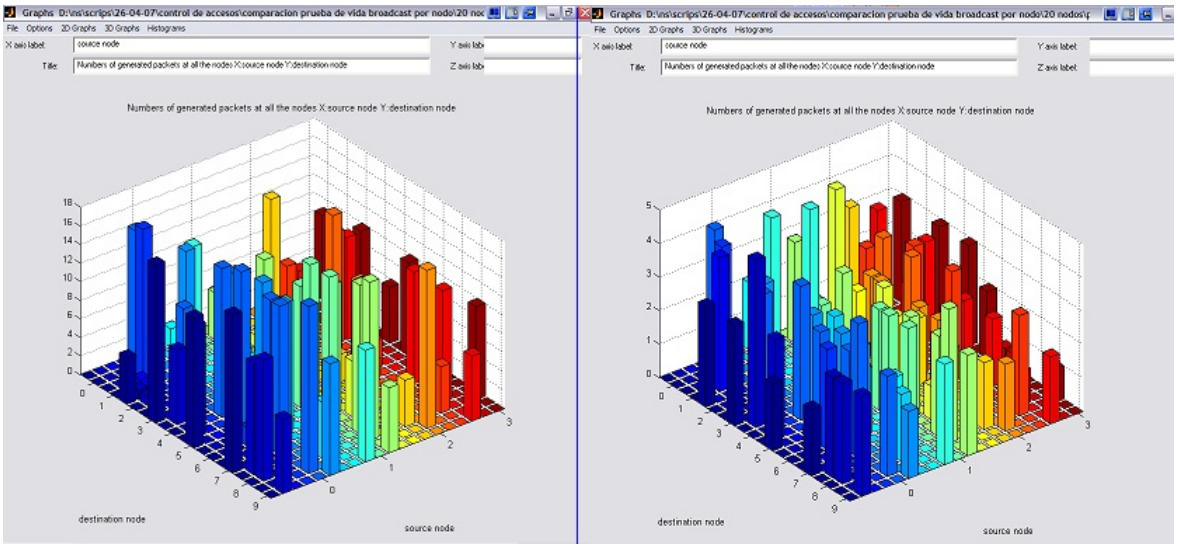


Figura B.1: General vs. Optimised Broadcast

B.1.4. Outline of the Proposal

The SLCM scheme presented here has been designed as an authentication scheme for membership in a group because when a node wants to become part of the network, it must be accepted by some legitimate nodes. The number of legitimate nodes required for the insertion of any node must be large enough to ensure that potential attackers captured by multiple nodes can not introduce new malicious nodes. This number depends on the size of the network.

According to the authors of [136], in any group management protocol it is necessary to establish robust methods to insert and delete nodes, and to allow access only to legitimate members of the group. For that reason, this work not only describes the procedure for controlling the access of legitimate nodes, but also the procedures for updating the network associated with insertions and deletions of nodes. In particular, in this paper the procedure for deciding which nodes are removed from the network is based on the time the node has been offline, so if a node has been offline for a long time (compared with a pre-arranged threshold parameter), it is removed from the network.

The cryptographic paradigm of Zero-Knowledge Proof forms the theoretical basis of the access control procedure described below. In particular, the protocol is applied for the particular case of the Hamiltonian Cycle Problem (HCP). A Hamiltonian cycle of a graph is a cycle that visits each vertex exactly once and returns to the starting vertex. The determination of whether there are cycles in a graph Hamiltonian is called the Hamiltonian Cycle Problem, which is an NP-complete problem. This problem was chosen for our design mainly because the upgrade of a solution due to an insertion or deletion of a vertex in the graph does not require a large computational effort. These operations are common in our application due to the high dynamism of the analyzed networks. However, similar schemes could be described on the basis of other NP-complete problems on graphs where the updating of a solution after the individual changes in the graph is also easy. This is the case of problems such as of Vertex Cover, Independent Set or Clique Problems, for example.

The proper performance of the proposed system is only possible thanks to the use of a chat application via the GRI broadcast scheme proposed above, since it makes possible

for some legitimate online nodes can send a message to all online network nodes. The application allows publishing all information associated with the network upgrade. Although it is not necessary that the chat messages are transmitted secretly because they are useless for illegitimate nodes, since that information is necessary to update the authentication information, it is required that only online legitimate nodes can launch the GRI broadcast of the chat application.

All the data received through GRI broadcast of the chat application for an interval of time must be stored by each online node in a FIFO queue. These data allows the update of authentication information for all legitimate offline nodes whose access is authorized by the online nodes. The duration of this period, which will be denoted T, is an essential parameter because it indicates the maximum time allowed to be out of line for any legitimate node, and also the frequency of the proofs of life that will be described below. Consequently, this parameter must be agreed by all legitimate nodes.

The network life cycle has three main phases, as shown in Figure B.2. Initialization is the first phase, where each member of the original network receives, either off-line or on-line, a piece of secret information playing the role of secret network key. The knowledge of such a secret network key can be used for access control to demonstrate eligibility of nodes in order to access protected resources or to offer some service to the network.

After the initialization phase, the legitimate nodes can participate in the network, so the node life cycle begins.

Through the access control a legitimate node that has been offline proves its membership to a legitimate online node. In order to do it, the prover node must demonstrate its knowledge of the secret network key with a challenge-response scheme.

When a legitimate node is given permission to access the online status of the network, it has full access both to protected resources such as the GRI broadcast chat application, and to provide network services such as insertion of new nodes.

The secret network key is continuously updated according to the changes in the network topology, so the secret key of a legitimate node expires if it is offline for too long. In that case, the node would have to be re-inserted in the network by a legitimate online node if it wants to join the network again.

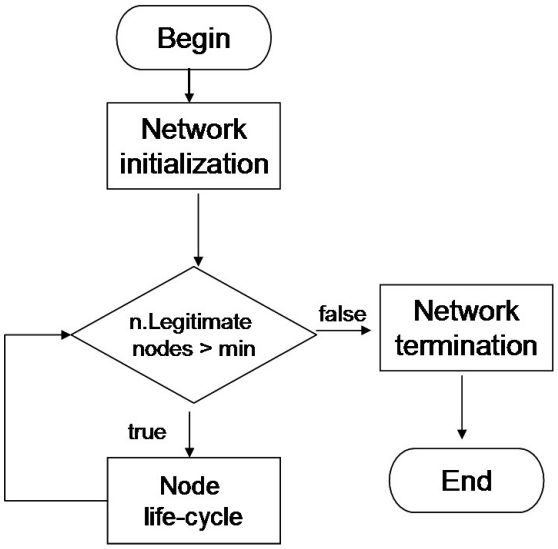


Figura B.2: Network Life-Cycle

In our proposal, the secret network key is based on the difficulty of the HCP, so the number of legitimate nodes is a very influential parameter in such a difficulty. Thus, if the number of legitimate nodes decreases and becomes too small, the network termination is automatically performed, the life cycle of the network ends.

Note that no adversary can steal any significant information, even though it accesses to all information sent through the GRI broadcast, or if it sniffs the data exchanged between a prover node and a verifier node through an access control procedure.

Notation

Notations used in the proposal are given below:

- $G_t = (V_t, E_t)$ denotes the undirected graph used at stage t of the network life-cycle.
- $v_i \in V_t$ represents both a vertex of the graph and a legitimate node of the network.
- $n = |V_t|$ is the order of G_t, which coincides with the number of legitimate nodes of the network.

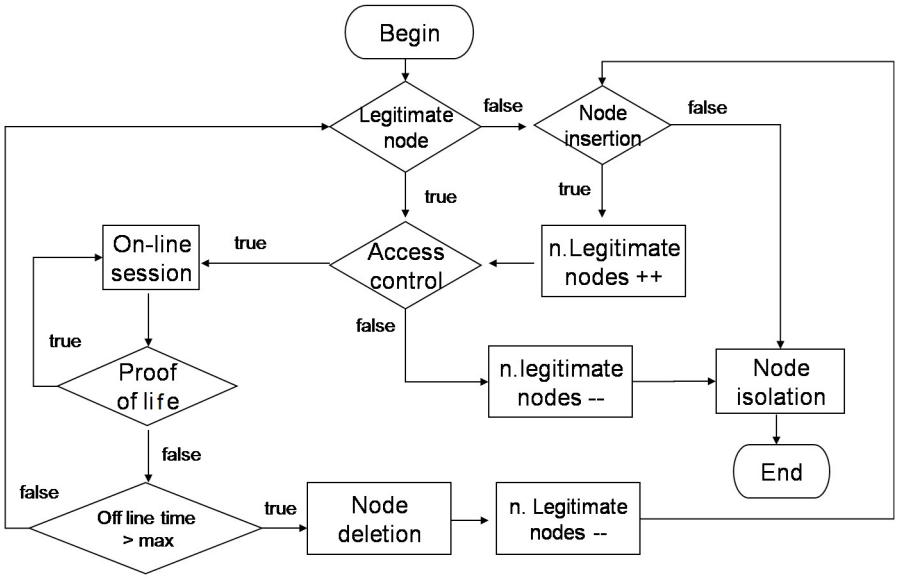


Figura B.3: Node Life-Cycle

- $m \leq |E_t|$ is a lower bound on the number of edges in the graph G_t.
- r is a large random number.
- $N_{G_t}(v_i)$ denotes the neighbours of node v_i in the graph G_t.
- $\Pi(V_t)$ represents a random permutation over the vertex set V_t
- $\Pi_i(V_t)$ denotes a random permutation over V_t chosen by v_i.
- $\Pi(G_t)$ denotes the graph isomorphic to G_t corresponding to the permutation $\Pi(V_t)$.
- $c \in_r C$ indicates that an element c is chosen at random with uniform distribution from a set C.
- HC_t designates the Hamiltonian cycle used at stage t.
- $\Pi(HC_t)$ represents the Hamiltonian cycle HC_t in the graph $\Pi(G_t)$.
- $N_{HC_t}(v_i)$ denotes the neighbours of node v_i in the Hamiltonian cycle HC_t.

- S and A stand for the supplicant and the authenticator, respectively, both during an insertion phase and during the execution of a ZKP-based access control.
- $S \rightleftharpoons A$ symbolizes when node S contacts A.
- $A \leftrightarrow S : data$ means that A and S agree on $data$
- $A \overset{s}{\rightarrow} S : information$ means that A sends $information$ to S through a secure channel.
- $A \overset{o}{\rightarrow} S : information$ means that A sends $information$ to S through an open channel.
- $A \overset{b}{\rightarrow} network : information$ represents when A broadcasts $information$ to all on-line legitimate nodes of the network.
- $A \overset{b}{\leftrightarrow} network : information$ represents a two-step procedure where A broadcasts $information$ to all on-line legitimate nodes of the network, and receives their answers.
- h stands for a public hash function.
- T denotes the threshold length of the off-line period for legitimate nodes.

B.1.5. Phases of the Scheme SLCM

In this section specific details about network initialization, node insertion, access control, and proofs of life and node deletion are given.

Initialization

The set of vertices of the graph corresponds exactly to the set of nodes of the real network during the whole life-cycle of the network. Consequently, the initialization process starts from a set V_0 of n vertices corresponding to the nodes of the initial network. Besides, each vertex sub-index may be used as ID (IDentification) for the corresponding node. The first step of the initialization consists of generating jointly and secretly a random permutation Π of such a set. The algorithm for generating the cycle HC_0 involves three basic steps. First, each node is assigned a different number $v_i \in [1, n]$ according to its IP, then it

generates a random permutation $\Pi_i(V_t)$ and shares it with the other initial nodes through a secure Bluetooth connection. Finally, every node computes the product of all permutation matrices in order to get $\Pi(V_t)$. Once this is completed, each legitimate node should know a Hamiltonian cycle HC_0 corresponding exactly to such a permutation. The partial graph formed by the edges corresponding to such a Hamiltonian cycle HC_0, is completed by adding n groups of $[2m/n]$-2 edges, producing the initial edge set E_0. Each one of these n groups of edges must have at end-vertex v_i, $i = 1, 2, ..., n$, and be randomly generated by the node v^i. The cardinality of those edge groups must be large enough so that the cardinality of the resulting edge set $|E_0| = m$ guarantees the difficulty of the HCP in G_0.

Initialization Algorithm

Input: V_0, with $|V_0| = n$

1. The n nodes of the network generate jointly, secretly and randomly the cycle $HC_0 = \Pi(V_0)$.
2. Each node $v_i \in V_0$ builds the set $N_{G_0}(v_i) = \{\{v_j \in_r V_0\} \cup N_{HC_0}(v_i)\}$ with $|N_{G_0}(v_i)| = [\frac{2m}{n}]$.
3. Each node broadcasts $v_i \xrightarrow{b} network : N_{G_0}(v_i)$
4. Each node merges $E_0 = \bigcup_{i=1,2,...,n} \{(v_i, v_j) : v_j \in N_{G_0}(i)\}$

Output: $G_0 = (V_0, E_0)$, with $|E_0| \geq m$

Insertion

The insertion phase described in this section works under the assumption of having mutual trust and a secure Bluetooth connection among the authenticator legitimate node A and the supplicant new node S. The first step that node A should do is to assign to the new node S the lowest vertex number v_i not assigned to any node in the vertex set V_t. This means either using a number previously used by some deleted node or a new number v_{n+1}. Afterwards, A should broadcast such an assignment to all on-line legitimate nodes of the network in order to prevent another simultaneous insertion with the same number, and receive their answer. If A receives less than $n/2$ answers, it stops the insertion procedure

because the number of nodes aware of the insertion is not large enough. Otherwise, A chooses the corresponding upgrade of the secret Hamiltonian cycle HC_t by selecting at random two neighbor vertices v_j and v_k in order to insert the new node v_i between them, chooses at random a set of $[2m/n] - 2$ nodes in V_t such that none of them are neighbors in HC_t, and broadcasts the set of neighbors $N_{Gt+1}(v_i)$ of S in the new graph G_{t+1} to all on-line legitimate nodes of the network.

Insertion Algorithm

Input: At stage t a supplicant node S wants to become a member of the network.

1. $S \rightleftharpoons A$ and node S convinces node A to accept its entrance to the network.
2. A assigns to S the vertex number v_i such that $i = min\{l : v_l \notin V_t\}$
3. A broadcasts $A \overset{b}{\leftrightarrow} network : v_i$
4. - If A receives less than $n/2$ answers, she stops the insertion procedure.
   - Otherwise:
     (a) A chooses at random $\{v_j \in_r V_t, v_k \in_r N_{CH_t}(v_j)\}$
     (b) A chooses at random $N_{G_{t+1}}(v_i) = \{v_j, v_k\} \cup \{w_1, w_2, ..., w_{[\frac{2m}{n}]-2} \in_r V_t$ such that $\forall w_l \notin \{v_i, v_k\}\}$
     (c) A broadcasts $A \overset{b}{\rightarrow} network : N_{G_{t+1}}(v_i)$
     (d) Each on-line node computes $V_{t+1} = V_t \cup \{v_i\}$, $E_{t+1} = E_t \cup N_{G_{t+1}}(v_i)$ and $HC_{t+1} = \{HC_t \setminus (v_j, v_k)\} \cup \{(v_j, v_i) \cup (v_i, v_k)\}$
     (e) A sends openly $A \overset{o}{\rightarrow} v_i : G_{t+1}$
     (f) A sends securely $A \overset{s}{\rightarrow} v_i : HC_{t+1}$

Output: The supplicant node S is a legitimate member of the network.

Access Control

If a legitimate node S that has been off-line from stage t wants to connect on-line to the network at stage r, it first contacts a legitimate on-line member A. Afterwards, A should check whether the off-line period of S is not greater than T. In this case, S has to be

authenticated by A through a ZKP of its knowledge of the secret solution HC_t on the graph G_t. The parameter setting of T can be based on the mean time that legitimate nodes of the network have been off-line previously. This value must be regularly updated after each successful access control through the addition of the updated mean and standard deviation plus a positive value epsilon. The initialization of T is done to a value large enough.

Access Control Algorithm

Input: At stage r a supplicant node S that has been off-line from stage t wants to connect on-line to the network.

- $S \rightleftharpoons A$
- $S \xrightarrow{o} A : G_t$, S sends openly to A the graph G_t
- A checks whether $t \leq r - T$
  - if $t \leq r - T$ then S is not authenticated
  - otherwise:
    - A and S agree $A \leftrightarrow S : l$
    - $\forall j \in \{1, 2, \ldots, l\}$
      1. S chooses $\Pi_j(V_t)$ and builds $\Pi_j(V_t)$, y $\Pi_j(G_t)$ y $\Pi_j(HC_t)$, isomorphic graph to G_t and correspondent Hamiltonian cycle, respectively
      2. S generates two large random numbers r_1 and r_2
      3. $S \xrightarrow{o} A : \{h(\Pi_j(G_t)||r_1), h(\Pi_j(HC_t)||r_2)\}$, S sends openly $S \xrightarrow{o} A : \{h(\Pi_j(G_t)||r_1), h(\Pi_j(HC_t)||r_2)\}$
      4. A sends openly the challenge $A \xrightarrow{o} S : b_j \in_r \{0, 1\}$
      5. S sends openly $S \xrightarrow{o} A$:
         (a) if $b_j = 0$ then S sends openly $S \xrightarrow{o} A : \{\Pi_j(G_t), r_1\}$
         (b) if $b_j = 1$ then S sends openly $S \xrightarrow{o} A : \{\Pi_j(G_t), \Pi_j(HC_t), r_2\}$
      6. A verifies
         (a) that the hash function h on the result of Π_j on V_t concatenated with r_1 produces the value received in step 3, if $b_j = 1$

(b) that the hash function h on $\Pi_j(HC_t)||r_2$ produces the value received in step 3, and that $\Pi_j(HC_t)$ is a valid Hamiltonian cycle in $\Pi_j(G_t)$, if $b_j = 0$

* if $\exists j \in \{1, 2, \ldots, l\}$ such that the verification is negative, then S is isolated.
* otherwise A sends securely $A \xrightarrow{s} S$: the necessary information to have full access to protected resources of the network.

Output: Node S is connected on-line to the network.

In the second step of the algorithm, a single commitment scheme based on a cryptographic hash function is used, so that after a random selection of the committed isomorphism, a hash of it and of the isomorphic HC is sent. To open the commitment, S reveals one of those pieces of information thus letting to recalculate the hash and to compare the result with the received hash value.

Proofs of Life

Every on-line legitimate node has to confirm its presence in an active way every certain interval of time of length T through the broadcast of a proof of life. During such a broadcast every node adds its own proof of life to the broadcast so that when the broadcast reaches the last node, a broadcast back starts and when the starting node receives the proofs of life of all on-line legitimate nodes, it rebroadcasts them. Since several nodes might try to broadcast their proofs of life at the same time, in order to reduce such concurrent broadcast, a random timer can be introduced so that each node defers a random time before it sends its proof of life. If it hears another proof of life during this random time, it then gives up its broadcast.

Proofs of Life Algorithm

Input: At stage t node A is an on-line legitimate node of the network.

- A initializes its $clock = 0$ just after its last proof of life
- if $clock > T$ then

1. A broadcasts $A \overset{b}{\leftrightarrow} network : A's\ proof\ of\ life$

– If A receives less than $n/2$ proofs of life as answers to its broadcast, it stops its proof of life and puts back its clock.

– Otherwise: A broadcasts $A \xrightarrow{b} network : Received\ proofs\ of\ life$

Output: At stage $t+1$ node A continues being an on-line legitimate node of the network of the network.

Node Deletion

Each node that has not proven its life is deleted from the network, and the corresponding vertex is deleted from the graph and from the Hamiltonian cycle. This way to proceed guarantees a limited growth of the graph that is used in authentication, and at the same time, allows that always legitimate nodes of the network correspond exactly to vertices in that graph.

Deletion Algorithm

Input: At stage t a node v_i is an off-line legitimate node of the network.

- A initializes her $clock = 0$
- if $clock > T$ then
  1. $\forall v_i \in V_t$: A checks v_i's proof of life in A's FIFO queue
  2. A updates $V_{t+1} = V_t \setminus \{v_i \in V_t \text{ with no proof }\}$
  3. A updates $E_{t+1} = E_t \setminus \{(v_i, v_j) : v_i \in V_t \text{ with no proof}, v_j \in N_{G_t(v_i)}\} \cup \{(v_j, v_k) : v_j, v_k \in N_{HC_t(v_i)}\}$
  4. A updates $HC_{t+1} = HC_t \setminus \{(v_j, v_i), (v_i, v_k)\} \cup (v_j, v_k) : v_i \in V_t$ with no proof, $v_j, v_k \in N_{HC_t(v_i)}$
- If A was the starter of the broadcast used for the v_i's deletion, A adds this information to the second step of the proof-of-life broadcast: $A \xrightarrow{b} network : v_i$ is deleted.

Output: At stage $t+1$ the node v_i has been deleted both from the network and from the graph.

B.1.6. Security Analysis

In the above sections, several secure algorithms have been presented so that there is no piece of information revealed by any of them that interferes with the security of the others. Thus, the resulting composite protocol is secure. In this section we discuss this issue. For the initialization of the network, there must be a minimum number of nodes to ensure the reliability of the key. Furthermore, these nodes must be legitimate and not be compromised. After initialization, the network will remain working as long as it does not fall below the threshold where the key is no longer safe.

This proposal assumes the ideal environment where all legitimate nodes are honest and where no adversary may compromise a legitimate node of the network in order to read its secret stored information. Such assumption is well suited as a basic model in order to decide under which circumstances the designed authentication scheme is applicable to MANETs. For instance, a possible adaptation of the proposal in order to avoid that hypothesis could be the consideration of a threshold scheme for every step of the scheme, so that every proof of life, insertion, access control or deletion should be done by a group or all nodes each time instead of only one node. In this way, a single dishonest node would not affect the correct operation of the network.

Another requirement of the scheme is the necessary establishment of a secure channel for both the initialization and the insertion procedures where trust between pairs of nodes is assumed. However, that aspect may be easily fulfilled thanks to the fact that most wireless devices can communicate with each other via Bluetooth wireless technology, which however is not valid for general communications because of the short distance it requires.

With respect to possible attacks, due to the lack of a centralized structure, it is natural that possible DoS attacks have the chat application as their main objective. In order to protect the scheme against this threat it must be assured that chat messages, although are publicly readable, may be only sent by legitimate on-line members of the network. Another important aspect related to the use of the chat application is the necessary synchronization of the on-line nodes, so a common network clock is necessary. This requirement has been

implemented during simulations through the chat application thanks to the broadcast GRI. The clock can be synchronized with the periodic Proofs of Life. This method is not 100 % accurate, but it has acceptable margins of error.

MANETs are in general vulnerable to different threats such as spoofing and the man-in-the-middle attack. Such attacks are difficult to prevent in environments where membership and network structure are dynamic, and the presence of central directories cannot be assumed. However, our proposal is resistant to spoofing attacks because access control is proved through a ZKP that makes useless the reading of any information published through the chat application or sent openly during an access control. On the other hand, the goal of the man-in-the-middle attack is either to change a sent message or to gain some useful information by one of the intermediate nodes. Again the use of ZKPs in our protocol implies that reading any transferred information does not reveal any useful information about the secret, so changing the message is not possible since only legitimate nodes whose access has been allowed can use the chat application.

Another active attack that might be especially dangerous in MANETs is the so-called Sybil attack. It happens when a node tries to get and use multiple identities. The most extreme case of this type of attacks is the establishment of a false centralized authority who states the identities of legitimate members. However, this specific attack is not possible against our scheme due to its distributed nature. In our scheme, the responsibility of controlling general Sybil attacks will be shared among all the on-line nodes. If an authenticator node detects that a begging node is trying to get access to the network by using an ID that is already being used on-line, such access control must be denied and the corresponding node must be isolated. The same happens when any on-line node detects that an authenticator node is trying to insert a new node to the network with a new ID, and such a node has already assigned a vertex ID. Again, such insertion must be denied and the corresponding supplicant node must be isolated. Anyway, if a Sybil attacker enters the network, any of its neighbours will detect it as soon as it sends proofs of life for different vertex IDs.

B.1.7. Performance Evaluation

We now analyze the efficiency of the proposal both from the energy consumption and from the computational complexity points of view. We consider the energy consumption, which is the result of transmissions of data and processor activities due to authentication tasks. In the proposal there are two phases when computational overhead is more significant: the ZKP-based access control and the periodic checking of stored FIFO queue. A reduction on the number of rounds of ZKP has a direct effect on the total exchanged messages size in insertions, but a trade-off should be maintained between protocols robustness and performance. Indeed, regarding total data transmission over wireless links, the ZKPs take less than 10 % in a usual situation according to the data we have estimated and obtained in simulations. For the performance analysis of the proposal we used the Network Simulator NS-2 with DSR routing protocol. We created several Tcl based NS-2 scripts in order to produce various output trace files that have been used both to do data processing and to visualize the simulation. Within our simulation we used the visualization tool of Network Animator NAM and the NS-2 trace files analyzer of Tracegraph. For the simulation of mobility we used the setdest program in order to generate movement pattern files based on the random waypoint algorithm.

The periodic proofs of life accounts for approximately 90 % of the total exchanged message size in many cases. However, we have found that these compulsory proofs of life imply an incentive technique for stimulating cooperation in authentication tasks. This is due to the fact that nodes that are broadcasters of deletions or authenticators in insertions or access controls are exempted from their obligation to broadcast their proofs of life.

In order to reduce data communication cost of the protocol, an increase on the threshold period T might be an option, but again an acceptable balance should be kept. According to our experiments, T should depend directly on the mean time that nodes are off-line and on the number of legitimate and/or on-line nodes in order to prevent a possible bandwidth overhead of large networks. The number of packets generated in the network grows linearly with the number of authenticated nodes on the network. In addition, communications are initiated periodically so although the total number of packets in the

network grows, the number of packets in a network area remains nearly constant and this number is only affected if the density in that area is increased.

The energy that a node needs is not affected by the growth of the network, but it is affected by its density. However, the size of storage that a node needs increases as the network grows. This aspect together with the routing problem are the main reasons by which it is necessary an upper limit on the size of the MANET.

Simple examples of a simulation using a few nodes consisting of scenario files that describe the movement patterns of the nodes and communication files that describe the traffic in the network were used to produce trace files that were analyzed to measure various parameters. The trace files were used to visualize the simulation using NAM, while the measurement values are used as data for plots with Tracegraph. An example of the final graph and Hamiltonian cycle associated to the example network is shown in Figure B.4 where green is used to indicate the Hamiltonian cycle, blue is used for the inserted nodes and red is used for the edges deleted from the Hamiltonian cycle when inserting new nodes.

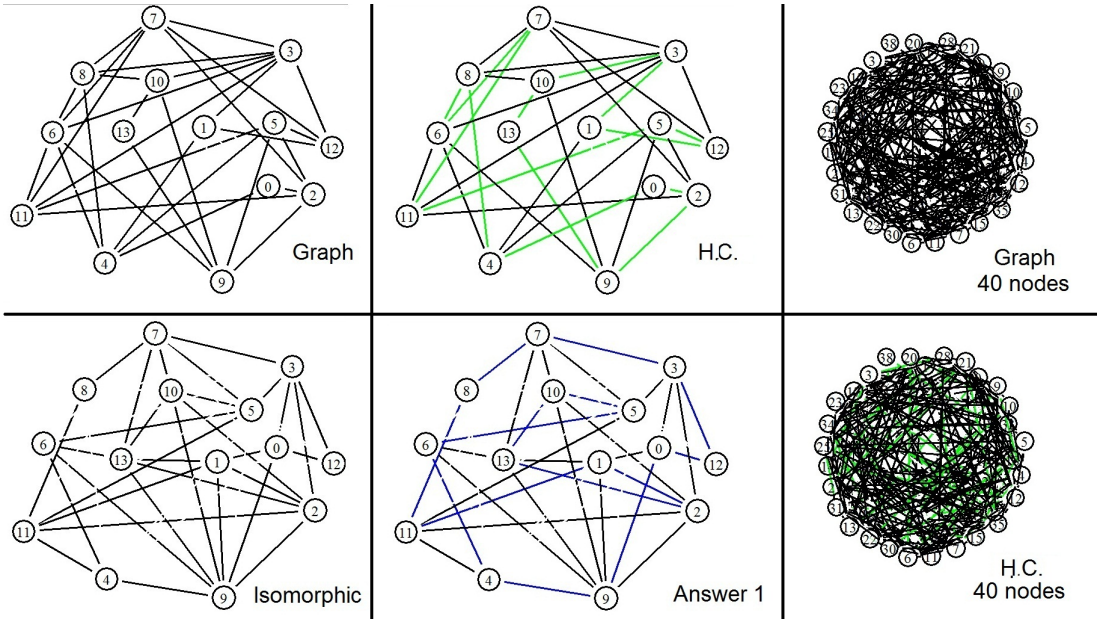


Figura B.4: ZKP Example based on HCP

As aforementioned, the HCP is NP-complete. Indeed, searching an HC by backtracking is computationally intractable and the only practical approach is through heuristic

algorithms, and even most heuristic algorithms are useless for different types of random graphs with more than 200 nodes [66].However, since the protocol does not require solving the problem but constructing a graph from a chosen solution, the difficulty of the HCP is not a disadvantage against the efficiency of the scheme but an advantage in favor of the security of the scheme.

We conducted different simulations to see the effects of different metrics by varying network density and topology. We were changing the number of nodes from 10 to 100, the area from 400 to 1000 m^2, and the period of simulation from 60 to 200 seconds. We also used the probabilities of insertions and deletions in each second from 5 % to 25 %, in order to modify the mobility rate and antenna range of nodes from 2 to 15 m/s and 100 to 250 meters respectively. This range also defines different frequencies of accesses to the network.

For the simulation we distinguished different states where nodes can be in the network, depending on different factors:

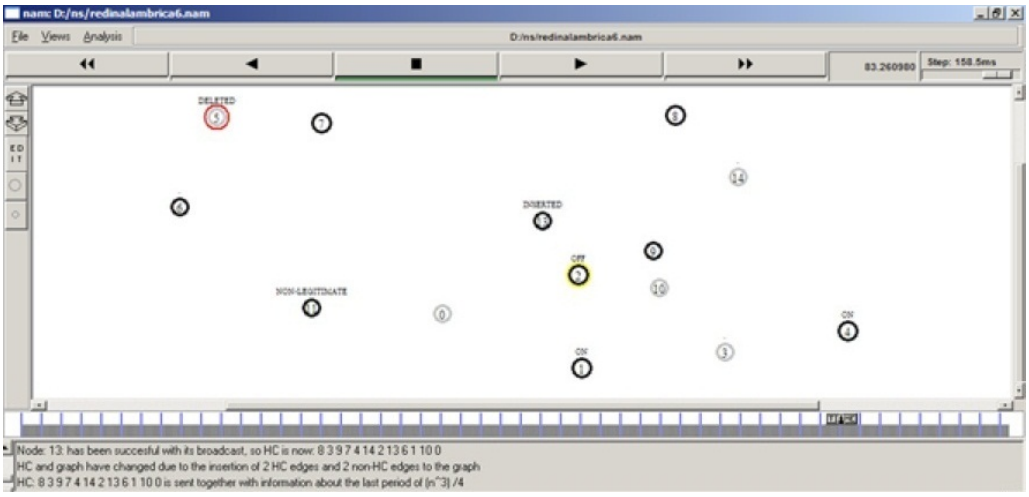


Figura B.5: Example of Network Simulation with NS-2

Time	Event	H.C.
0.1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 are legitimates	8,3,9,7,4,2, 6,5,1,10,0
1.29	Insertion of Node 14, Node 4 sends broadcast	8,3,9,7,4,14, 2,6,5,1,10,0
1.30	Nodes 3, 1, 0 do not respond to the proof of life	
11.65	Node 1 is off	
13.97	Proof of life initiated by Node 3	
14.27	Nodes 1, 2 do not respond to the proof of life	
17.27	Proof of life initiated by Node 2	
17.57	Nodes 1, 5 do not respond to the proof of life	
21.71	Node 5 is off	
31.40	Node 1 is on and Node 2 performs the ZKP	
31.46	Node 4 if off	
32.51	Proof of life initiated by Node 1	
32.78	Nodes 4, 5, 6 do not respond to the proof of life	
38.51	Proof of life initiated by Node 6	
38.79	Nodes 4, 5 do not respond to the proof of life	
41.46	Node 1 is off	
53.25	Node 1 is on and the Node 0 performs the ZKP	
59.61	Proof of life initialited by Node 6	
59.99	Nodes 4, 5 do not respond to the proof of life	
64.26	Node 5 is deleted of the network	8,3,9,7,4,14, 2,6,1,10,0
64.71	Node 2 is off	
72.58	Node 4 is on and the Node 0 performs the ZKP	
75.41	Insertion of Node 13, Node 14 send broadcast	8,3,9,7,4,14, 2,13,6,1,10,0
75.43	Node 2 do not respond to the proof of life	

Tabla B.1: Example of Trace

- *On and Authenticated*: Nodes having no label at the beginning of the simulation or nodes that are labeled *On* who have gone from *Off* to On are in this state.
- *Non-Legitimate*: There are nodes that are Off and do not belong to the network. These nodes are candidates to enter the network when they turn on.
- *On to Authenticate*: When a node is On and asks another node to be authenticated, the node is turned on but still does not belong to the network. Nodes that are turned

on but are not authenticated on the network appear to be Off to network effects.

- *Off*: This stage corresponds to the nodes that belong to the network but are off-line. These nodes either can go on and become part of the network after previously demonstration that they know the secret of the network or can be turned off until their period of life ends, in which case the node is removed from the network.

- *Deleted*: When the node is off-line for a too long time, it goes to this state where it is removed both from the HC and the graph.

- *Out of Service*: A node that is legitimate and on-line but does not respond to a proof of life started by another node because it is unreachable, would have to show that it belongs to the network when it finds another node on the network.

- *Added*: A non-legitimate node that receives enough network information from some legitimate node after an insertion procedure changes its state to *Added*.

To study the performance of the proposed scheme, simulations were performed by using the same density with different numbers of nodes and running time enough to study the effects of the proofs of life by varying only the number of nodes. From these tests we collected data on the number of connections and the number of generated, forwarded, and lost packets, which are shown in Figure B.6

This section analyzes different aspects of experimental results, which show the quality and security of the proposed scheme, considering in particular, the relationship with the number of nodes. Figure B.6 shows that according to simulations of the proposal both the number of connections and the number of generated packets increase linearly with the number of nodes. This happens when the density of the network is maintained by increasing the number of nodes. The picture also shows that the number of forwarded or lost packets also increases with the number of nodes, but in a more contained way than in the case of generated packets. This happens when the number of nodes and thereby their connections increase, but also the size of the plane increases to maintain a constant density so that the interference between nodes does not vary. Thus, the obtained results regarding lost packets can be considered positive.

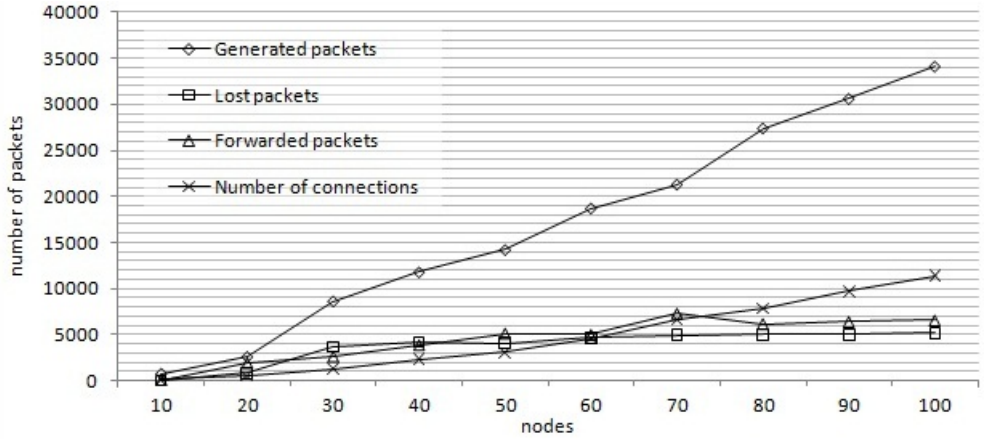


Figura B.6: Generated Packets

Figure B.7 reflects the average energy and the maximum power consumed by each node. These parameters are calculated from the processing time of packets of each node. This chart allows us to see that the maximum processing time increases with the number of nodes, although there are some exceptions. This is because with a higher density of nodes that initiate the proof of life, more computational work exists in the network. The picture shows that the average processing time is quite low and does not follow a pattern that can be used to relate it to the number of nodes. Anyway, we could conclude that on average the energy consumed by nodes does not increase too much when the network grows.

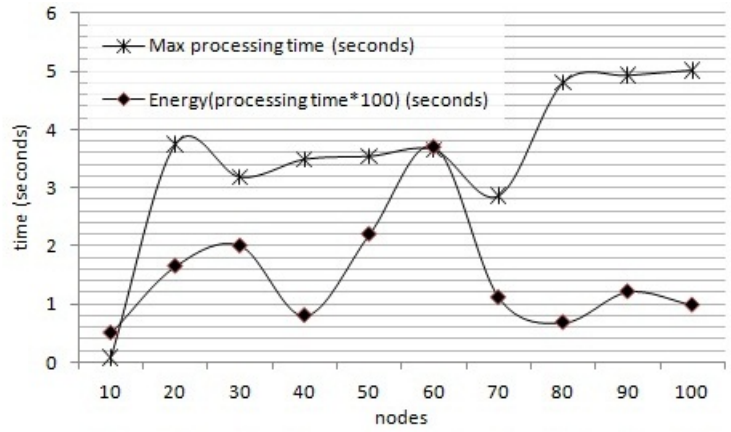


Figura B.7: Processing Time

Figure B.8 shows both the delay signal between nodes and the biggest delay that occurred in the simulation for different numbers of nodes. In both cases we see a large growth with 30 nodes and then a slight increase. The maximum delay that occurs after the 30 nodes is almost constant in 7 seconds, whereas the average time delay increases to 30 nodes and then fluctuates. These results show a good behavior of the proposal regarding delay of messages produced by the network growth.

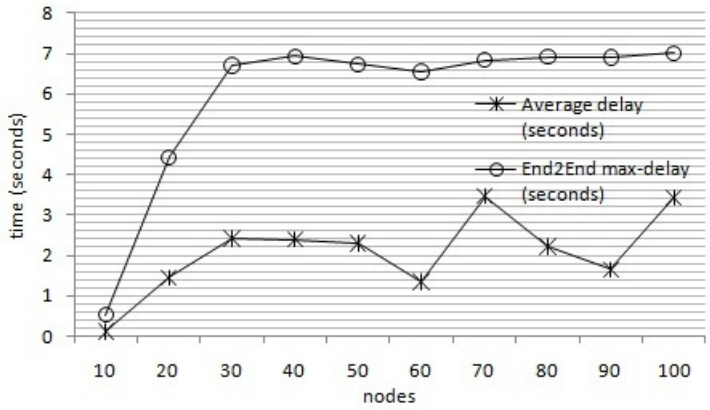


Figura B.8: Delay between Nodes

Figure B.9 shows the need for maximum storage required for each node in bits. Indeed, since the proposal does not require almost any storage, the shown growth is because each node can need to store the public keys and other data of the remaining nodes in the network. Also each node could store a number of certificates signed by and for the other nodes to authenticate them. We compared the need for storage using 1024-bit keys for RSA and 160 bits for Elliptic Curve Cryptography (ECC), which are cryptographically equivalent.

Thus, some conclusions that we can deduce from the simulations are:

- The SLCM protocol scales to any sort of networks with different levels of topology changes.

- Node density is a key factor for the mean time of insertions, but it is not as large as it might be assumed.

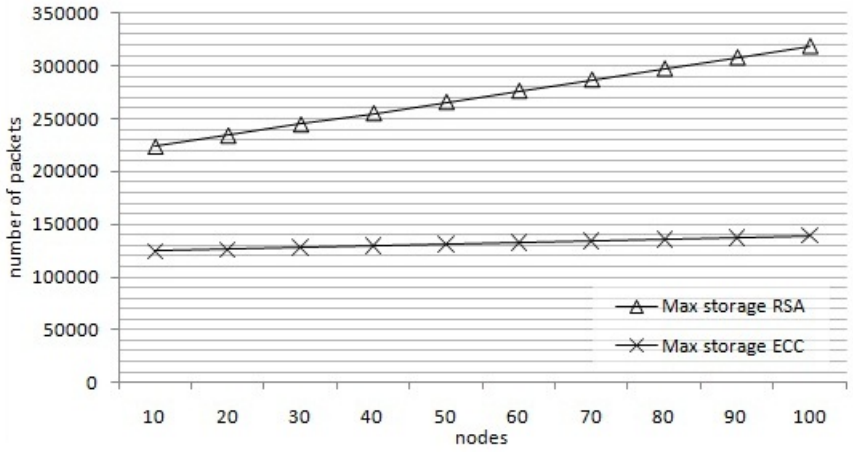


Figura B.9: Maximum Storage Requirements

- *A* right choice of parameter T should be done according to mean off-line time, number of nodes, bandwidth of wireless connections and computation and storing capacities of nodes.

- *A* positive aspect of the proposal is that the requirements in the device hardware are very low.

B.2. Public Key Management

Public key cryptography represents a cryptographic solution for the difficult problem of secret key management and allows the use of several of cryptographic applications such as the practical digital signature. However, the use of public key cryptography is impossible without solving the issue of certification of public keys.

In this chapter we discuss the problem of managing and certifying public keys in MANETs. Besides analyzing the problem and some solutions found in the literature, we propose improved schemes for certification of self-organized public keys that enable the practical use of public key cryptography in MANETs. This entire study is completed with NS-2 simulations from which many conclusions are drawn.

B.2.1. Certificate Graphs

The proposed scheme for the certification of public keys in MANETs is based on the approach described in [54] which replaces the usual centralized certification authority in the wired and wireless networks with infrastructure, by a self-organized scenario in which the certification is done through chains of certificates issued and signed by the network nodes. This scheme uses the information stored by each node and the fact that each node relies on its neighboring nodes. This last feature is the essence of a self-organized infrastructure and can be used, and in fact is used to generate trust from the trust between neighboring nodes.

In the scheme based on certificates graphs each node has a public key, a private key, and a repository including a list of all certificates of all nodes who are trusted nodes (out-bound) and the list of certificates of all nodes that rely in it (in-bound). Thus, each certificate is always stored twice, by the signing node and by its owner.

When a node wants to verify the validity of the public key of another node, it must find a certificate chain from it to the other node in the graph that results from combining its repository with the repository of the other node.

Public keys and certificates are represented as a directed graph $G = (V, E)$, known as certificate graph where each vertex u represents both a public key and its owner, and each edge (u, w) symbolizes a public key certificate w signed with the private key corresponding to u. A certificate chain is a directed path in this graph.

Note that the speed of creating a certificate graph that contains enough connections to allow communication between two nodes of a network depends on the motivation of users to distribute certificates and their mobility because as nodes share their repositories with nearby nodes, the more nodes move, the more exchange their repositories with other nodes.

The public key authentication of a node v by another node u is done by looking at the union of their repositories for a chain of unexpired and correct certificates between u and v in the graph resulting from the union of both repositories because:

1. The first certificate in the chain can be checked directly by u as it was signed by him.

2. Each one of the other certificates in the chain can be checked by using the public key

of the previous certificate in the chain.

3. The last certificate is the target user's public key v.

The choice of the certificates that each node keeps in its repository must be done carefully to satisfy two requirements: storage limitation of the nodes, and usefulness of the repository to find certificate chains for the largest possible number of nodes.

The simplest algorithm that has been proposed for the construction of the repository of each node is known as the maximum degree algorithm, named after the criteria followed for the election of the certificates which is the degree of the vertices of the certificate graph [54].

There is another more sophisticated algorithm proposed by the authors, called Shortcut Hunter Algorithm, where the chosen certificate, when they are deleted from the graph, increase the length of the shortest path between nodes connected with those certificates by more than two units. The analysis of both algorithms shows that anyone can find in the resulting graph after a sufficient number of iterations at least one chain of certificates between any pair of nodes with high probability.

In our proposals described below using a subgraph G_u of the certificate graph G containing the repository with the verified certificates by the node u, and an independent graph denoted as ${G_u}^r$ containing the certificates collected by u. Before a certificate expires, the issuer should distribute an updated version, but it may not do so. In that case, the graph ${G_u}^r$ (called compiled repository) is very useful because it provides a good estimate of the certificate graph that is not included in G_u (called verified repository).

Next, the four steps required for the management of public key repositories are briefly described:

1. Key Initialization

   Creating a pair of public and private keys by each user.

2. Initialize the Certificate Graph

   By issuing certificates to trusted nodes, the inicial certificate graph that no one really knows is defined.

There may be many reasons why u believes that K_v belongs to v, such as for example, they have exchanged their keys through a secure channel. The dynamic nature of MANETs makes users get more information on other, they distribute more certificates and assess better their confidence in the distributed certificates.

The distribution and revocation of public keys are the only conscious and intentional operations that nodes perform, since the remaining, including validation and exchange of certificates, are performed automatically.

3. Update of Compiled Repository

   The certificate exchange with neighboring nodes to create the compiled repository is a low-cost procedure that allows nodes to share their repositories. It is described as follows:

   (a) Each node u transmits a hash of the stored certificates in their repositories G_u and ${G_u}^r$ to its neighbors. The neighbors that receive this message respond in turn with the hashes of the certificates of their repositories.

   (b) Each node compares the received with that it has and asks its neighbors only about the certificates that it does not have.

   (c) If the node's local memory is too small, it deletes expired certificates from the collected repository, sorted by date.

   (d) Thus, nodes accumulate certificates in their collected repository ${G_u}^r$ such that after a short period of time, these repositories contain most of the certificate graph G. After this, nodes only need to exchange new generated certificates.

4. Creation of the Verified Repository

   The verified repository G_u of node u is updated by applying over its compiled repository ${G_u}^r$ an algorithm to choose the most appropriate certificates to include in G_u. After the execution of the algorithm, u must check by contacting the issuers, the validity of each certificate of ${G_u}^r$ to add to G_u.

B.2.2. Maximum Degree Algorithm with Two Chains

Here we propose an algorithm for the selection of certificates of ${G_u}^r$ to add to G_u for the construction of the checked repository. The objective is to store the minimum information necessary to allow nodes check if other nodes are reliable and thus, they can communicate with most of them. To achieve this, in the specific version described in this section, nodes store exactly two chains containing nodes with higher levels among those who may choose at any time according to a set of constraints. Also in our scheme we simplify the model by considering two-way communication and trust so that the corresponding certificate graph is undirected.

Each node u will have in its upgraded repository G_u all the nodes that are relying at distance 1 on the certificate graph.

After the stages 1 and 2 of initialization, nodes start their timer to update the compiled repository to a random number less than x units of time where the value of x depends on the size of the network. In the simulation we use $x = 3$. At this moment, the node runs the stage 3 upgrading its compiled repository contacting with all nodes within its range at that time.

During stage 4, construction of the checked repository, the algorithm described below is used, where chain represents each chain in the repository of the executing node and B.chain is each chain of the authenticated node.

Algorithm Maximum Degree with 2 Chains

```
00: ...//Authentication with nodes where they exchange repositories
01: //Update degrees of nodes in the repositories.
02: for (i = 0; i < size(chain); i++)
03:    for (j = 0; j < size(B.chain); j++)
04:       if (chain(i) == B.chain(j))
05:          if (degree(next(chain, i)) < degree(next(B.chain, j))
06:             updateChain(i, B.chain, j);
07:          end if
08:       end if
09:    end for
```

10: **end for**

The verified repository G_u is initialized by u first including certificates issued by nodes that directly signed its public key, which are the owner of certificates signed by u. Then, node u exchanges with these nodes its verified repository to determine its initial collected repository ${G_u}^r$ by merging them.

In each authentication, nodes attempt to update their verified repositories from their collected repositories to try ensuring confidence with the maximum number of nodes in the network by storing the minimum number of keys in their checked repositories.

After selecting the certificates to include in the checked repositories, first of all nodes must verify the trust in these certificates.

In the proposed algorithm two chains are used to update G_u from ${G_u}^r$

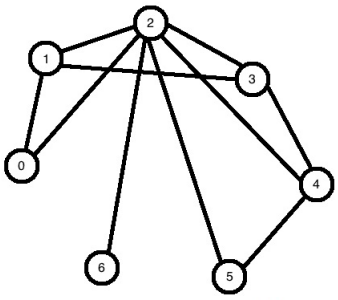
- **Chain1** $C1_u$**:** This list stores a chain that begins in one of the nodes with largest degree in ${G_u}^r$, and does not repeat vertices.

- **Chain2** $C2_u$**:** This list stores a chain without repeating vertices, starting with one node with the second highest degree in G_u or, if only there is only one, with that one.

In Fig. B.10 and table B.2 we can see an example of how the repository G_u would be and the possible chains $C1_u$ and $C2_u$ depending on with whom it exchanges repositories at any time.

In communication from the collected repository ${G_u}^r$, if a node u relies on the node v, during the merging of repositories checks whether the node v is in one of its chains $C1_u$ or $C2_u$. If so, it checks whether if it has v in any of its repositories the first element in a chain, otherwise, it adds it to its corresponding chain, and applies the same procedure recursively.

If the node v does not correspond to any vertex node of any of the two chains of u, it checks whether the first node of one of the two chains of the node v is in one of the chains of u. If so, it adds the node v to the corresponding chain of u.



Adjacency Matrix

	0	1	2	3	4	5	6
0	-	1	1	0	0	0	0
1		-	1	1	0	0	0
2			-	1	1	1	1
3				-	1	0	0
4					-	1	0
5						-	0
6							-

Vector of Degrees

Node	0	1	2	3	4	5	6
Degree	2	3	6	3	3	2	1

Figura B.10: Vector and Repository

	C1	C2
0	0 2 1 3 4 5	0 1 3 4 5 2 6
	0 2 3 1 0	
	0 2 3 4 5	
	0 2 4 3 1	
1	1 2 3 4 5	1 3 4 5 2 6
	1 2 4 3	
2	2 1 3 4 5	2 1 3 4 5
	2 3 1 0	2 3 1 0
	2 4 3 1 0	2 3 4 5
		2 4 3 1 0
3	3 2 1 0	3 1 2 4 5
	3 2 4 5	3 4 5 2 1 0
4	4 2 1 3	4 3 1 0 2 6
	4 2 3 1 0	
5	5 2 1 3 4	5 4 3 1 2 6
	5 2 3 1 0	
	5 2 3 4	
	5 2 4 3 1 0	
6	6 2 1 3 4 5	6 2 1 3 4 5
	6 2 3 1 0	6 2 3 1 0
	6 2 3 4 5	6 2 3 4 5
	6 2 4 3 1 0	6 2 4 3 1 0

Tabla B.2: Possible Chains to Store

B.2.3. Simulation of the Maximum Degree Algorithm with Two Chains

For the implementation of the NS-2 simulation of the highest degree with two chains algorithm, the code is organized in the following files:

- *Redinalambrica7.tcl:* Main file thqt is responsible for capturing the movement files and life of network generated to create the simulation of the network and its trace file.

- *Gengestionclaves.tcl:* File in which it is developed network life: insertions, nodes off or on, connections, updates of repositories, etc. This file creates other files with information about the created network that the graph generator used to create the certificate graph associated with that network. These files are:

- *grafogestionclaves.txt:* Stores the number of elements in the first line and then for each node keeps a line with the verified repository initial, a line with its chain C1 and another with the chain C2. It allows to create the certificate graph of the network and to paint the edges that are in the repositories in successive network connections.

- *conexionesgestionclaves.txt:* Saves the network connections with, containing one line per connection: connection time, first and second connection node. It is used to know when two nodes are connected. Its edges are painted them.

- *noconexionesgestionclaves.txt:* Collects failed login attempts with the same file format of connections. It is used to know when two nodes try to connect and can not. It paints the edges to make it clear that they have no node in common.

- *nuevosnodosred.txt:* Stores the input nodes in the network. At each node it keeps a line inserted with the format: timing of insertion, inserted node, list of verified repository of that node. It serves to not let them painting the edges of nodes that have not yet been inserted into the network. Once it exceeds the specified time of insertion, the edges of this node can be painted.

- *nuevosenrepositorio.txt:* Gathers the edges that nodes have inserted during the lifetime of the network. The used format in this file is:

- first line: time of insertions
- second line: inserted nodes in the repositories
- third line: source nodes of edges
- fourth line: destination nodes of edges

Used to prevent that edges that have not yet been integrated into the repositories of nodes are painted. Once the node adds the edge, this may be painted.

- *Gengrafogestionclaves.tcl:* File that captures what has happened in the network (generated by gengestionclaves.tcl). It shows the certificate graph of the network at each instant and colors the connections made between the different nodes and new edges introduced in the repositories of the nodes or by inserting a new node in the network.

In Tables B.3 and B.11 shown below, we can see a trace example and corresponding changes in the certificate graph and network connections.

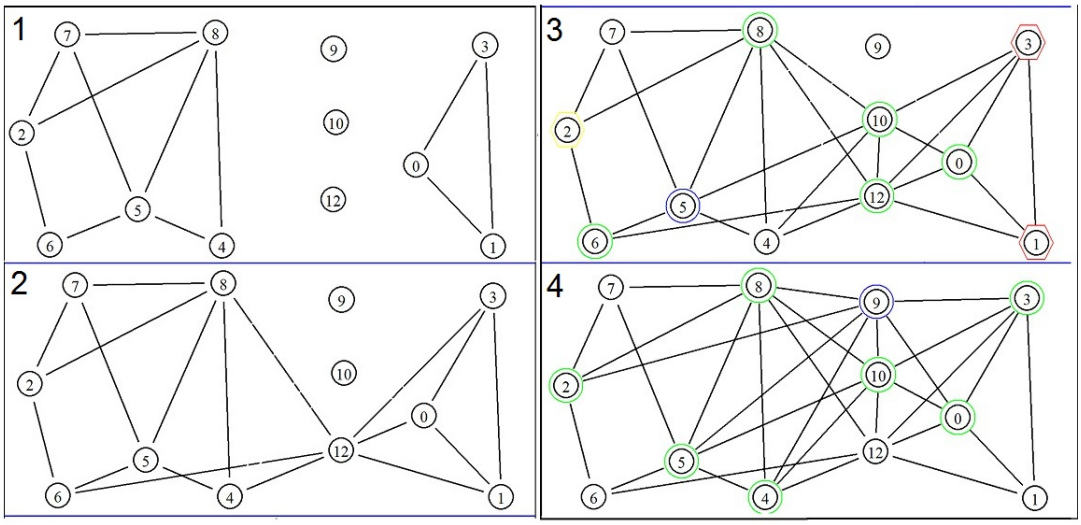


Figura B.11: Transformations of Certificate Graph

B.2.4. Maximum Degree Algorithm by Sectors

In the method of maximum degree by sectors described below the subgraph repository of a node will grow into all parts of the certificate graph of the network.

Moment	Event
0.1	Nodes 0,1,2,3,4,5,6,7,8 are On
0.1	Formed two Subnets, 0,1,3 and 2,4,5,6,7,8
1.1	GRI Broadcast to the network knowledge
2.85	3 updates repositories with 1 and 0. It does not add to the repository. Period increases
2.95	1 updates repositories with 0 and 3. It does not add to the repository. Period increases
3.14	Node 7 is Off
3.27	0 updates repositories with 1, 3, 8. It does not add to the repository. Period increases
5.26	4 updates repositories with 0, 5, 8. It does not trust in 0. Add edge 5-7. Period decreases
5.28	8 updates repositories with 0, 4, 5. It does not add to the repository. Period increases
5.31	2 no nearby nodes to update repositories. It does not add to the repository. Period increases
5.36	5 updates repositories with 4, 6, 8. It does not add to the repository. Period increases
5.85	6 updates repositories with 5. Add edge 5-8. Period decreases
7.85	3 updates repositories with 1. It does not add to the repository. Period increases
7.95	1 updates repositories with 3. It does not add to the repository. Period increases
8.26	4 updates repositories with 8. It does not add to the repository. Period increases
8.27	0 updates repositories with 8. It does not trust in 8. It does not add to the repository. Period increases
8.85	6 updates repositories with 5. It does not add to the repository. Period increases
10.28	8 updates repositories with 0, 4. It does not trust in 0. It does not add to the repository. Period increases
10.31	2 updates repositories with 5. It does not add to the repository. Period increases
11.78	Connection between 5 and 4. Confidence is checked. Data exchange begins.
	...
16.51	12 New node in the network, change signatures with 0,1,3,4,6,8.
	GRI Broadcast for the information of new network.
	...
43.90	10 New node in the network, change signatures with 0,3,4,5,8,12.
	GRI Broadcast for the information of new network.
	...
81.87	9 New node in the network, change signatures with 0,2,3,4,5,8,10.
	GRI Broadcast for the information of new network.
	...

Tabla B.3: Example of Trace in Key Management

The idea behind this method emerged after analyzing the algorithm of maximum degree with 2 chains because we detected a problem in how the graph expands their chains in the repositories of the nodes in certain cases of long networks. For example, in Fig. B.12 we can see how the updating of certain repositories makes that the two chains tend to spread to the nodes with higher degree, they grow to the same side of the graph of the figure, and therefore they do not achieve nodes of the other side. In this example, it uses two rows to reflect C1 and C2.

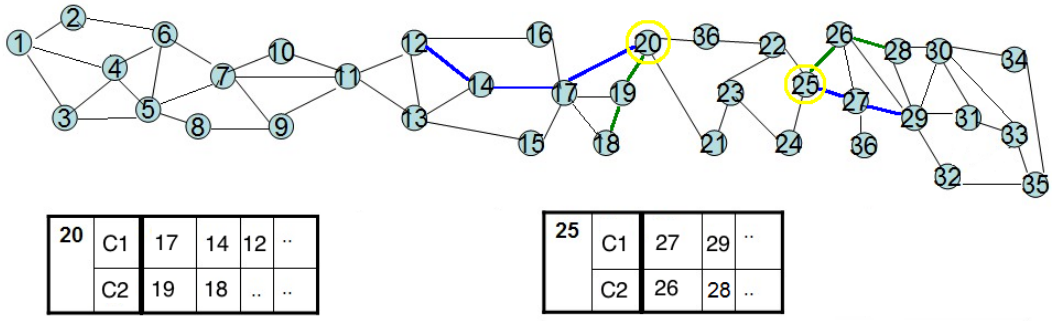


Figura B.12: Failure in Maximum Degree Algorithm with Two Chains

It can be seen that the repository of node 25, when trying to expand into the neighboring nodes of higher degree it tends to grow toward the subgraph that is on its right, ignoring the subgraph on the left. The same happens to the other side for node 20, the repository expands into the left subgraph, ignoring in this case the right subgraph. These two nodes despite being at a distance of two hops can not communicate with each other because they can not find a common known node.

Initialization is the same as in the method of maximum degree algorithm with two chains.For the update of the verified repository, the following algorithm is executed, where store represents the verified repository of the executing node, and B.store is that of the authenticated node.

Algorithm Maximum Degree by Sectors

00: ...//Authentication with Nodes where they exchange repositories

01: //Update degrees of the nodes in the repositories.

02: **for** $(i = 0; i < size(store); i++)$

```
03:    for (j = 0; j < size(B.store); j++)
04:      if (store(i) == B.store(j))
05:        if (((size(store) < limStore)&&(notHave(next(B.store(j))))))
06:          add(next(B.store(j)));
07:        if not, if ((size(store) == limStore)&&(degree(B.store(j)) > minDegree))
08:          deleteNodeMinDegree();
09:          add(B.store(j));
10:          updateNodeMinDegree();
11:        end if
12:      end if
13:    end for
14: end for
```

Data structures used to store the nodes in this method vary over the previous method. For this method, the used repositories are denoted by Distancia1 and Origin-Destination.

- *Distance1:* This repository stores the same information as in the maximun degree algorithm with two chains.
- *Origin-Destination:* This repository saves the edges that each node adds when it upgrades its repository through the exchange of information with other nodes.

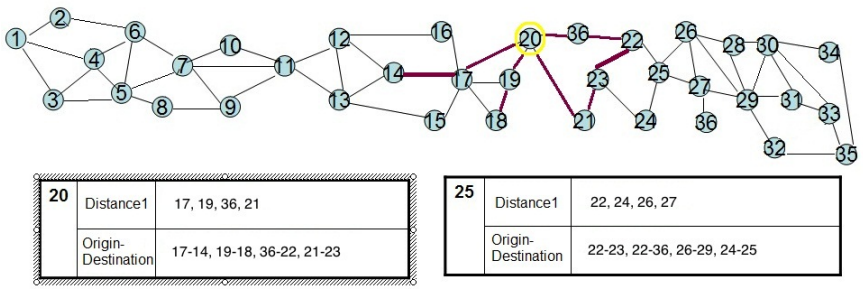
As in the previous algorithm, before exchanging their repositories nodes should check the trust with the nodes it connects. First, the Origin-Destination repository contains the edges corresponding to the vertices of highest degree of all neighbors of the node.

To update the repositories, nodes must rely on the nodes with which they update its repository. For this, node B sends to node A the two nodes in its repository (distance1 and origin-destination) with the highest degree. If the third successive node has the same degree as the second best node, it also sends it. Node A checks whether it has the received node with highest degree, and if it does not have it, it adds it to its origin-destination repository. Then, it checks the second one, and so on till it runs out of nodes to be tested.

If it finds at least one new source-destination edge to add, the time to update repositories decreases in a unit to a minimum of two units. If there is no edge to be added by any of the nodes at distance 1 at the time of update, the time for the next update of repositories is increased in one unit for that node.

With the maximun degree algorithm with two chains we can observe that it does not have limited the growth in x number of chains, so they can grow to all parts of the network and always get the best chains. On the other side, with this method the number of nodes stored in repositories will be greater than with the maximum degree algorithm with two chains.

In Fig. B.13 we can check the same example of Fig. B.12, but applying the highest degree algorithm by sectors and see that there is not the growth problem that the maximum degree algorithm with two chains has.



20		
	Distance1	17, 19, 36, 21
	Origin-Destination	17-14, 19-18, 36-22, 21-23

25		
	Distance1	22, 24, 26, 27
	Origin-Destination	22-23, 22-36, 26-29, 24-25

Figura B.13: Example of Maximum Degree Algorithm by Sectors

B.2.5. Simulation of the Maximum Degree Algorithm by Sectors

The simulation of the key management scheme by using the maximum degree algorithm by sectors is similar to the simulation performed for the maximun degree algorithm. However, there are some notable differences with respect to the updating and use of repositories that each node stores.

Initialization of the Network

In this case it is necessary to introduce the higher degree nodes into two structures C1 and C2. It simply performs the key exchanges with neighboring nodes and initializes repositories distance1 of each node. Then, nodes exchange these repositories distance1 to initialize the repository origin-destination.

In the implementation of origin-destination repository, we can see two lists for each node, one containing the source node and the other one containing the corresponding destination node in the same position of the source list.

Insert Nodes

In the process of insertion it is not necessary to save the nodes with higher degrees in C1 and C2, since in this algorithm the nodes are all stored in the same repository origin-destination.

Shutdown Nodes

The shutdown of a node, as in the maximun degree algorithm with 2 chains, does not affect anything on the network or on the repositories stored by each node.

Update Repositories

The repository update time does not vary with respect to the previous algorithm. It is performed before and after the time intervals indicated in each insert on and shutdown of nodes in the network in the main program loop.

The Check-update-repository process starts as in the previous algorithm but it varies the time to update the repositories.

It initially checks for nodes that are upgrading their repositories. For each of these nodes, after marking them with a blue circle and indicate that its update of repository begins, the nodes closes to it are calculated. With those nodes a communication is established where they exchange repositories and checks whether they can communicate by calling the function “Check_Trust.”. If possible, it introduces the time and nodes that are connected

in the file conexionesgestionclaves.txt. Otherwise, they are introduced in noconexionesgestionclaves.txt.

All nodes which are connected have a green circle around it. Then, all nodes in the repository of B are taken and sorted by degree. Node A introduces the node with the greatest degree of the node B's repository if it has not them.

Next, the node with the second highest grade is inserted. If it is has not that element, it still looks for nodes having the same degree as the second node with biggest degree.

Finally, a message indicating whether the node has added some edge or not is writting and therefore, the period increases or decreases. The following list list_ordenada_final is recalculated to introduce the element.

B.2.6. Comparative Study

After performing 25 simulations for 15, 20, 30 and 60 nodes and getting the mean results for the simulations of different types of networks we found that the maximum degree algorithm by sectors has in general the same or better performance than the maximum degree algorithm with two chains, but the results are quite similar. Below the results are shown.

15 nodes	Maximum Degree 2 Chains	Maximum Degree by Sectors
Total connections	876,0	966,9
Successful connections	812,24	909,7
Failed connections	63,8	57,15
Added edges	27,55	102,28
Not added, full repository	13,4	0
Updates of repositories	576,3	628,7
20 nodes	Maximum Degree 2 Chains	Maximum Degree by Sectors
Total connections	1069,8	1011,52
Successful connections	1037,9	985,4
Failed connections	31,9	26,12
Added edges	36,16	56,4
Not added, full repository	2,3	0
Updates of repositories	435,4	420,2
30 nodes	Maximum Degree 2 Chains	Maximum Degree by Sectors
Total connections	2885,8	2764,7
Successful connections	2870,3	2749,8
Failed connections	15,52	14,92
Added edges	75,07	80,51
Not added, full repository	2,6	0
Updates of repositories	714,39	690,24
60 nodes	Maximum Degree 2 Chains	Maximum Degree by Sectors
Total connections	5291,9	5309,0
Successful connections	5183,	5216,5
Failed connections	108,9	92,5
Added edges	142,8	191,9
No added full repository	1,0	0
Updates of repositories	1455,52	1475,9

B.2.7. Certificate Revocation

With the time, keys lose some of their safety because a malicious node has more time to test different combinations by brute force, and has more information through communications using that key. For this reason, it is advisable to change passwords from time to time to prevent that a malicious node can discover and break all communications encrypted with it.

For this reason it is necessary to define the certificate revocation process, which may occur in two ways: *implicit* or *explicit*.

The implicit revocation is a revocation of the certificate for completion of its

expiration date. After that date the certificate becomes invalid for all purposes and therefore the signatures of other certificates signed by using the corresponding private key also become invalid.

The explicit revocation happens when a node believes that its certificate may have been violated by some fraudulent node. Then the node sends an explicit revocation of the certificate for the network removed nodes or replaces it by a new certificate.

Both forms of revocation make that certificate chains that are held in local repositories of the nodes are removed from any signed certificate with a private key whose certificate has been revoked or nothing happens with these chains if the corresponding signature is replaced by another signature with the new certificate from the same node.

If a chain is removed because of certificate revocation, a node must attempt to recover lost certificates or find new substitutes chains. In each broadcast GRI should publish the list of revoked certificates explicitly.

To revoke a certificate implicitly required or that there is an authority to keep a watch on the network, or that you have a *distributed network clock*. The aim of this work MANETs there is no central authority, but this is not a problem because there is used a *distributed network clock* . This clock starts at the beginning of the network and maintained individually on each node. In the tests of life of the nodes *network clock* is synchronized to the entire network.

The explicit revocation is performed by using the broadcast GRI where the node that wants to revoke or renew its certificate can inform the rest of the network or at least, more than half of the nodes in the network at this time. Then, nodes that have not updated their repositories update it when they are in contact with nodes that have done so.

B.3. Topology Management Through RFID

This section addresses the critical problem of authentication in RFID. It describes a new lightweight scheme for mutual authentication between readers and tags that fulfills the EPCGen2 standard and all practical requirements of low-cost RFID such as resource limitation of tags and minimal interaction between tags and readers. Furthermore, the

proposal does not rely on RFID readers as they are portable, and instead of that, it bases its security on trust in the back-end server because all shared secrets are stored only by the tag and the back-end server, with no possible access by the reader at any time.

- Each node in the MANET has a tag attached so that it can be located.
- Tag readers are added to the MANET and new special nodes that are connected securely to a central server, so that the MANET becomes really a hybrid network.
- It manages to have the topology of the MANET controlled, which facilitates various management issues such as packet routing and broadcast, distribution of tasks, access control, insertions and deletions of nodes, etc.

In conclusion, the proposed scheme in this section is an centralized alternative to the scheme SLCM decentralized presented at the beginning of the chapter.

B.3.1. EPC Gen2 Standard of RFID

RFID (Radio Frequency IDentification) technology implies the use of tags and readers for the purpose of identification through radio waves. A typical RFID system consists of tags, readers, and a back-end server with a database containing information about the tags it manages. Tags and readers are connected through radio communication whilst readers and the back-end server are connected through a secure channel. In 2004 the standard EPC Class 1 Gen 2 (EPC Gen2) was ratified for RFID implementations [173]. According to it, tags are passive, i.e. they reflect back the energy they receive from the reader, so they have their computational capabilities very restricted. Also tag memory is limited and must be considered unsafe and susceptible to physical attacks. In particular, according to the EPC Gen2 standard, tags only support on-chip a 16-bit Pseudo-Random Number Generator (PRNG).

As RFID technology is being used increasingly in more applications, researchers are paying more attention to security and privacy problems such as the unauthorized access to tag ID information, or the existence of potential adversaries that can mislead the reader by using gathered ID information of valid tags. These problems can be solved through

authentication techniques [117]. A comprehensive repository of publications on mutual authentication between tags and readers can be reached online at [7]. We can find various protocols based on different tools such as hash functions, message authentication codes, block ciphers, pseudo-random functions, etc. However, the authentication problem in EPC Gen2 RFID can be considered still open because most proposals require too many resources and/or do not fulfill the standard. Here a new lightweight solution is presented for mutual authentication that fully conforms to the EPC Gen2 Standard.

B.3.2. Basis for the Proposal

A typical mutual authentication solution based on a secret key shared between two entities consists in that each entity has to convince the other that it knows the shared secret key. Thus, to prevent tag cloning, a challenge-response scheme based on symmetric key cryptography can be used. This is the main idea behind the mutual authentication scheme here proposed.

Replay attacks represent another possible weakness of RFID technology. In order to prevent them, typical cryptographic solutions are incremental sequence numbers, clock synchronization or nonces. Passive RFID tags cannot use clocks because they do not have any power supply so clock synchronization is not feasible. On the other hand, incremental sequences are not adequate to avoid tracking. Therefore, in the scheme described in this paper we use nonces. To protect data transmitted between tag and reader against eavesdropping, the typical solution is encryption. In particular, the simplest encryption function is the XOR operation used in stream cipher. However, in that case the problem is not encryption, but key generation and management because it is necessary to produce a new encryption key for each session. This is solved with our authentication proposal. Finally, to prevent tag tracking, the update of tag ID can be used. If tag ID knowledge is only shared between the back-end server and the tag, an easy way to update it is to use the same PRNG both by the tag and by the back-end server, what implies the need for synchronization between tag and server. Such a tag ID update is used in the scheme here proposed.

B.3.3. Mutual Authentication Scheme Reader-Tag

A new mechanism to provide authenticity for low-cost RFID systems fulfilling EPC Gen2 is now sketched. The proposed method can be used by reader and tag in order both to mutually authenticate each other and to establish a shared session secret key. Such a method assumes that the reader is linked through a secure communication channel to a back-end server with a database where each tag t is related to a pair given by a 16-bit secret identification number $ID_{t,i}$ and a 16-bit shared secret key $SSK_{t,i}$ for each session $i = 1, 2, ...$ It is also assumed that both reader and tag are able to use a shared pseudorandom number generator $PRNG$.

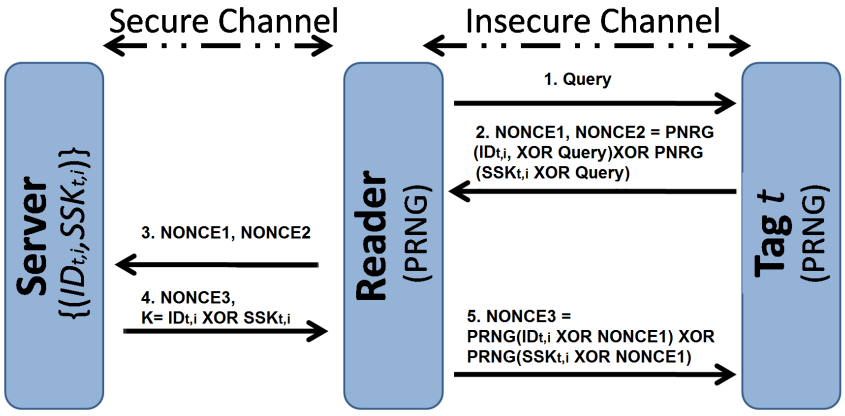


Figura B.14: EPC Gen2 Access Protocol

Algorithm Lightweight RFID Authentication

1. The reader sends to the label a random message Query of length 16.

2. The label t feed the PRNG with ($ID_{t,i}$ XOR Query) and ($SSK_{t,i}$ XOR Query) to produce two cipher sequences of $(16 + n)$ bits whose last n bits are added to be sent to the reader, along with a 16-bit NONCE1.

3. The reader sends the data received in step 2 to the server, and then it is compared to all outputs corresponding to the stored pairs ($ID_{t,j}$, $SSK_{t,j}$).

4. If the server finds no crash, identified as a possible fraud attempt. Otherwise, if it finds a single collision ($ID_{t,i}$, $SSK_{t,i}$), the reader sends the session key as K =$ID_{t,i}$ XOR $SSK_{t,i}$, as the result NONCE3 of the XOR between the last n bits of the two cipher sequences ($16 + n$) bits produced by the PRNG on ($ID_{t,i}$ XOR NONCE1) and ($SSK_{t,i}$ XOR NONCE1) and updates the data of t to ($ID_{t,i+1}$, $SSK_{t,i+1}$). Otherwise, if the server finds more than one collision (although the probability of this happening is negligible), the server informs the reader about the failure to restart the process from the step one.

5. The reader sends to the tag the sequence NONCE3 received.

According to the above scheme we have the following properties.

In the last step, the tag checks if the received data are consistent with the sequence produced by itself on the correct data, and if this check is successful upgrade its EPCdata to ($ID_{t,i+1}$, $SSK_{t,i+1}$).

- After the execution of the five steps of the scheme, both the reader and the tag can use the same secret session key K = ($ID_{t,i}$ XOR $SSK_{t,i}$).

- The information that a spy can obtain from an insecure channel between the reader and the tag by listening the channel is useless.

The established shared secret session key K can be then used both by the tag and by the reader to initialize the PRNG in order to obtain the same key stream Z to encrypt and decrypt all messages exchanged between them during the session. K may be also used then by tag and reader for fast challenge-response authentication based on symmetric cryptography.

In ubiquitous environments we can assume that not many problems of connectivity exist, so for simplicity and practical security in our proposal we have assumed the existence of continuous and secure connectivity between readers and back-end server. The scheme does not provide any information useful for potential eavesdroppers of messages exchanged between tag and reader. On the other hand, without knowledge of the corresponding ID of the tag, it is very difficult to build a value that the server can recognize as valid. Therefore the

proposed protocol actually provides tag authentication. Regarding tag privacy protection and tracking attack prevention, the proposed protocol protects both because the response of the tag in step 2 is random in each authentication request due to the update of its ID and the randomness of the nonce sent by the reader. Note also that the update of its secret identification number and its shared secret key involves forward security feature and resistance against replay attack. In addition, the tag never provides its ID to any reader, therefore there is no possibility that a legitimate and malicious reader can perform impersonation attacks against any tag.

Finally, Man In The Middle (MitM) attacks are impossible to cope with the proposed scheme because they require that the attacker can make independent connections with the reader and the tag in order to relay messages between them to make them believe that they are talking directly to each other when in fact the entire conversation is controlled by the attacker, and this is not possible in our proposal as either the server or the tag detect the attack due to the unknowledge of the attacker about $SEED1$. Thus, if the server detects an attack, it informs the reader that the message received in step 2 does not produce any collision by using $NONCE1$. If with a negligible probability the server finds some random collision after a MitM attack, then the tag will detect the attack when from the message received in step 5 the recovered $SEED1$ does not correspond with $NONCE1$ sent in step 1. Thus, we can conclude that the proposed mutual authentication scheme is immune to MitM attacks.

In conclusion, the proposal described in this section can be considered strong enough to be used for topology management in MANETs.

Appendix C

Vehicular Ad-hoc NETworks (VANETs)

A Vehicular Ad-Hoc Network, or VANET is a technology that uses moving cars as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 metres of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created. It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes.

C.1. Node Authentication

This section proposes a new node authentication solution for the practical, fast and secure deployment of vehicular networks. Its main contribution is a self-managed authentication method that does not require the participation of any certification authority because the nodes themselves certify the validity of the public-keys of the nodes they trust, and issue the corresponding certificates that are saved in local key stores according to an algorithm here proposed. In addition, the new node authentication method includes a cryptographic protocol that each node can use to convince another node about the possession of certain

secret without revealing anything about it. Thanks to all these tools, cooperation among involved vehicles can be used to detect and warn about abnormal traffic conditions. One of the most interesting aspects of the proposal is that the required devices can be simple existing mobile devices equipped with wireless connection. This work includes a performance analysis of a simulation of the proposed algorithms.

With respect to requirement minimization, several papers focus on different aspects and applications of VANETs. [162] proposes a parking notification scheme that does not need any extensive infrastructure, but RSUs are required in the supported parking spaces. [191] proposes a key management scheme for VANETs, which is used to authenticate messages, identify legitimate vehicles and remove malicious vehicles. However, such a proposal is based on the use of a public-key infrastructure. [218] focuses on decentralized vehicular communications without any fixed infrastructure and proposes a method for dynamic establishment of secure communications in VANETs.

There are other bibliographic references that propose different types of authentication schemes for self-managed VANETs, following approaches that are entirely different to the one here presented. [49] proposes an authentication scheme that is based on pseudonyms, while [128] describes a scheme that combines authentication, key establishment and blind signature techniques. With respect to public-key certification, [125] presents a method for certificate revocation based on car-to-car epidemic distribution, and [101] proposes another mechanism for revoking security certificates, which needs a certification authority and certificate revocation lists.

Another related paper with the same general objective as this work is [72], but its authors do not address the issue of the security of communications. Unlike the previous work, two papers that analyze security issues in VANETs are [168], which proposes a security infrastructure based on asymmetric and symmetric cryptography to protect the privacy of users, and [208], which describes a scheme to protect node privacy by using session keys.

The proposal here presented has as starting point the consideration that the introduction of a complete model of VANET including Road Side Units (RSUs) and On Board Units (OBUs) would be extremely expensive both for users, who would have to buy new cars or install specific devices in their vehicles, and for the state, which would have to deploy

a huge infrastructure to support VANET services. Therefore, this work proposes a self-managed VANET following a cross-layer approach that does not require any infrastructure, and that might be used as a fast and secure introduction to more complex and complete VANETs.

Our proposal takes into account that the practical implementation of VANETs will be gradual, without any RSUs or OBUs, and with only a few mobile devices at the beginning. The growth of VANETs will be faster or slower depending on their popularity, acceptance and ease of use. In this paper we focus on the first phase, when the number of cooperating devices on the road will be low. Once VANETs have grown, the model should be checked to avoid unnecessary communications that might degrade the network. In fact, when RSUs have been fully deployed, a different scheme like the proposal in [214] based on vehicle-to-roadside communications might be used

The admission control scheme included in the node authentication proposal described below is based on the general scheme of ZKP used in [44] based on the graph isomorphism problem, for the particular case of the hamiltonian cycle problem.

C.1.1. Node Characterization and Beacons

The present proposal assumes that each node in the network is characterized by the following parameters:

$ID, (KU_{ID}, KR_{ID}), (ID_i, KU_{ID_i}, Cert(KU_{ID_i}))_{ID_i \in KS_{ID}}$

which include:

- A unique IDentifier (denoted ID), obtained as the output of a one-way function on a single value. For example, if the used device is a mobile phone the value can be its number, while in other cases an email address might be used. The one-way function could be any hash function.
- A fixed public/private key pair (denoted (KU_{ID}, KR_{ID})) and called identity keys, which are used in an asymmetric cryptosystem such as RSA.
- A key store KS_{ID} containing various IDs and corresponding public-keys and certificates, which the node keeps always updated, in the form:

ID1	(KU_{ID1})	$Cert(KU_{ID1})$	
ID2	(KU_{ID2})	$Cert(KU_{ID2})$	
ID3	(KU_{ID3})	$Cert(KU_{ID3})$	
.	.	.	.
.	.	.	.
.	.	.	.
ID_{lim}	$(KU_{ID_{lim}})$	$Cert(KU_{ID_{lim}})$	

Sending multicast beacons containing variable sender identifiers are required both for the active node discovery process and also to avoid vehicle tracking. In the same step where beacons are sent, each node commits to its secret by sending to its neighbours also a witness of its secret. The variable identifier of each node that is sent as part of its beacon is the hash of the identifiers that are present in its key store at that moment. El envío de beacons de multidifusión conteniendo los identificadores variables del remitente se requiere tanto para el proceso de descubrimiento de nodos

In particular, the beacons sent by a node are formed by the following elements:

- Frame-Control (FC), which indicates the type of data being sent.
- Pseudonym (Pseu), which is a temporal identifier of the node.
- Timestamp (Time), which allows knowing the specific time when the information was generated.
- Pair formed by public-key and timestamp (KU, Time) encrypted with the private-key (KR) of the node, which is used by nodes who have already authenticated it when its Pseu changes.

C.1.2. Public/Private Key Pair Generation

Within this proposal, the device associated to each network node should be able to generate its public/private key pair and also to sign the public-keys of other nodes that want to become part of the network and are trustable.

In order to make easier the implementation of the ZKP for the hamiltonian cycle in the node authentication process described below, the public-key of each vehicle KU_{ID} is computed from the decimal value of the binary representation for the upper triangular

submatrix of the symmetric adjacency matrix containing the elements corresponding to a hamiltonian cycle in a graph (see Figure C.1).

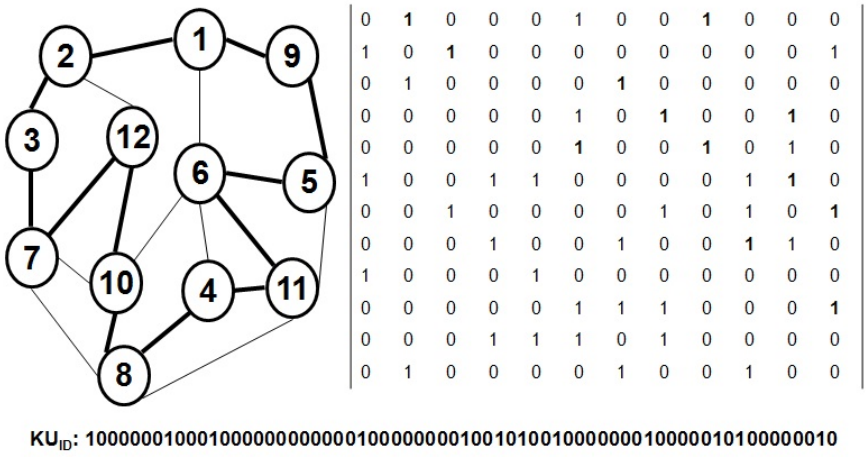


Figura C.1: Example of Hamiltonian Cycle Based Public-Key

The decimal number corresponding to the binary representation is used in the proposal as the exponent of the public-key in RSA encryption used by the mobile device to encrypt and decrypt messages and to sign public-key certificates.

In Figure C.2 we can see a trace of an election of a public/private key pair by using the hamiltonian cycle for the generation of the public-key. After choosing the prime numbers p and q, the public exponent e is generated from a random Hamiltonian cycle, so that it is lower than and coprime with $(p-1)(q-1)$. Afterwards, the private exponent is generated.

In order to be able to authenticate its public-key, every node must exchange signatures with a number of legitimate network nodes that depends on the width of the VANET. At the beginning, two signatures are enough to prove that the user is reliable and cannot self-sign certificates to compromise the network security, but the number of required signatures must grow with the expansion of the VANET.

Self-organized certification of public-keys makes possible to authenticate the public-key of a node without knowing it and with no need of any trusted third party. Such a

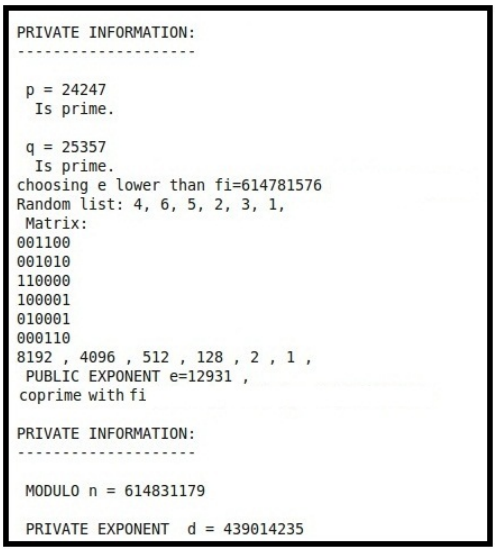


Figura C.2: Implementation of Hamiltonian Cycle Based RSA

certification is based on trusting the neighbours of your neighbours by forming a certificate graph. In this work we understand that a certificate between A and B consists always in two signatures: A signs the public-key of B with its private key, and vice versa.

When a node A wants to check the validity of the public-key of another node B, A must find a certificate chain from it to B in the certificate graph that results from merging the subgraphs G_A and G_B corresponding to KS_A and KS_B respectively.

C.1.3. Scheme Based on ZKP

Special packets are sent between users to authenticate each other. Among other information, they contain the data FC, source Pseu and destination Pseu. Figure C.3 shows schematically all the possible phases of interaction included in the proposed self-managed protocol for authentication between two nodes A and B.

The phases are fully described below. The first phase is the discovering process, which includes part of the beacons sent by node A, and in particular the hash of the IDs stored in KS_A. In the second phase of ZKP authentication, a node B who wants to communicate with A, asks A to send the list of IDs in its store. After that, B checks whether a common key X can be found in the joint $KS_A \cap KS_B$, and in that case both execute a

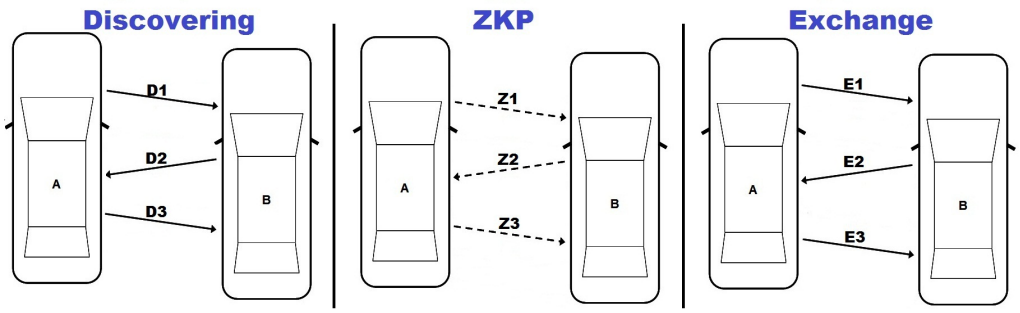


Figura C.3: Self-Managed Authentication Protocol

mutual ZKP on the knowledge of X so that in the last phase both nodes are sure they can use the shared key X to exchange their temporal secret keys and key stores.

Algorithm Authentication Scheme

function *Autentication_Scheme*()() (...)

D1. $A \rightarrow B$: beacon with $\{(ID_i) : ID_i \in \mathrm{KS}_A\}$

D2. $B \rightarrow A$: $\{(ID_i) : ID_i \in KS_B\}$ and a graph $G_B(x)$, if $\exists x \in KS_A \cap KS_B$

D3. $A \rightarrow B$: a graph $G_A(x)$ if $\exists x \in KS_A \cap KS_B$

Z1. $A \rightarrow B$ $(B \rightarrow A)$: a graph $GI_A(x)(GI_B(x))$ isomorphic with $G_A(x)(G_B(x))$

Z2. $B \rightarrow A$ $(A \rightarrow B)$: a binary random challenge $b(a)$

Z3. $A \rightarrow B$ $(B \rightarrow A)$: if $b = 0(a = 0)$ $GI_A(x) \approx G_A(x)(GI_B(x) \approx G_B(x))$

Z3. Otherwise a Hamiltonian circuit in $GI_A(x)(GI_B(x))$

E1. $A \rightarrow B$ $(B \rightarrow A)$: $E_x(KU_A)(E_x(KU_B))$

E2. $B \rightarrow A$ $(A \rightarrow B)$: $KU_A(K_B)(KU_B(K_A))$

E3. $A \rightarrow B$ $(B \rightarrow A)$: $E_{KB}(KS_A)E_{KA}(KS_B)$

end function

The above algorithm allows any node to authenticate another node as well as to exchange both secret keys to update both key stores.

Figure C.4 shows an implementation of the proposed authentication scheme performed using Microsoft Visual Studio in *C#*.

A client-server capable of multiple connections at the same time is implemented in each device. All signals about authentication and beacons are performed with UDP packets.

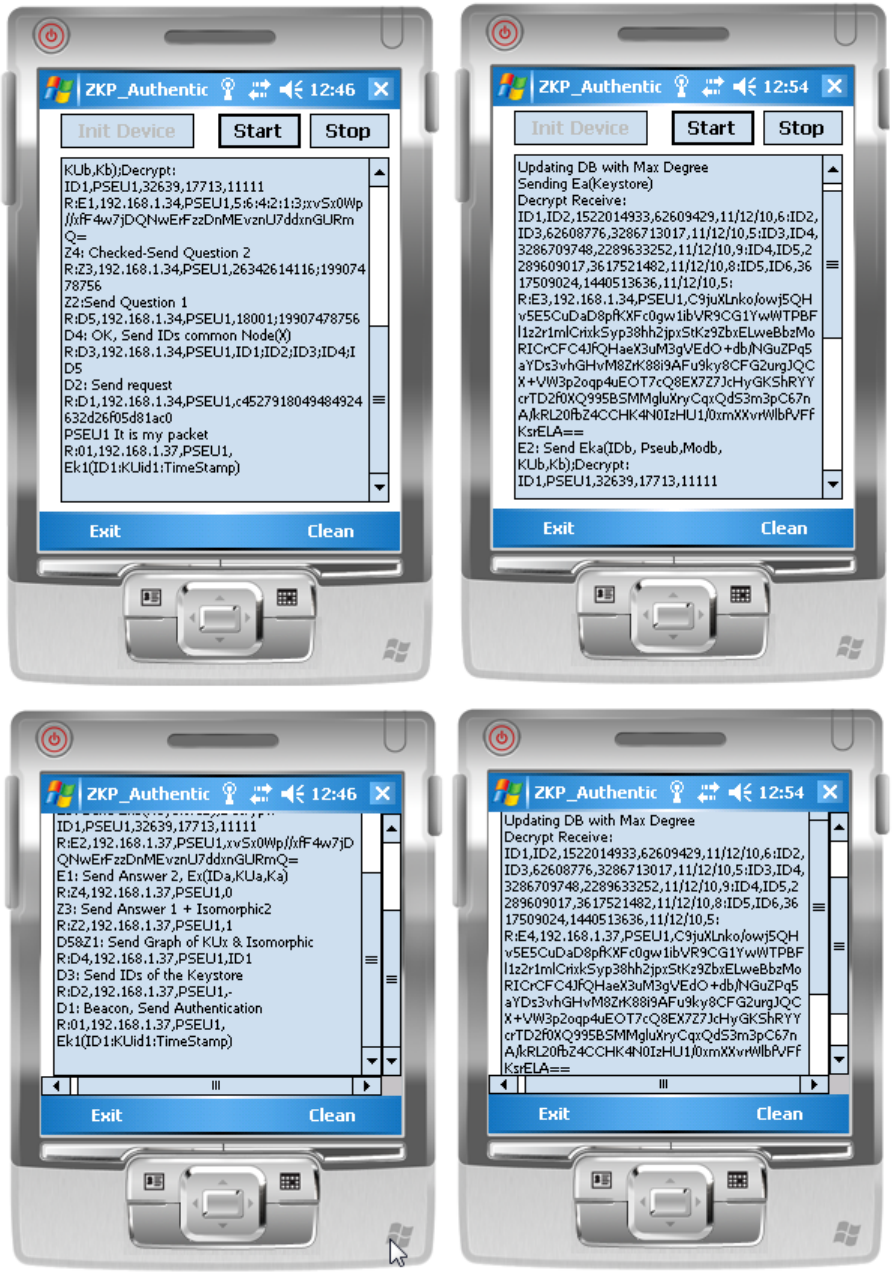


Figura C.4: Implementation of the Authentication Scheme

Each client broadcasts beacons periodically to all connected devices in the network. Each beacon is formed by the following data:

"01," + thisIpAddr + "," + PSEU + "," + Ek1(ID1,KUid1,TimeStamp)

Before starting to use the device, the node needs information to communicate with other devices, and in particular a database with three tables is loaded. These tables keep data for a low number of users whose data (certificates and public key) are generated with the generator C.2:

certificateStore (idcolumn INT PRIMARY KEY, idA NTEXT, idB NTEXT, certAB BIGINT, certBA BIGINT, date DATETIME);

KS (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, module BIGINT, publicKey BIGINT, secretKey BIGINT, degree INT);

myStore (idcolumn INT PRIMARY KEY, idA NTEXT, PseuA NTEXT, modulo BIGINT, publicKey BIGINT, privateKey BIGINT, secretKey BIGINT, degree INT);

Incoming connections are managed on the server so that when one is received, the server checks the identify of the node who sent the packet. After that, it checks whether the node is already authenticated in the network, and if not, the authentication protocol begins. The procedure the nodes use to send and receive information as indicated in Algorithm (Authentication Scheme).

C.1.4. Key Store Update

For the self-managed VANET here proposed, it is required that each node has its own key store in order to authenticate other nodes. In these networks the number of users might be huge, so we propose a scheme for storage of public-key certificates that exploits the aforementioned principle of six degrees of separation. Thanks to such a property, it is not necessary that each user stores the certificates of all nodes in order to be able to authenticate them. Instead of that, only a mininum number of certificates have to be stored by each node so that by merging the key stores of two users who want to authenticate each other, the probability to find at least one certificate chain in the merged graph will be high.

Consequently, the optimal update of the key stores is an important part of the

proposal as it allows to limit the number of stored keys to a value here denoted lim. Such a value is generally less than the number of users forming the network, and equal to the minimum number that allows any node to connect to any other node in the network.

In order to maximize the probability that any node is able to authenticate to any other node while limiting the size of the key stores, different algorithms to update the key stores can be used. A possible algorithm is proposed below. To update its key store, each node chooses those public-key certificates corresponding to nodes that have issued or received more valid certificates, what is represented by the degrees of the vertices is the corresponding certificate graph. This choice maximizes the probability of intersection between key stores, what is necessary for the authentication process.

Algorithm Key Store Update

01:**function** Update_KeyStore() (...)
02: Initialize data structures;
03: Union:= $KS_A \cup KS_B$;
04: $KS_B = \{B\}$
05:**for** each $i \in KS_B$
06: **for** each $j \notin KS_B : (i,j) \in Union$
07: **if** ((degree $(j) = max(degree(neighborofiinUnion))$)
&&$(cardinal(KS_B) < lim)$)
08: $(i,j) \in (KS_B)$
09: **end if**
09: **end for**
09: **end for**
09: **end for**
10: **end function**

An implementation of the proposed key store update scheme has been performed with the Network Simulator tool NS-2.

In the performed simulation, an initial wireless network where nodes are randomly located (see Figure C.5) produces the first certificate graph. At this time each node saves in its local key store the certificates of the nodes at distance 1. Then, the nodes begin to move

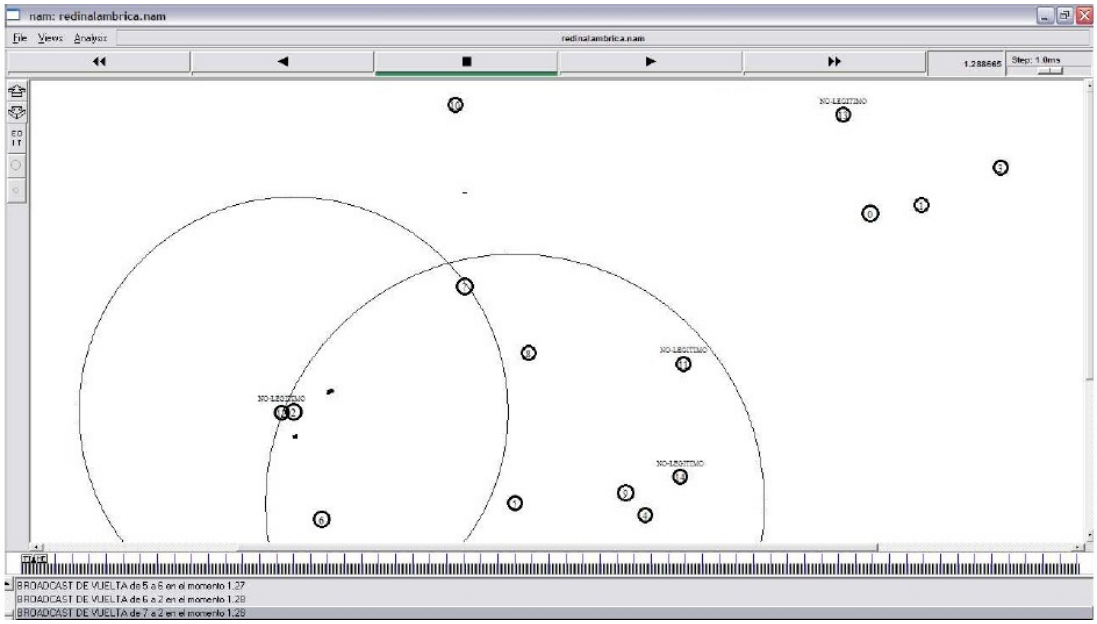


Figura C.5: Initial Network

randomly, and each time two nodes are at distance 1, they verify whether they can trust each other or not (see Figure C.6) and initiate a key store exchange for key store update.

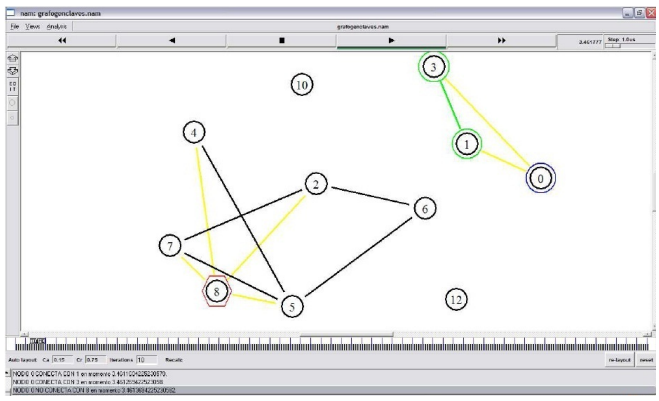


Figura C.6: Trust Checking

In Figure C.6, the node 8 wants to communicate with node 0 but they can not trust each other. However, in the same figure, nodes 8 and 5 want to communicate with node 4 and they can trust each other.

In Figure C.7 we can see a part of the certificate graph after a life period of the

	20 nodes	30 nodes	60 nodes
N Sent Packets	1011,52	2764,7	5309,0
N Receibed Packets	985,4	2749,8	5216,5
N Added Certificates	56,4	80,51	191,9
N Storage Upgrades	420,2	690,24	1475,9

network. In the second image we can see the key store update of node 10 after having exchanged its key store with node 4 and inserted the certificates (8-9) and (9-5) in its key store.

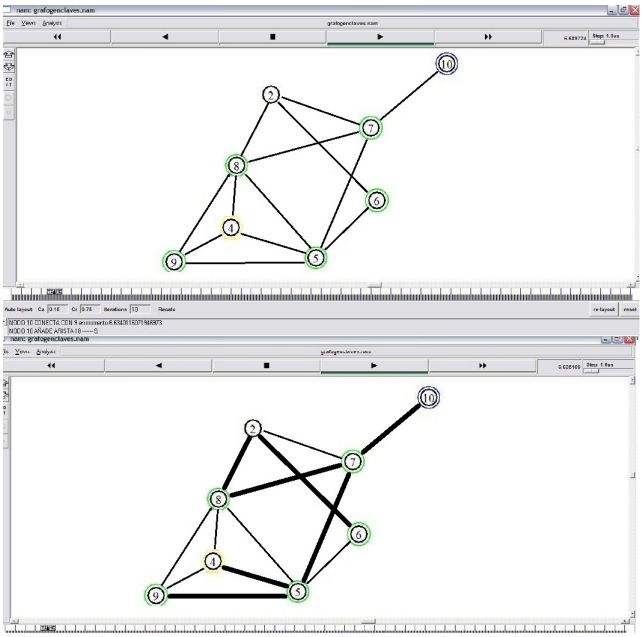


Figura C.7: Key Store Update of Node 10

After performing 25 simulations for 15, 20, 30 and 60 nodes, the average results of executions with different types of networks show that performance may be considered in general acceptable. According to the simulations we can also conclude that the scheme is affected by the mobility of the nodes because a higher mobility leads to a faster increase and balance of the key stores. Thus, this is a convincing arguement why we consider vehicular networks since these are high mobility networks.

C.2. Architecture for Self-organized Clustering

This section proposes the use of clusters to reduce communication overhead generated in VANET scenarios with dense traffic. In particular, a distributed clustering architecture is here presented to create a dynamic virtual backbone in the network, formed by cluster-heads and cluster-gateways so that such nodes are responsible for the efficient message propagation in the VANET.The main aim of the proposal is to balance both the stability of backbone connections and the cost/efficiency trade-off. At the same time, the use of clusters allows combining public-key with secret-key cryptography, what also helps to improve throughput and safety of communications. Full definitions of all the procedures that form part of the proposed clustering architecture are here provided, including a cluster-head selection algorithm based on a version of the independent set problem and a secret key agreement scheme based on a generalization of Diffie-Hellman protocol. Simulations show that our proposal improves VANET performance while guaranteeing real-time delivery of safety-related messages.

C.2.1. Introduction

The proposed architecture implies that one node from each cluster acts as clusterHead (CH), and limits inter-cluster interactions to the CHs. Additionally, since vehicles may produce highly redundant information in VANETs, in order to avoid this problem known as broadcast storm problem, identical packets from different sources in any cluster may be aggregated through different functions such as elimination of duplicates, minimization and/or average, which can be performed fully in each node.

In this section we propose a collection of distributed protocols that allow building a VANET backbone formed by a virtual chain of vehicles to make possible the fast propagation of broadcast messages. The backbone formation and management is performed by exploiting some specific characteristics of VANETs, like the persistence of clustering in common scenarios.

Clusters are here defined as conceptual structures according to which groups of nearby vehicles traveling in the same direction self-organize around their selected represen-

tative called the cluster-head. This special node assumes the role of manager for intra-cluster communications among the members of its cluster, who must be in close communication range.

In our scheme the role of gateways for communications inter-cluster are delegated to other members, depending on their proximity to other clusters. This is shown in Fig. C.8 where four basic states of nodes are identified: cluster-Head (CH), Member Node (MN), GateWay (GW) and Not Definite (ND).

Clusters are especially useful under heavy traffic conditions, when density of vehicles in a close geographic zone is high, such as dense traffic or traffic jams, because in these cases the number of V2V communications is much higher. Under these circumstances, the highly dynamic topology of VANETs can disturb cluster formation and re-organization, increasing cluster instability. Therefore, clustering algorithms must be designed to maintain cluster structure as stable as possible in order to protect the performance of communication.

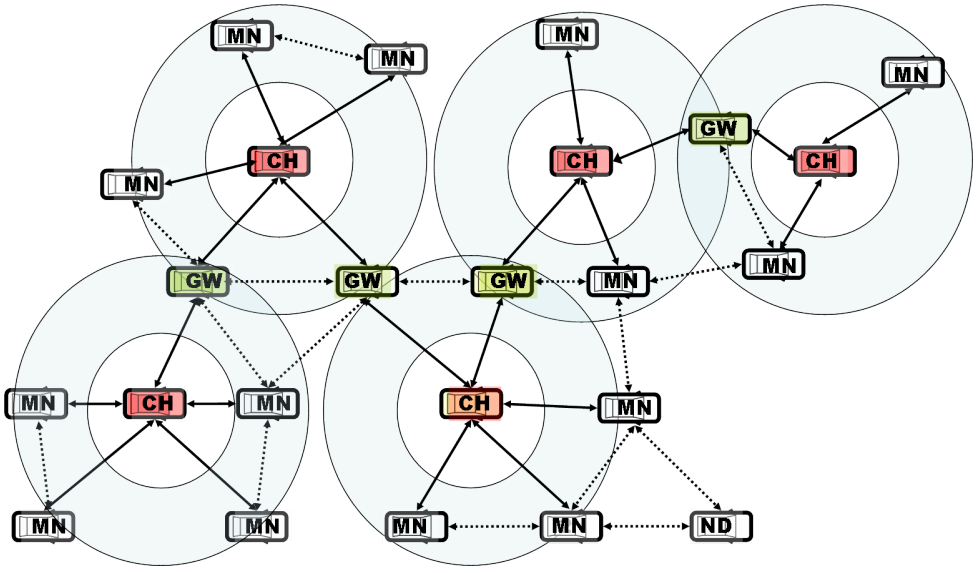


Figura C.8: Basic State of the Nodes

The purpose of the algorithms proposed in this paper is to manage clusters where the CH is directly connected to all nodes in its cluster. In particular, clusters are here defined according to dynamic cells where the CH is the node thathas the largest number of potential neighbor members of its cluster, among other characteristics. In particular,

the decision rule for CH selection takes into account factors such as the average velocity, the average location and the direction of vehicles. Also the state of neighbors is taken into account as no two CHs can be neighbors, and each node must have at least one CH in its neighborhood. Thus, the autonomic definition of clusters implies that vehicles circulating in the same direction and at the average speed of the cluster has a low probability of changing cluster during its route.

Cluster management must satisfy two important requirements. First, it should minimize resource consumption and message exchange. Second, it must take into account the highly dynamic topology of the network.Our proposal implies a significant reduction in the number of retransmissions through broadcast. In particular, if n denotes the number of nodes in the vicinity of a vehicle, without clusters the vehicle sends approximately n packets for every received data and each neighbor receiving such packets are expected to broadcast again the same information so the total number of communications among the $n+1$ nodes in the neighborhood is $n(n-1)$. However, when using the proposed cluster-based scheme only n connections are generated per cluster for each data retransmission. The first connects the member node that first receives or produces the information with its CH, then the CH sends a broadcast to the remaining $n-1$ members of its cluster, including the gateway, who is the responsible for sending the information to neighbor clusters.

In our proposal, nodes are assumed to be periodically broadcasting *beacon* messages containing the following sender's information:

$$< Pseu, loc, speed, dir, state >$$

where

- $Pseu$ is a pseudonym used by the sender in order to enable the other nodes to link messages sent by it, but protecting its anonymity.
- loc denotes the GPS coordinates of the sender's location.
- $speed$ is the speed of the sender.
- dir is the direction of the sender.

- *state* indicates whether the sender is CH, MN, GW or ND.

The GPS coordinates *loc* of neighbors will allow checking at least partially the information about neighbors of neighbors that is sent during the cluster creation phase explained in the following section. The *speed* of neighbors is used not only to decide who will be the CH but also to exclude those vehicles whose speeds are outliers with respect to the remaining velocities of the other neighbors. The parameter *dir* is here used to identify the nodes that can form part of a cluster as all nodes in a cluster must travel in the same direction. These data are also useful to determine the destination of messages as for example some messages have to be propagated only in one direction, but others, such as warnings of congestion due to an accident, must be propagated in both forward and backward directions. Finally, with respect to the parameter *state* since in our scheme all nodes have to belong to some cluster, at least formed by itself, the state ND can only be used for the initial state of the node before executing the protocols described in the following section. Furthermore, when a node belongs to more than one cluster, it becomes a GW for the inter-cluster communication, forming part of the backbone for message propagation in the VANET.

C.2.2. Notation and Architecture Description

This section contains the description of the procedures that form part of the proposed clustering system architecture, including all the details of every possible stage in cluster management, depending on the specific situation of vehicles.

In the following, the basic notation used throughout the algorithms that form part of the proposed architecture is described.

- x denotes the executor node.
- NeighborsCH(x) is the set of CHs neighbors of x.
- neighbor(i) denotes the i-th neighbor of the executor node, which is a potential cluster member.
- isCH(i) is a boolean function indicating whether an input node i is CH or not.

- CreationRequest(x) represents a message sent by x containing a cluster creation request.
- Receive(Message,i,x) indicates that Message from node i is received by x.
- ClusterList[] contains the members of the cluster.
- CHNom is a message stating that the sender node will be CH.
- Weight(i) is the value associated to node i that indicates how suitable it is for the role of CH according to parameters such as its number of neighbors, location, speed, etc.
- KeyRequest(x) represents a message sent to x containing a key share request.
- p is a prime number.
- g denotes a generator element of Z_p.
- S_i is an integer in $[0, p-2]$ randomly chosen by node i.
- g^{S_i} denotes the public commitment of i with integer S_i.
- h stands for a hash function.
- K_x is the secret key of the cluster with cluster-head x.
- Wait(T) implies to wait for a time T before proceeding with the next step.

C.2.3. Vehicle Initialization

This is the first stage that is launched when a vehicle is in its initial conditions and its state is ND because it does not belong to any cluster yet. This stage is described in Algorithm 1, that the node executes in order to discover whether there is a CH nearby or not.

Every vehicle that is in ND state periodically has to check whether among its neighbors travelling inside the range of its own speed, some CH exists. If there is at least one candidate neighbor who is a CH, the node proceeds to the join procedure. Otherwise, it

proceeds to the cluster creation stage. Note that this stage does not generate any additional traffic of control due to the fact that all the necessary information to execute it is contained in the beacon messages that nodes periodically broadcast.

Algorithm Vehicle Initialization

```
01: function VehicleInicialization (...)
02:   i = 1;
03:   cardinal(NeighborsCH(x)) = 0;
04:   while (exist neighbor(i)) do
05:     if isCH(neighbor(i)) then
06:       cardinal(NeighborsCH(x))++;
07:     end if
08:     i++;
09:   end if
10:   if (cardinal(NeighborsCH(x)) == 0) then
11:     ClusterCreation();
12:   if not
13:     for (j=1; j ≤ cardinal(NeighborsCH(x));j++) do
14:       ClusterUnion();
15:     end if
16:   end while
17: end function
```

C.2.4. Cluster Creation

This cluster creation stage (Algorithm 2) is launched every time a node is in ND state, has previously executed the initialization phase and has discovered that it is not close to any CH. In order to begin a new cluster creation process, the executor node broadcasts a cluster creation request towards all neighbors traveling in the same direction, and with distance equal to 1 and speed inside the range of speeds in the neighborhood. Nodes that receive this request respond accepting the invitation and indicating the number of its neighbors that are candidates to become members of a new cluster having itself as

CH. After this, the CH selection stage will be launched by the nodes who answered to the invitation, and the shared secret key will be established according to the cluster secret key agreement protocol. After that, the new cluster can be considered completely established.

Algorithm Cluster Creation

```
01: function ClusterCreation (...)
02:   l = 1; i = 1;
03:   Retransmit(CreationRequest(x));
04:   while (exist neighbor(i))
05:     if Receive(Answer,i,x) then
06:       ClusterList[l] = neighbor[i];
07:       l++;
08:     end if
09:     i++;
10:   end for
11:   if (l ≥ 1) then
12:     SelectionCH(ClusterList[]);
13:   if not
14:     state=CH;
15:   end if
16: end function
```

In conclusion, this cluster creation stage basically requires a broadcast of invitation to join the new cluster and unicast responses from the n receiving candidate nodes, what means a total of $2n$ packets. Consequently, management packets generated at this stage do not decrease the communication throughput.

C.2.5. Cluster Head Selection

In this section an algorithm to select a node as CH of a cluster is proposed. The main idea of the CH selection algorithm is to allow a node to evaluate its potential as a CH before becoming one and stepping down if it is not the best CH at the moment. When a node decides to become a CH, it broadcasts an invitation message to recruit its neighbors.

After getting the invitation from the new CH, the neighbor nodes join the new cluster. Each CH periodically checks the ability of its cluster members for being a better CH than itself and, if one of these neighbors is a better candidate for CH, it decides to step down and propose such a node to become the new CH. This renovation process is also executed automatically if the CH leaves its cluster.

The criteria that are used for the CH selection are multiple. On the one hand, the CH has the least probability (when compared to others within the same cluster) to move out of the current virtual cluster because its velocity is close to the average velocity in the cluster. This ensures that a highly mobile node related to its neighbors is not elected as a CH. At the same time, the efficiency of intracluster communications is maximized with the election of the CH because the CH has the minimum distance from the respective virtual cluster centre.

The problem of arranging VANET nodes into clusters is here treated as the problem of finding a maximal weighted independent set of nodes. We introduce a distributed algorithm for the efficient determination of a maximal weighted independent set in the graph that represents the VANET, which only requires that every node has certain knowledge of its neighborhood.

The maximal weighted independent set problem is a well known NP-hard problem [91], but in this paper we are only interested in solving the problem for the specific class on graphs that represent the topology of a VANET.

According to our cluster definition, no two CHs can be neighbors. Furthermore, the network has to be covered with a backbone of CHs (and GWs), what implies that each node must have at least one CH in its neighborhood. Consequently, the clustering problem can be reduced to the hard problem of finding a maximum independent set of nodes in the network graph. In particular, in our solution we associate one weight to each node to indicate how suitable it is for the role of CH according to parameters such as number of neighbors, location, speed, etc. Therefore, the algorithm for selecting the CHs is equivalent to the problem of finding a maximal weighted independent set in the graph of the network and the nodes in the independent set will be the CHs. In order to execute the distributed Algorithm 3, each cluster member only has to know the weights of its neighbors. Initially, only those

nodes with bigger weights with respect to their neighborhood broadcasts a message to their candidate neighbors stating that they will be the CH. In a second round, if a node does not receive any of these messages, it broadcasts one of them. Otherwise, it checks whether its role is MN or GW.

Algorithm Cluster Head Selection

```
01: function CHSelection (...)
02:    CHNom = 1;
03:    for (i=1; i ≤ Cardinal(ClusterList);i++)do
04:       if weight(i) > weight(x) then
05:          CHNom = 0;
06:       end if
07:       if CHNom == 1 then
08:          state=CH;
09:          Relays(CHNom);
10:          SecretKeyEstablishment(ClusterList[]);
11:       si no
12:          if Cardinal(Receive(CHNom, ClusterList[],x))==0 then
13:             state=CH;
14:             Relays(CHNom);
15:             SecretKeyEstablishment(ClusterList[]);
16:          if not
17:             if Cardinal(Receive(CHNom,ClusterList[],x))==1 then
18:                state=MN;
19:             if not
20:                state=GW;
21:             end if
22:          end if
23:       end if
24:    end for
25: end function
```

C.2.6. Cluster Secret Key Establishment

Most references about secret communications in VANETs suggest the use of public key cryptography based on a Public Key Infrastructure (PKI) with certificates issued by a Certification Authority (CA). This solution implies that a public/private key pair is assigned to each node and stored in its tamper-proof device and public-key certificates are authenticated either by a centralized or a distributed CA.

Our proposed cluster-based management scheme allows combining PKIs with the use of secret key cryptography. It assumes that each message sent in a VANET contains a digital signature that can be used to identify the sender node, but adding certain communication and computation overhead. In order to reduce this overhead, the establishment and use of secret keys shared in clusters is here proposed because secret-key cryptography is in general more efficient than public-key cryptography. The big size of VANETs disallows vehicles from preloading shared keys, so secret key establishment must be dynamic. Note that communication with shared secret key and proximity of cluster members who communicate in promiscuous mode, allow nodes of the cluster to control that both the CH and other nodes in the cluster act properly.

In order to preserve equal roles of OBUs in VANETs, we take advantage of the distributed nature of the proposed clusters to define a key agreement process as general recommended approach for key establishment. Several methods can be used, but our proposal implies that the CH broadcasts certain information to all members, which allows them to compute independently the same shared secret key. The proposed agreement protocol establishes a secret key for all the members of a cluster, based on each node's contribution exchanged openly over an unsecured wireless medium. The secret key derived with the Algorithm 4 can be used to establish a secure channel between all the members of the cluster.

In particular, in the scheme described below, nodes forming a new cluster generate a shared secret key through a scheme based on the difficulty of the discrete logarithm problem, which consists in computing the value of S, given $g^S (mod\ p)$, g and p. This problem is the basis of the well known Diffie-Hellman method to exchange a shared secret between

two parties. Consequently, this work proposes the use of a generalization of Diffie-Hellman key agreement protocol to more than two users.

The following algorithm is based on a bit-commitment scheme so that each node i commits to the CH its contribution to the shared secret key. In this way, the CH, which in Algorithm 4 is denoted by the executor node x, can neither change this contribution, nor read it. The use of a commitment scheme makes possible the exchange of public information for enabling the generation by each node of the shared secret without putting the shared secret key or the different contributions at risk.

Algorithm Secret Key Establishment

01: **function** SecretKeyEstablishment(...)
02: Relays(KeyRequest(x));
03: **for** (i=1; $i \leq$ Cardinal(ClusterList);i++) **do**
04: Receive(g^{S_i} $(mod\ p)$, i, x);
05: **end for**
06: Relays($\{h(g^{S_i}), g^{S_i S_x}(mod\ p)\}\ \forall_{i \neq x}$);
07: $K_x = g^{S_x(1+\sum_{i \neq x} S_i)(mod\ p)}$;
08: **end function**

Note that the broadcast in step 6 of the above algorithm poses no threat to the secret of the cluster key as it is useless for any node that has not contributed to the secret. It is also important to remark that although the above algorithm is launched by the CH, every cluster member i can check if its contribution was correctly included in the message sent by the CH. In such a case, it can compute independently the cluster secret key with the message received from the CH, by removing its share from $g^{S_i S_{CH}}$ to get $g^{S_{CH}}$ and then computing the secret key according to the expression:

$$K_{CH} = g^{S_{CH}} \cdot \prod_{i \neq CH} g^{S_i S_{CH}} = g^{S_{CH}(1+\sum_{i \neq CH} S_i)}(mod\ p)$$

According to the aforementioned algorithm, the cluster key is generated with the contributions of the first members of the cluster. While the cluster exists, those nodes that join it receive the secret key encrypted with the public key of the new node from the CH.

Note that the proposed use of clusters reduces overhead but in general does not allow defining different security levels among members. Instead, it mainly protects the network from potential outsider attackers. Hence, delivering existing secret keys shared within a cluster during member joins is required but member leaves do not involve any update for the cluster key.

Secret-key encryption is in general more efficient than public-key encryption, so thanks to the shared secret key establishment process proposed above, any secret-key encryption can be used in VANETs to get confidentiality through secret-key encryption. Apart from safety-relation applications, other scenarios exist where confidential communications may be necessary. This is the case for example of certain commercial applications.

C.2.7. Joining a Cluster

This stage starts when a vehicle finds among its neighbors at least one node that is CH. Algorithm 5 shows the stage according to which a member node joins all the clusters corresponding to CHs in its neighborhood.

In order to proceed with this stage, the node first has to send a login request encrypted with its public key to every CH neighbor. After authenticating it, the CH sends its corresponding cluster secret key encrypted with such a public key and in this way, the vehicle becomes part of the cluster and proceeds to the maintenance phase.

Algorithm Cluster Union

```
01: function ClusterUnion(...)
02:    for (j=1; j ≤ Cardinal(NeighborCH(x));j++) do
03:       KeyRequest(j);
04:       Receive(K_j, j, x);
05:       ClusterMaintenance();
06:    end for
07: end function
```

C.2.8. Cluster Maintenance

Mobility in VANETs is usually highly dynamic. For instance, nearby vehicles usually drive close to each other for several kilometers while other vehicles bypass them quickly. This is the main reason why it is necessary to be continuously executing the cluster maintenance phase, consisting in checking the validity of the clusters.

Algorithm shows the process that a MN or GW has to carry out while it belongs to the cluster. The node checks that has not lost contact with its CH every T time units. The node considers that it has lost contact with its CH if it has not received any message from its CH for two times T. In such a case, it changes its state to ND and begins the initialization stage.

Algorithm Cluster Maintenance

```
01: function ClusterMaintenance (...)
02:    while (Receive(Message,CH,x) do
03:       Wait(T);
04:    end while
05:    Wait(T);
06:    if (Receive(Message,CH,x) then
07:       ClusterMaintenance();
08:    if not
09:       state=ND;
10:       VehicleInicialization();
11:    end if
12: end function
```

C.2.9. Message Management

Thanks to the use of clusters, the number of communications can remarkably decrease without missing any useful information. Algorithm 7 shows the steps that a node has to execute after receiving or generating a message. El algoritmo de gestión de mensaje muestra los pasos que un nodo debe ejecutar después de recibir o generar un mensaje.

If the executor node is the final destination of the message, it simply processes the information. Otherwise, its reaction depends on whether the node is CH or not. In this last case, it sends it through encrypted unicast to the CH. If the executor node is CH, its action depends on whether the message is to be propagated or not. In the first case, it sends through encrypted multicast it to all its cluster members. Otherwise, if the message has a single destination and this destinations belongs to its cluster, it unicasts the message to the destination through encrypted intra-cluster communication. Otherwise, if the single destination does not belong to its cluster, it sends through encrypted unicast the message to all GWs, who will forward it to other CHs through encrypted inter-cluster communication.

Note that the existence of GWs is of high importance because they provide some degree of overlap among clusters that facilitates inter-cluster communication not only for spreading messages but also for other applications such as topology discovery and node localization.

It is also remarkable that especially when a message has to be propagated and when a message requires inter-cluster communication, mechanisms for enforcement cooperation can be useful because without them, intermediate vehicles may not have the necessary incentives to relay others' connections.

Algorithm Message Management

```
01: function MessageManagement (...)
02:   if(Destination(Message)==x) then
03:     Processes(Message);
04:   if not
05:     if (state==CH) then
06:       if (Cardinal(Destination(Message))==1)) then
07:         if (Destination(Message) in ClusterList[]) then
08:           Unicast(Message,x, Destination(Message));
09:         if not
10:           Multicast(Message,x, GWs);
11:         end if
12:       if not
```

```
13:              Multicast(Message, ClusterList[]);
14:          end if
15:       end if
16:       if not
17:          Unicast(Message,x, CH);
18:       end if
19:    end if
20: end function
```

C.3. Simulation of the Proposals

Both the feasibility and effectiveness of our approach are shown through the following figures where a simulation exemplifies its performance. In the first part of our demonstration (see Fig. 2), a NS-2 and SUMO display shows the VANET state when clusters are operating. In this simulation we compare three models: a VANET implemented without clusters, our cluster based proposal and the CARAVAN approach, where every node is in the range of transmission of all nodes in the cluster. Static clusters or clusters with more than one hop have not been taken into account because the differences in number of communications are obvious. On the one hand, with static clusters every node changes cluster many times. On the other hand, by using more than one hop, the number of control packets is higher than by using only one because the CH has to know all the two-hop distance nodes.

The most relevant choices for the demonstration have been:

- total number of vehicles: 80,
- number of vehicles with OBUs: 80,
- number of lanes for each direction: 3,
- simulation time: 100 seconds,
- moment when retransmissions begins: 40 seconds,
- retransmission period: 10 seconds,

- distance relay nodes: 75 meters,
- traveled distance before the traffic jam happens: 800 meters.

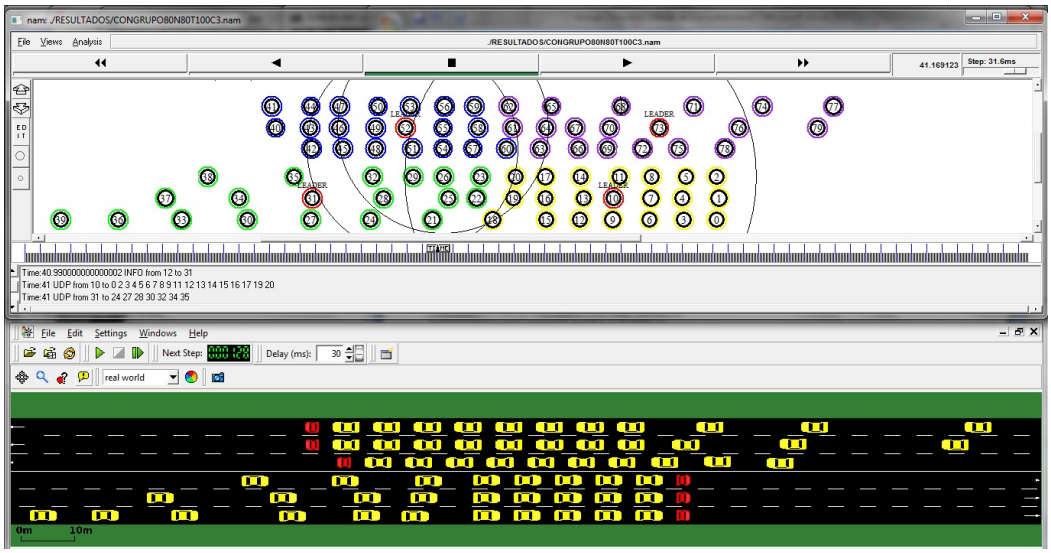


Figura C.9: Simulator Snapshot

The implemented simulations for clusters consider four layers of development: vehicle mobility, node energy, cluster formation and P2P communications:

- The vehicle mobility layer manages the node movement in the simulation pattern, which defines roads, lines, different speed limits for each line, traffic jams, etc.
- The node energy layer is used to distinguish between vehicles with and without OBUs. Vehicles without OBUs are present on the road but do not contribute to the communications.
- The cluster formation layer defines which vehicles belong to each cluster and their roles, that is to say, who is the CH of each cluster, who generate traffic information and who are the GWs and relay information to other clusters.
- The P2P communications layer is responsible for the definition of which nodes are in the transmission range of the retransmitting node at any time.

Simulations give essential statistics such as numbers of generated and lost packets. These basic statistics data are useful to make efficient simulations for large scale scenarios.

Both the implemented simulations with our proposal of clusters and with CARAVAN can be compared with results obtained from the simulation without clusters with the same topology (see Fig. 3). This helps to illustrate the cluster-based vehicular network evaluation. Simulated VANETs are formed by vehicles connected by WiFi 802.11b/g, which is estimated to reach between 50 and 300 meters. The simulation does not consider Doppler effect because in dense traffic conditions this factor is minimized due to the low speed of vehicles.

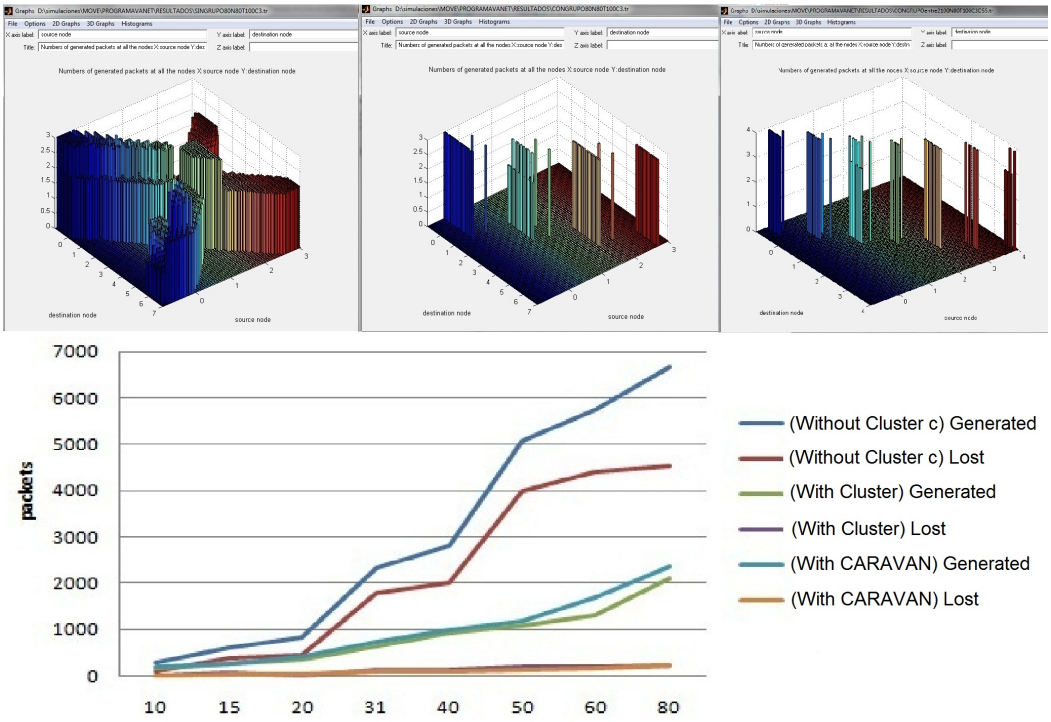


Figura C.10: Packets Without and With clusters, and with CARAVAN

Among the obtained information from the simulations we have the number of packets and bytes generated, sent, broadcast, received, lost, etc. for each node.

Table C.1 shows some simulation results. Also, other shown information is the number of packets generated or lost in the whole network, the number of formed clusters,

which nodes are the CHs, which nodes generate packets and which nodes forward them, etc. In addition to all this information, another interesting aspect is that it provides a detailed simulation of what happens in each moment in the VANET thanks to the use of the NS-2 display. It also shows the traffic model through the SUMO tool while the information is represented using TraceGraph.

Tabla C.1: Simulation Results

	With Clusters		Without Clusters		With CARAVAN	
n. of nodes	packages sent	packages lost	packages sent	packages lost	packages sent	packages lost
10	278	107	167	12	187	20
15	598	402	277	43	271	48
20	825	443	351	0	427	52
31	2343	1804	638	79	749	88
40	2805	2014	932	95	1009	100
50	5077	3981	1101	132	1190	137
60	5732	4415	1314	159	1693	168
80	6675	4529	2120	215	2357	232

In Fig. C.11 we can see a comparison between the average generated and lost packets. It is clear that without the use of clusters in VANETs, the number of generated packets grow up much faster than with the use of clusters, and so it does the number of lost packets. The main reason is likely to be the heaviest traffic load that VANETs generate in traffic jams conditions. Indeed, the original protocol makes a massive use of broadcast operations. It is clear that clusters help to decrease the percentage of lost packets and to perform VANET operation. In the comparison between our proposal and CARAVAN, both the numbers of generated and lost packets are lower in our proposal. Furthermore, the retransmission time in CARAVAN is bigger than in our proposal because the formed clusters are smaller and the number of nodes through which the information passes is greater.

Fig. C.11 shows the average size of the clusters for different densities in our approach and in CARAVAN approach. The largest clusters are those with the highest density of vehicles. The proposal was simulated in a realistic traffic jam environment, with a three-

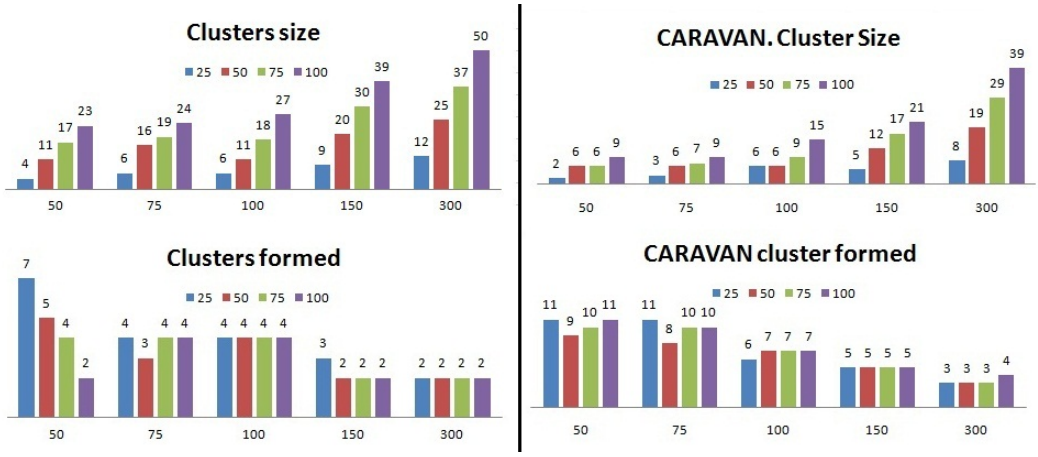


Figura C.11: Size and Number of clusters

lane highway in each direction and 100 vehicles. It can be seen that the number of vehicles inside the cluster increases in a linear way. The second graphic shows the average size of the clusters using CARAVAN approach. In this case we can see that the number of nodes belonging to each cluster is smaller than in the first approach, and they are about half the nodes of our approach. Fig. C.11 also shows the number of clusters formed using the same parameters as above. In this case we can see that increasing the range of transmission reduces the number of clusters. This is a normal result because the size of clusters is larger. In CARAVAN approach we can see the same but in this case the number of clusters is bigger than in our approach. This is because all nodes must have connection with all nodes of the cluster and the clusters in CARAVAN have a smaller size.

Appendix D

VANET in Phones (VAiPho)

This chapter contains the main details of actual implementation made with some algorithms designed in this thesis as part of a mobile application developed called VAiPho (VANET in Phones) to deploy a real VANET immediately useful in detecting traffic jams and squares parking, and more. The main ideas in this chapter were subject of publication in [31], as well as patent in [35] whose operating license has been acquired by the company DETECTOR [71]. In particular, the chapter is divided in two sections that specify the interfaces and schemas used for the basic implementation, as well as ideas to solve the practical problem of merging subnetworks.

In particular when starting a vehicle, the device synchronize with the GPS and keeps the vehicle coordinates in the database. Then the coordinates are signed with his private key and sends them to other devices that authenticate with it, if the period of validity of the parking is valid. Nodes that receive this information, and they have begun the search for parking, will see an icon with the possible parking drawn on their map.

When designing a tool to create a self-organized vehicular network with the goal of increasing road safety, the first prerequisite to be considered is the accuracy and reliability of transmitted information. Thus, security is the most important topic to be taken into account when a communication system is designed for VANETs [172]. Vehicular ad-hoc networks are susceptible to different kinds of attacks. In the bibliography we find some proposed schemes for self-organized VANETs [1] [54] [163] [212], which try to solve all

or part of the security problems existent in these types of networks. However, a different approach is presented in this paper. The work [72] has the same objective as this work but it does not address the main aspect of security of connections. Furthermore, its general approach is different. Another important security aspect is user anonymity, We will use a random pseudonym generator to guarantee with a high probability that it is not possible that an attacker can track a vehicle. This generator also avoids coincidences in two generated pseudonyms [180]. In [16], a pseudonyms scheme is proposed, the authors try to solve the problem of privacy when coordinates and speed of the vehicles are sent in the beacons. In our proposal, coordinates nor privacy are sent in beacons.

Our proposal takes into account that the integration of VANETs will be gradual, so that at the beginning, there will not be any kind of Road Side Unit, and the VANET will start from a few vehicles, to a larger number of vehicles. This growth will be faster or slower depending on the popularity, acceptance and ease of use of VANETs. For this reason the user interface is an important requirement. An ideal complement to VAiPho is described in [14] where the mobile device applications are displayed in a larger terminal. VAiPho can be also combined with [2] so that the user could interact with the device using voice to ask for information such as *"show free parking lots"* or *"show traffic jams in route X"* where route X must be previously stored.

As for the solution to the search for parking spaces there are fewer solutions, none of them implanted. [142] presents a solution to the search for parking where through a device installed in the passenger door, the free parking are found and reported to a server, from which users can locate free parking slots. This solution also requires the use of 3G or GPRS to operate. In [162] authors show a solution where users can find parking slots in a secure network without extensive infrastructure support. [60] presents a solution to find a park slot depending on the distance of the user, they can know the exact information of a park slot or the number of park slots in a area if they are far. The solution to find the parked car is the easiest to implement and more applications can be found for differents mobile platforms. [15], [55], [56], [90], [127], [133].

D.1. Design and Implementation of VAiPho

This section show the interfaces and operating mode of VAiPho (VANET in Phones), a new tool that provides solutions to ensure the operation of VANETs for detecting traffic events on the road and exchanging information about them using only mobile phones without any central authority. VAiPho interface has been designed to avoid distractions while driving because it operates automatically and independently of the driver through voice prompts. Besides, it takes into account security features that are essential in networks such as authenticity, privacy, integrity and non-repudiation. All this is achieved without increasing the price of vehicles and without integrating new devices, simply by using mobile phones equipped with wireless connection and GPS.

Among the different types of information that can be made available to vehicles through VANETs, safety information to prevent accidents and traffic conditions to reduce congestions are the most important ones.

D.1.1. VAiPho Utility

The main goal of this work is to define a simple, scalable model for VANETs where users can cooperate through their mobile devices and obtain updated information of interest about the traffic area in order to choose the best updated route to their destinations. Every year there are more and more traffic jams on the roads because the number of vehicles continues increasing very fast, so that in 2011 there are about 1000 million cars in the world. In [203], authors estimated a loss of $78 billion in 2007 in the form of 4.2 billion lost hours and 2.9 billion gallons of wasted gasoline in the United States alone.

Traffic jams produce high levels of stress in people. Furthemore, it is the main cause of air pollution. It has been shown that people caught in traffic are three times more likely to have a heart attack than those who are not stuck in a jam. It is not clear whether heart attacks are due to traffic-related stress or to exposure to high levels of pollution. Anyway, communications among vehicles could help to prevent those problems by reducing traffic jams, what would also avoid enormous wastes of time and money of users, and of oil reserves.

Nowadays, many centralized GPS software applications offer traffic services based on feeds from local road authorities, police departments and systems that track traffic flow. However either they are not real-time data so they do not reflect the events that have just produced, or they do no respect users' privacy so that people is reluctant to use them.

Currently there are many projects for the congestion detection solution, Google Traffic [96], TomTom [202], Sygic [195] or Waze [209] are solutions to detect traffic jams. The main differences with a VANET is that all of them need to be connected by 3G or GPRS to work and another disadvantage is that users completely lost their privacy because they have to give the information to companies that support the service.

The main result of this thesis is VAiPho, a secure communication system for spontaneous and self-organized networks based on smart phones with GPS and wireless communication and without the need of any infrastructure in vehicles or on roads. The operating mode is completely distributed and decentralized. VAiPho takes into account the protection of privacy and integrity against different possible attacks. Its main goal is to increase passenger safety and comfort thanks to the exchange of warning messages among vehicles. It also helps to reduce CO2 emissions, increase efficiency avoiding wasted of time and fuel in traffic congestions and enhances comfort by cutting down the hours on the road to reach the destination and the number of abrupt decelerations. Furthermore, its structure allows taking advantage of other services such as free parking space detection and parking location remainder.

In this section we focus on the first phase of VANET deployment, when the number of cooperating devices on the road is low. As soon as the VANETs expand, the model should be checked to avoid unnecessary communications so that the high number of communications does not degrade the network. This issue is also being studied so that possible solutions have been proposed based on groups in this thesis.

D.1.2. Requirements

The mobile phone application development has gone multiplatform, so VAiPho currently is being developed for Windows Mobile, Symbian and Android. The following system specifications see Fig. D.1 are necessary for the optimal use of VAiPho acts as

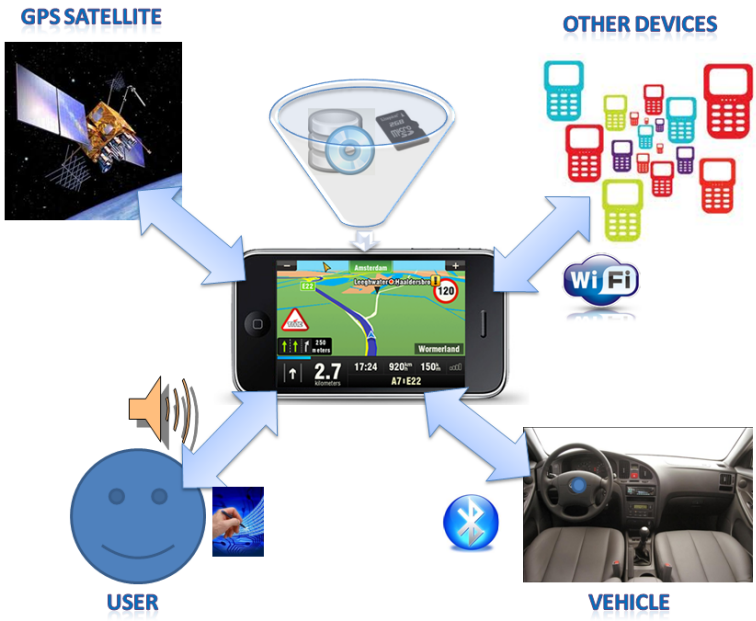


Figura D.1: Connections with Interfaces

application platform:

- Bluetooth®: Allows the connection of the device with the vehicle providing automatic activation of VAiPho without requiring user attention.
- Wi-Fi® IEEE 802.11 b/g: Allows the free exchange of information on possible events between different devices.
- Database: Allows the storage of different events, as well as information on possible parking lots and reminder about where the vehicle is parked. It also stores other users' information to authenticate them.
- GPS antenna: Allows obtaining coordinates where the different events happen as well as speed and direction in which the vehicle is circulating.
- Storage Space: Provides enough capacity for storing necessary programs.

As we can see, VAiPho uses the powerful application platform capabilities of today's mobile devices and so it is not expensive for users. Furthermore, people are used to

working with them and then the user interface of VAiPho is not strange for users, what avoids the usual difficulties of learning to use a new device.

VAiPho wireless communications are performed using the wireless communication standard IEEE 802.11b/g. This standard is not the best suited for road safety applications because these communications, which are in the 2.4GHz range emissions may interfere with other devices such as Bluetooth, Zigbee or other WLAN networks that can occupy the same channel emission that VAiPho network.

Added to this problem there are other problems such as timing problem, antenna of the mobile that has limited size of retransmission and its lack of direcction, battery consumption of the mobile communication, and finally, IEEE 802.11b/g is not designed to work at different speeds of vehicles.

The standar of wireless communications IEEE 802.11p [116] has been designed to this proposal and it is more appropriate for this type of communications, however, currently there are no devices capable of broadcasting in the frecuency range that IEEE 802.11p uses, but most of the smartphones that are made today have the capability to communicate with the IEEE 802.11b/g. VAiPho is a tool for mobile phones, for this reason it can expand quickly and easily growing to a considerable number of users. VAiPho is designed in modules, in the future the communications module can be easily adapted to standard IEEE802.11p when the standard is sufficiently widespread.

There have been many test of communication by using the IEEE802.11b/g, these tests have been satisfactory. By this standard, vehicles traveling in opposite directions do not have enough time to communicate, but the vehicles in the same direction or into the city where the speed is low have time enough to establish communications and exchange data.

D.1.3. VAiPho Structure

VAiPho is composed by three main applications perfectly differentiated, as shown in Fig. D.2:

- *VAiPho WatchDog*, which runs in background, is very light and is in charge of detec-

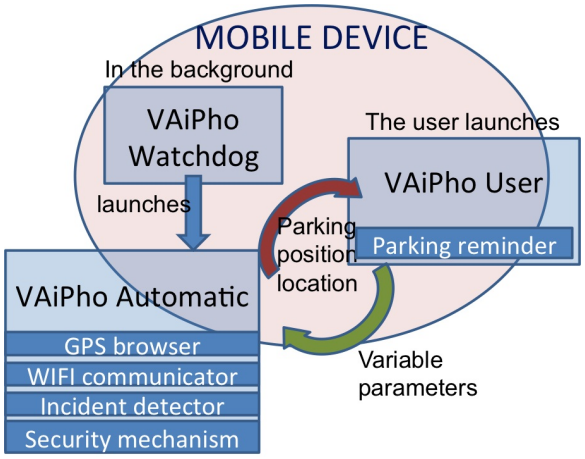


Figura D.2: VAiPho - Functions of the different Interfaces

ting Bluetooth to connects with the car hands free.

- *VAiPho Automatic*, which starts the automatic application that is responsible of detecting and forwarding the events that happen on the road.

- *VAiPho User Application*, which is specially designed for the users allowing them to interact with it providing interesting functionalities.

In this section, these applications are explained and their internal structure is fully detailed.

VAiPho Watchdog

One of the most important issues when designing applications for road safety is that users keep the focus on driving, because this is a major cause of deaths on the road. For this reason we propose VAiPho Watchdog, which is a very light application that listens to the registry and when the mobile phone connects with the car hands free, it automatically starts the VAiPho so it does not need any driver attention to connect it avoiding distraction when users forget to turn it up. VAiPho Watchdog starts when the user powers on the mobile phone and it runs in background. It hardly consumes resources because its unique function is to listen to a registry and to start *VAiPho Automatic* in case the Bluetooth device starts.

A possible drawback of VAiPho is that it uses multiple communication interfaces at the same time, which implies high battery consumption. We are aware that priority of mobile phones is making phone calls and VAiPho battery consumption could cause discomfort for users. In order to solve this problem *VAiPho Watchdog* checks the battery level before starting *VAiPho Automatic*. To carry out this process, we consider that the phone charge can be in 5 states: Very high, high, medium, low and very low. If the battery status reaches a specific value, it does not launch *VAiPho Automatic*. This value is set by users who can assign their preferred value among 5 possibilities. The default value is set to low.

VAiPho Automatic

VAiPho's main application is *Automatic Application*, which is the most complex because it has got many tasks to carry out. VAiPho requires both connectivity with other phones via Wi-Fi®, and GPS, so connectivity using standard interface is necessary. Besides, VAiPho uses GPS information, like lane speed, vehicle speed or coordinates in order to detect possible congestion or free parking lot. This information has to be processed, stored and sent to other vehicles. For that purpose the *Automatic Application* implements the following functions:

- Starting the wireless interface
- Creating or connecting the ad-hoc network called *VAiPho*
- Loading data from the database and password file
- Loading and starting the client beacons and the server
- Starting the GPS browser
- Starting the incident detector

When the *Automatic Application* is launched, it always follows the same process. First the mobile wireless interface is started and a new ad-hoc network called *vaipho* is created. In case it already exists, the device connects to the network. Then it creates and fills the database with user data like private and public key pairs, secret key, pseudonym

and etc. These are security issues that are detailed in section D.1.5. Finally, both a client and a server processes are started to receive and send beacons from/to the network. At this point, the system is ready to communicate and exchange information about events with other devices that are in the range of transmission. This entire process is automatic and transparent to the user, and just a voice message indicates that VAiPho has begun.

After setting the communication system, the GPS browser is started and an incident detector system is launched. The aims of this system are to detect some anomalous situation and to generate events in order to alert other users about that situation. This process is automatic and uses GPS software to get some useful information such as speed and location. Specifically for this work we have used the Sygic GPS navigation Software Development Kit (SDK) [195] that allows developers to add navigation features into any software solution running over mobile devices. The goal is to automatically detect congestion on the road so the SDK provides functions that retrieve actual location and speed as well as speed limit of current road. With this information, our incident detector computes the parameters and if the vehicle is travelling at an abnormally low speed, it concludes that the vehicle can be in a traffic jam on the road. Once detected the incident, the process generates an event with the road name, direction of movement and location in which the incident is located. This event is stored in the database and relayed to other vehicles. Thanks to these event detections, and through cooperation among devices, it is possible to know more about road conditions.

When a vehicle receives this information, two possible states of the vehicle are considered regarding the incident. The first is the case in which the vehicle is travelling on the same road and also detects the same event. For this state we use a data aggregation in order to avoid network overload. This is a security issue that is explained in Section D.1.5. The second state corresponds to a vehicle that does not detect the event because either it is not circulating on the same road where the congestion has been detected, or it is on the same road but at a large distance of the incident. In this case we distinguish two possible cases:

- The vehicle has not such a road as part of its route: In this case, the vehicle acts as a simple packet transmitter and it relays the packet to all vehicles that are in its range

of transmission.

- The vehicle has such a road as part of its route: In this case it takes more processing, on the first step is determine whether the busy road is in our route, and if so, the process must calculate if it is better to choose an alternative route or to keep the same one. If the system determines that it is better to use an alternative route, the new route has to be calculated.

In order to implement all these processes, many security issues have to be taken into account because it is easy to generate false events, or that some attackers try to alter the contents of the packets, or even they can deny relaying packets, trying to attack the network. Therefore, communications among vehicles and information about detected events relayed in the network should provide evidence of being truthful. For this purpose, a security module has been created that will be presented in section D.1.5, where we explain possible attacks and how VAiPho resists each of them.

Other important task of the *Automatic Application* is the possible parking detection. In all kind of cities and especially in big cities, there is huge problem with the lack of parking lot. This application can help to find parking spaces in real time. The procedure is simple, when a user turns on the car to leave a parking, the GPS synchronizes with the GPS signal and gets the coordinates where the vehicle is parked before it leaves the space. These coordinates are broadcast as possible free parking. Received events of parking spaces have a fast expiration, between 30 seconds and 4 minutes, what is configurable by the user. The default value is set to 1 minute. Fraud can not be controlled for parking events. There are several situations where a vehicle can leave a certain location that is not a real parking lot, such as:

- private outdoor parking space,
- a location where the driver parked in double parking,
- parking where is prohibited.

For this reason, when the tool warns about a parking, it indicates that it is a potential parking space, and there is no guarantee that this place exists or when it exists

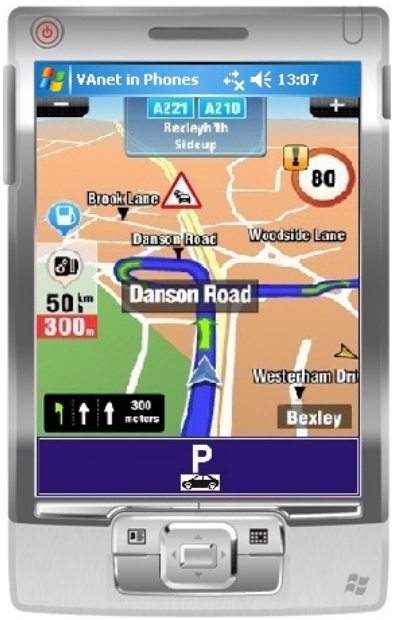


Figura D.3: Driver Interface

and that it continues free when the receiving vehicle reaches it. This parking task is started by the user. A voice message is launched when a parking is detected and a parking signal is shown on the map.

On the other hand, when the user turns off the car, the coordinates where the vehicle is parked are stored in the database in order to help the owner of the vehicle. On many occasions the users find it difficult to find the place where they left their cars parked, especially if they move in large cities, where the streets are new or unknown to them. In this regard, with the coordinates stored in the database, VAiPho will draw on the map the location where the vehicle is parked, allowing the user to calculate the route to reach its car using their GPS tool.

VAiPho User

A good user interface design can be the difference between product acceptance and rejection in the market. If end-users feel it is not easy to learn or to use a product, although it is escellent it may fail. A user interface design can mean a product either is easy

to understand and use, which results in greater user acceptance, or just the opposite.

We have designed two simple interfaces. The interface that is shown when the user is not driving Fig. D.4, where the user can configurate VAiPho's variable parameters like parking expiration, battery level, application language or confidence data load generated in the VAiPho website [207]. Additionally the user can check the events that are updated and stored in their mobile phones. *VAiPho User Application* provides a tool to locate where the vehicle is parked, this application, is one of the most interesting, but it is only valid if *VAiPho Automatic* was active when the car was parked and the location where it parked had GPS coverage. For this purpose *VAiPho User Application* provides a parking remainder button, *find*, which the users click to locate their cars. This application starts the GPS navigator on walking mode with an icon showing the place where the vehicle is located beside the user's current location. This will calculate the route on the map, from the user current location to the car, so that the users can follow it and find their vehicles. However, this parking feature is only available if *VAiPho Automatic* was active when the car was parked and the parking had GPS coverage. In this case, the user message is displayed indicating that this information is not available.

The interface displayed when the user is driving is very similar to that used by conventional GPS. This interface, to avoid driver distraction, not only uses the icons on the maps but also voice messages. Therefore, when it detects a traffic jam, an icon on the screen is shown and a voice message indicating congestion on the route is heard. The same method is used to show possible parking lot when the user reaches the destination, at this moment the user presses the parking finder button that is show in Fig. D.1.3.

D.1.4. VAiPho Website

The website [207], besides advertising VAiPho and allowing users to download the tool, is vital for operating profit. This page has a menu from which we highlight points of interest such as downloads and updates, where users can get VAiPho for different platforms and updates them to any new features that are developed. In the user menu, people interested in VAiPho must register. This allows them to generate a file of signatures and certificates that are required to use this tool. In addition, we provide a support service

Figura D.4: Pedestrian Interfaces withouth and with vehicle location saved

where users who install the application can post any questions about its operation. Finally, in the *Clients* section, companies or users who are interested in advertising through this tool may contact the administrators to provide geolocation advertising.

Certificates

VAiPho is a self-organized tool based on the confidence of users who use a trust model where device signs each other expressing its confidence. The more signatures a device has, the greater the trust from other nodes that can be used during its life in the network. Therefore, we can find a similarity between our network and social networking sites like Facebook, Twitter, etc. In this way, VAiPho operates according to what is called viral marketing, that is to say, it uses techniques based on pre-existing social networks.

With this in mind, when a user logs in the web to get its pair of public and private keys in order to start using VAiPho, it will ask for information about their contacts like a social network. The tool currently can import contacts from a mail account. In this way, VAiPho searches for other users also registered and friends of this new user. This list

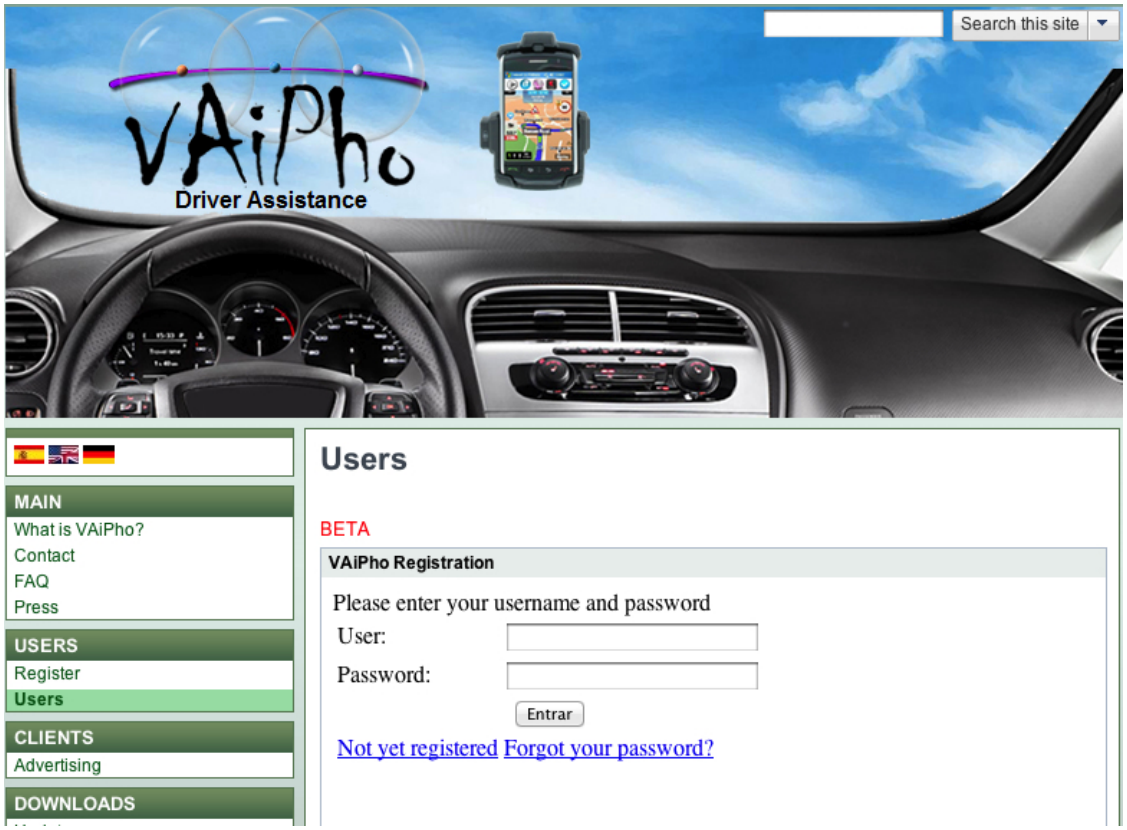


Figura D.5: Website - User registration in Webpage

will be displayed to the new user so he/she can confirm which these friends are trusted so as to belong to the network VAiPho. With the user's private key, the website will sign the public keys of its friends when both of them have been mutually authenticated. These signatures are called certificates. In the process of signing, *VAiPho Website* exchanges the generated certificates so that each user holds the signature generated by itself and the signatures produced by its friends. Once a mutual authentication has been made, the user may download a file containing the certificates generated by itself and its friends, what is needed to generate the certificate graph of the trust network. All this process as well as where the file has to be stored is detailed on The VAiPho Website in the User menu.

Advertising

All registered users or companies can contract geolocation advertisement. To do this, VAiPho Website has a section *Clients*. The advertising data include Name of the company, Message, X coordinate, Y coordinate, area of interest, expiration date and the commerce logo. The message is limited to a short number of characters to be easily showed and read by the application.

D.1.5. Security Analysis

Security of communications in VANET represents an important challenge to be solved because it is expected that in the future these networks will imply a revolution for safety and comfort of road transport. In these networks, warning messages affect driver decisions who will reduce speed and/or choose alternative routes based on the received information, so a scheme is necessary to determine if the available road traffic information to the driver is trustful or not. Besides, the quality of communications in VANETs is degraded when the number of non cooperative vehicles is very large. For this reason, in VANETs the security, privacy and safety of users are the main goal, hence, VAiPho integrates the following security mechanisms in order to avoid possible attacks.

Also, VAiPho is a network formed by the limited wireless conections allowed by nodes that are inside vehicles. For this reason, VAiPho can not addressing. VAiPho will be formed by multiple sub-networks that continually change, and the nodes that make up

these subnets exchange information rapidly. This network topology does not ensure that information reaches all points, but at least the most populous. Besides this type of network has no limit on the number of nodes and distance support, since it can always make more subnetworks.

User Security

One of the most important issues in security is anonymity. Often it is undesirable that the communicating parties reveal their identities. Furthermore, an attacker could detect a particular phone signal to track somebody. In centralized system using 3G the situation could be worst because the attacker could track all the vehicles through their mobile phone numbers from a single site. This is a matter of personal freedom, so the identities of users should never be revealed in order to prevent tracking. VAiPho mechanism never uses 3G so this attack is impossible. However, the particular phone signal can be track. To provide user anonymity we propose the use of variable pseudonyms as identifiers.

Each node changes its pseudonym in random time periods and warn other users with which it is authenticated by the beacons.

Another important security issue is to ensure that the device corresponds to a legitimate user on the VAiPho network. To check this in a fully distributed network, where there is no central infrastructure to control identities, is very complicated. In this work, a mechanism to verify the authenticity of users has been designed and implemented. The authentication mechanism is based on a zero-knowledge proof, which is an interactive method for one user to prove to another that it knows a secret, without revealing anything about it. The secret is the public key of a node that both users know. The public key is specially selected to fulfill the requirements of the used zero-knowledge proof. It is necessary that both users have in common a know node to make this possible so that they maintain update their repositories in every authentication protocol, in their own repositories, the certificate of nodes with the highest degree. According to the so-called rule of 6 degrees of separation [213], nodes in this type of network have in their local repository at least one node in common with a high probability. Furthermore, this mechanism automatically devaluates the user certificates that repeatedly show bad behavior, leaving them outside and revoked

on the network in case of repeated misbehavior. The information about revoked users is also exchanged after the authentication process. Authenticated nodes de-authenticate other nodes when the time-span expires without contact with these nodes.

Information Security

On the one hand, an attacker could simulate a non-existent traffic jam in order to convince other users not to choose a route and in this way the attacker can have the road free of cars. For this reason, VAiPho uses a data aggregation mechanism to avoid this type of possible fraud. The first phone detecting an abnormal situation such as a speed much lower than expected for a long time, sends a traffic jam warning to its neighboring phones. If neighbor phones detect that their speed is abnormal too, they sign the received information corroborating it. When the promoter phone receives a minimum number of signatures, it adds them to a package with the information and sends it to all neighbors who will disseminate the message.

Thus, traffic jams must be detected by different vehicles, which must sign the traffic event with their private signature in order to aggregate all of them in a single packet. Therefore, we ensure that not only a vehicle has detected an incident but also several can corroborate it. This mechanism eliminates the possibility of spreading false traffic jams created by a single attacker and also avoids possible confusion generated by the system when a vehicle stops on the side of the road due to a flat tire or because the vehicle has been broken.

The number of required signatures depends on the expansion of the tool, that is to say, the higher the number of vehicles with VAiPho, the greater the number of required signatures. This will mean that the larger the number of vehicles incorporating VAiPho, the lower the possibility of attack. In order to calculate the number of required signatures, VAiPho checks the time from its first authentication with other user in the current journey and calculates the average number of users per minute. In the current implementation, this average is lower than one per minute, the number of required signatures is two. If the number is between one and four per minute, the number of required signatures is four. If the average is higher than four per minute, the number of required signatures is five.

Network Security

Nowadays, wireless connectivity is over WLAN but in the future it is expected that will be over direct Wi-Fi or by using the IEEE 802.11p standar. The packets are exchanged among vehicles using others as relaying nodes. An attacker could try that communications fails, what could cause a VANET to be broken into pieces so that the network can not provide services such as packet forwarding. In this sense, these attackers would cause a passive denial-of-service with the goal that the wireless network does not work properly. We must ensure that only those vehicles that belong to the network and help in its operation, benefit from information relayed in it. VAiPho prevents users who want to benefit from the VANET, without helping in forwarding the information because such a passive attack would degrade the functionality of tool and compromise the connectivity of the network.

VAiPho proposes a specific mechanism against these attacks. We propose using encrypted exchange of data as a method to strengthen cooperation in relaying packets. In particular, we describe a scheme to produce a secret key to encrypt sent information. This procedure prevents passive nodes that do not cooperate in relaying packets to get benefit from this information, what improves efficiency and security of communication in VANETs.

D.1.6. Simulation and Implementation

In this section we present the simulation and implementation settings and demonstrate that VAiPho is successful in accomplishing the goals of reducing CO2 emissions by avoiding traffic jams and helping people to find parking lot. All this is accomplished in a safe way, both regarding received information and user privacy.

The main goal of these proofs was to evaluate whether developing VANETs through mobile phones could fulfill the specific characteristics of VANETs such as their hybrid architecture, high mobility, dynamic topology, scalability problems, and intermittent and unpredictable communications. The first test was checking that communications between smartphones with WiFi by using the IEEE 802.11b/g standard were feasible using a simple client-server application between devices, when circulating with vehicles in urban environments or motorways at different speeds by using different number of devices. These tests

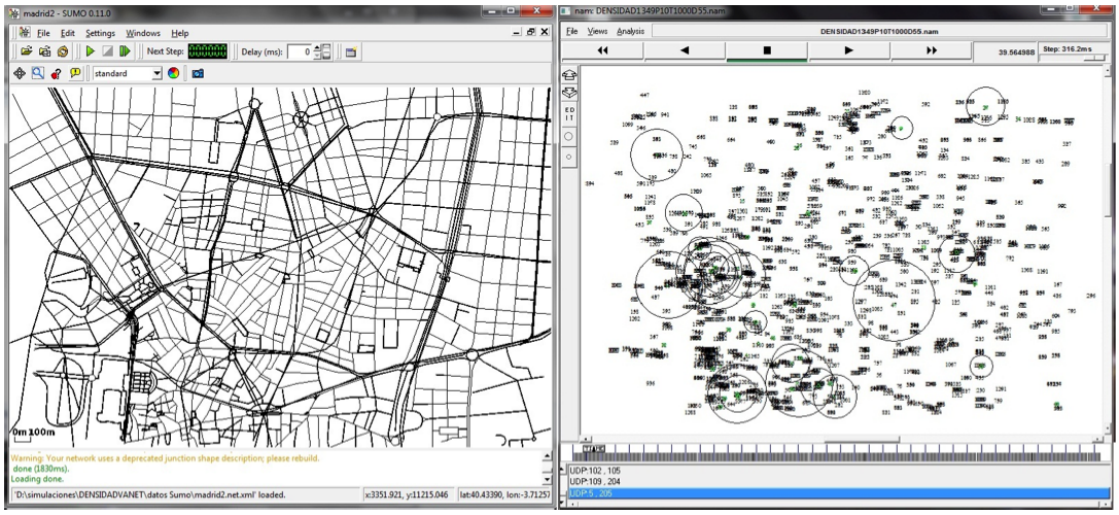


Figura D.6: VAiPho simulation with SUMO and NS-2 in Madrid

were successfull. After this, we create several simulations by using SUMO and NS-2 and then we implemented the complete VAiPho application for smartphones.

Simulation

Both the feasibility and effectiveness of VAiPho approach are shown through the figures where a simulation exemplifies its performance. In the first part of our demonstration D.6, a NS-2 [159] and SUMO [194] display shows the VANET state in one moment when VAiPho is operating. The most relevant options selected for the demonstration have been: Total number of vehicles: 600 - 15000, number of vehicles with OBUs: 1 %-100 % , simulation time: 100-216000 seconds, authentication period: 20 seconds, distance relay nodes: 75 meters.

Implemented simulations let us to know the number of connections generated in the network depending on the percentage of vehicles with OBUs,from here it can be extracted which is the necessary minimum number of vehicles with OBUs for the network starts to exchange information seamlessly.

On the other hand, it lets us to know from what percentage of vehicles with OBU begins to drop the quality of communications because of interferences in the channel. These

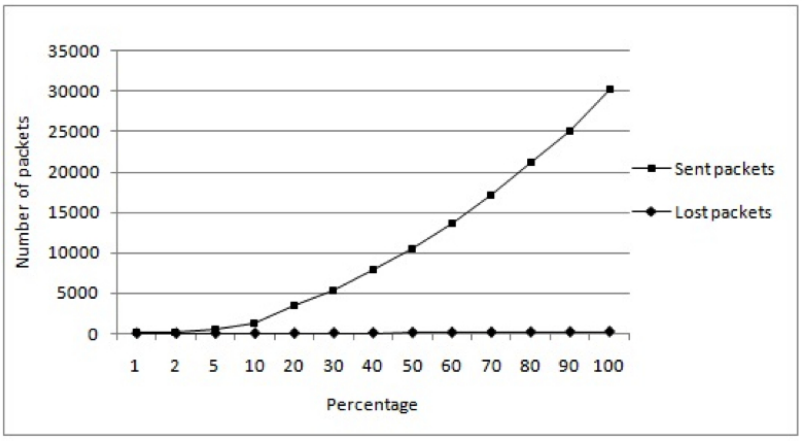


Figura D.7: Number of packets generated and lost in a VAiPho simulation

simulations do not have into account interferences from other devices or others WLAN networks that could affect wireless communications. It is possible that this threshold is reduced in real environments.

Implementation in Smartphones

For the evaluation, we built VAiPho application in different smartphones with Windows Mobile using C#.NET. A video about how this tool works can be find in the WebSite [207].

The tool has shown to be effective with traffic jams recognition, parking detection and finding parked vehicles. It also authenticates other users, protects user privacy with variable pseudonyms and exchanges secret information with other devices. The devices running VAiPho that receive information, display and forward such information correctly.

In particular tests showed good performance in:

- Simulating a traffic jam where a device automatically detects it and sends a warning message, and another device on the same route and direction detects the same traffic jam and signs the received message. The signed message is sent to a third device that displays the traffic jam and sends it to the last device, which also warns indicates the driver about the situation.

- Parking detection, so that a vehicle leaves a parking space and sends this information to the other vehicles and, other vehicles receiving this announcement forward the information.

- Finding a parked vehicle so that a vehicle was parked in a parking space and the user went away from the vehicle. Then the user presses the parking reminder and the tool showed the route on foot to the vehicle.

- Geolocation advertising, which is broadcasted in its area of validity.

D.2. Merging Sub-networks

The most widespread wireless technology for mobile ad-hoc networks nowadays is Wi-Fi technology based on IEEE 802.11xx protocol. The structure of this protocol produces some problems when sub-networks try to merge. Two different reasons that can cause inability to communicate are IPs duplication and the existence of sub-networks on different channels. In the bibliography some theoretical solutions have been mentioned and tested through simulation, but not with real devices. Here a practical solution is fully described and implemented in a new tool developed to create a vehicular ad-hoc network by using only mobile devices. For the choice of the sub-networks to merge, first a simple and deterministic algorithm is proposed, and large scale simulations based on data obtained from implementations with real devices are performed using NS2. Then, interferences between channels are taken into account under a fuzzy logic point of view.

There are many papers that try to solve problems in channels like in [211], where authors try to use multiple channels simultaneously or in [5] which shows how to estimate the noise in the channel and to avoid it by using fuzzy logic and the Vitervi algorithm or in [129] which analyzed the impact of channel estimation errors due to noise and interference on multiuser detection (MUD) under the framework of the replica method. Authors in [217] propose a cooperation efficiency of the multiple-relay channel and heuristic polynomial time algorithms to eliminate interference when carrier-level synchronization is not available and all nodes use a decode-forward scheme. Reference [215] proposes a framework for multi-radio

multi-channel cognitive wireless networks. Under this framework, data routing, resource allocation, and scheduling are jointly designed to maximize a network utility function.

Other papers try to solve different problems in the IEEE 802.11 protocol like [51], where they propose an application that using Bluetooth technology allow to solve the configuration problem of the terminals conforming an IEEE 802.11-based ad-hoc network by extending the connectivity range of nodes through packet forwarding, thereby avoiding the use of a fixed infrastructure. In [124] authors do a performance of the 802.11b protocol in such configurations using the analytical modelling technique PEPA (Performance Evaluation Process Algebra). Reference [137] proposes a protection for DoS attacks in 802.11 protocol for WEP, WPA and WPA2 by solving a problem with the control frames. Finally, [158] proposes and simulates the (ESPRIT) Estimation of Signal Parameters using Rotational Invariance Techniques algorithm in order to mitigate the Bluetooth (BT) interference with the channel estimation stage in IEEE 802.11g.

All these works try solve different problems of the IEEE 802.11 protocol but any of them is near to the problem here proposed. The nearer work to the proposed here is [198] where in multihop ad hoc networks that use conventional IEEE 802.11, long transient resynchronisation states are often generated when multiple IBSSs merge. They propose a modification of resynchronisation that permit times much shorter and reduces energy consumption. But this work does not perform implementations in real devices.

D.2.1. Statement of the Problem

The solution implies that the sub-networks have to be on the same channel and with no IP conflict to merge with no problem. However, this case is unusual spontaneously. The most usual problem is the first one because the choice of the channel is random. Moreover, the channel with the lowest number uses a lower frequency, and with a lower frequency the propagation is better because there is less attenuation. However, the problem of the difference in attenuation is not comparable with the loss of packets in the channel due to the interference between networks in adjacent channels. There are channels that are more saturated than others. That is the case of channels 6 and 11. Also interferences exist with Bluetooth and other systems using the 2.4 GHz ISM (Industrial, Scientific and Medical)

band. The broadcasting channels that are more common in the IEEE 802.11xx protocol are type a, b, g and n. Its range is from channels 1 to 11 in America, 1 to 13 in Europe and 1 to 14 in Japan. Channels share part of its bandwidth.

Figura D.8: Channels in Europe

Fig. D.8 shows the channels in Europe that devices can choose to create the wireless network and to allow the use of the maximum number of channels without interference. In particular five is the maximum number of wireless networks without interference from neighbouring networks.

The various branches of the IEEE 802.11xx protocol provide the advantage of being compatible with each other, so that the user does not need anything more than its integrated Wi-Fi adapter to connect to the network. Too much saturation in one channel causes packet collisions in the data transfer, what corresponds to a lower transfer rate and/or speed.

A solution is here proposed for the merging problem when two instances of the same network are in different channels. The ideal solution implies that the smallest sub-networks merge with the largest one, but nodes do not know the size of other sub-networks. This work describes an algorithm to compute the time that devices of a sub-network takes to reset and connect with other sub-network, depending on the size of its sub-network. Fuzzy logic rules have been used to use the interference between channels that affects this time.

D.2.2. Deterministic Proposal

If a node detects an IP conflict, it resets its network interface. This is enough to solve the problem that would exist when the corresponding sub-networks in the same channel merge.

To solve the merging problem, this paper proposes to reset the wireless interface of the devices that belong to one of the wireless sub-networks. The optimal way for resetting

sub-network could be by adjusting those sub-networks that have a lower number of devices, but this information is not transparent to the devices of each sub-network, which can only know how many devices are in its sub-network, and not in other sub-networks.

Basic Proposal

In a first approximation, by using only the channel interference, the nodes of the sub-networks with more interference possibilities are restarted. Then, the first node detecting this situation and belonging to one of those subnetworks sends a warning message to the remaining nodes of the network so that they have to restart its wireless interface. Thus, upon receipt, the nodes after broadcast the message, turn off their network interface for 1 second and later reactivate it again. Then, since a *vaipho* network already exists when they restart, the nodes will connect to this network and the problem will be solved. The nodes will remain authenticated with the nodes of the previous sub-network because authentication data do not vary with the change of used sub-network, which has exactly the same name, *vaipho*.

Algorithm Choosing the Subnetwork with Less Interference

```
00: ...
01: //There is networks != vaipho, what has less interference
02: int[] channelsDifference = new int[nNetworks];
03: //init channelsDiference to see what has less difference.
04: for (int i = 0; i < VaiphoInstances; i++)
05:   channelsDifference[i] = 50;
06: for (int i = 0; i < nNetworks; i++)
07:   for (int j = 0; j < nNetworks; j++)
08:     if (network[i].Equals(vaipho))//compare vaipho with others
09:       if ((i != j) &&
09:       (Abs(channels[i] - channels[j])) < channelsDifference[i])
10:         channelsDifference[i] = Abs(channels[i] - channels[j]);
11:   //diference of channels to vaipho networks
11:   //the biggest diference is ok, others have to reset
```

```
12:  for (int i = 0; i < nNetworks; i++)
13:    if ((biggestDifference < channelsDifference[i])
13:    &&(channelsDifference[i]!=50)) //save the biggest difference
14:      biggestDifference = channelsDifference[i];
15:      nodePositionBiggestDifference = i;
16:    if (nodePositionBiggestDifference != numberOwnNetwork)
16:    //if the network has not the biggest interference
17:      networkDetachProtocol(vaipho);
18:      return true;
19: ...
```

Optimized Choice

Devices can know exactly the number of nodes that are in their sub-network, the channel where they are, in which channels there are other instances of the *vaipho* network, and the channels that are being used by other networks that can interfere with the *vaipho* sub-network.

The most appropriate sub-network depends mainly on the number of nodes in such a sub-network, but also on the interferences between channels used by other networks. If the number of devices matches approximately, the interference in channels will determine the sub-network that will be restarted.

Fig. D.9 shows a typical example that can happen in a VANET where cars with on board units inside are being driven through urban scenarios. In this example two subgroups A and B running on the same road, come into the same range of emission but two sub-networks are formed in different channels. For this reason devices of one sub-network can not see devices of the other one. In addition, there are interferences from other wireless networks. Each node of a sub-network that can see nodes of other sub-networks checks the channel where its sub-network is and the channels of the remaining sub-networks, and analyze possible interferences with other wireless networks. Fig. D.9 shows an example of the above idea where group A is on channel 1, group B is on channel 6, and there are other two networks on channels 2 and 4. Therefore, group A is on channel 1, which is at distance

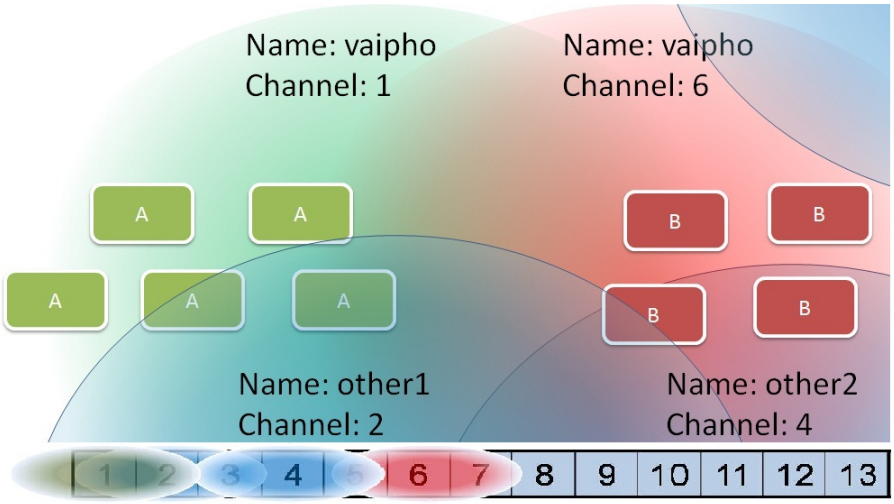


Figura D.9: Example of a Normal Situation

Tabla D.1: Mobile Devices

Model	Platform	CPU Speed	RAM	ROM	Battery Capacity
HTC HD Mini	WM 6.5	600 MHz	384 MB	512 MB	1200 mAh
HTC P3300	WM 5.0	201 MHz	64 MB	128 MB	1250 mAh
HTC Touch2	WM 6.5	528 MHz	256 MB	512 MB	1100 mAh
hp IPAQ 614C	WM 6.0	520 MHz	128 MB	256 MB	1590 mAh

1 from the nearest network (channel 2), and group B on channel 6 is at distance 2 from the nearest network (channel 4). Thus, it is concluded that the group A has more interference possibilities.

However, as aforementioned, the optimal solution would consist in resetting those sub-networks with fewer devices. In the next section an implementation analysis of the above proposal is included, and in the subsequent section a different proposal based on fuzzy logic is outlined as closer to the optimal solution.

D.2.3. Real-Device Implementation

The implementation in real devices has been developed by using mobile phones with Windows Mobile 5 and 6. The used platform was Microsoft Visual Studio 2008.

Algorithm Sub-Networks Detection

```
00: function ChekingSubnetworks (...)
01:    ...
02:   foreach (AccessPoint access in NearbyAccessPoints)
03:      network[nNetworks] = access.Name.ToString();
04:      channels[nNetworks] = access.Channel;
05:      MacDirection[nNetworks] = access.PhysicalAddress.ToString();
06:      if (network[nNetworks].Equals(vaipho))
07:         VaiphoInstances++;
08:         //save the nº of channel with the biggest network
09:         if (channels[nNetworks] > biggestChannel)
10:            biggestChannel = channels[nNetworks];
11:         //save the network
12:         if (MacDirection[nNetworks].ToString().Equals(MacOwnDirection))
13:            numberOwnNetwork = nNetworks;
14:         nNetworks++;
15: ...
```

Fig. D.10 shows a screenshot of two emulators that use the same *vaipho* sub-network while there is another *vaipho* sub-network in another channel. In this example the device on the right is the first one to detect the presence of multiple instances of the network *vaipho*. After checking that there are other *vaipho* instances, this node starts the merging protocol. In the merging protocol, the node who detects the *vaipho* sub-networks sends the warning message to the remaining nodes of the sub-network in order to restart its network interfaces. After that, every node turns off its interface for 1 second when it reactivates its network interface, and connects to the *vaipho* network that already exists.

D.2.4. Performance Analysis

Fig. D.11 shows several wireless networks in the transmission range. In particular, it can be seen that there are two instances of the *vaipho* network, the first one is on channel

Figura D.10: Real Device Implementation

1 while the second one is on channel 11. It can be also seen that there are other networks on channels 1 and 11. Such a situation implies the need of sub-network merging.

Figura D.11: Vaipho Network Instances

In order to check the average time that it would take any sub-network to integrate into another sub-network, several real executions were performed. In these cases an existing network is merged with another and to achieve it, always the first node detecting that there are two instances of the same network, begins to relay information to other nodes of its

sub-network to restart their network interfaces and in this way, they reconnect to a unique *vaipho* network.

Different implementations were carried out for different sub-network sizes. The time that devices take to restart their network interface and connect to the other instance of the network on the correct channel was obtained from real devices, by using the average. The times ranging from 2.94 seconds were recorded as the best case for the HTC HD Mini, and 9.15 seconds was recorded as the worst case for the HTC P3300. Therefore, the implementation produced a range between 2.94 and 9.15 seconds to reset the network interface. The times that depend on the used authentication protocol were not included. Thus, the considered time only includes the time it takes to shut down the network interface plus 1 second for the waiting, and the time to reconnect the device to the existing *vaipho* network. After such executions with real devices, the obtained data were used for a large scale NS2 simulation.

Fig. D.12 shows the simulation of a highway with three lanes. In this simulation the two sub-networks have different colours and in red colour we have the node that detects the two instances of the same network and sends the warning message to reset the network interface.

Fig. D.13 shows the obtained results from the NS2 simulations. These simulations were performed with sub-networks of sizes between 10 and 140 nodes. We obtained the average times that nodes take to reconnect to the network with less interference. Different levels were distinguished depending on the size of the sub-network. In such a figure, we use 'level' to refer to the subset of nodes that are in same transmission range of the transmitter node, so the first node detecting that there are multiple instances of *vaipho* network, sends a warning message, when the warning reaches the subset of nodes that are closest to it, they broadcast this information and then, restart their network interfaces.

Nodes that reach this information will be from another level, and so on. Fig. D.13 shows the average time it takes for the nodes in each transmission level to reconnect to the other *vaipho* sub-network. The simulations show an obvious result, the larger the size of the sub-network, the longer it takes to reconnect. However, if nodes are divided into transmission levels, it can be seen that it takes about the same time from the source node

Figura D.12: Simulation of the Proposal

regardless of the number of nodes that the sub-network have.

D.2.5. Proposal Based on Fuzzy Logic

The number of generated and lost packets D.14 will be always less if the sub-network whose devices restart its network interface is the sub-network that has fewer nodes. From this premise, we tried to find a method that allows, without knowing the number of nodes in the other sub-networks, to do that the sub-network with fewer devices is the sub-network whose nodes restart their wireless interfaces, in addition to this, another important factor that must be taken into account is the interference with the channels of other existing sub-networks. A correction factor based on the interference between channels is computed and added to the time that each *vaipho* sub-network waits to begin the merging process. It is important to minimize the time that nodes of a sub-network require to join another sub-network while ensuring that nodes belonging to the selected sub-network do not restart

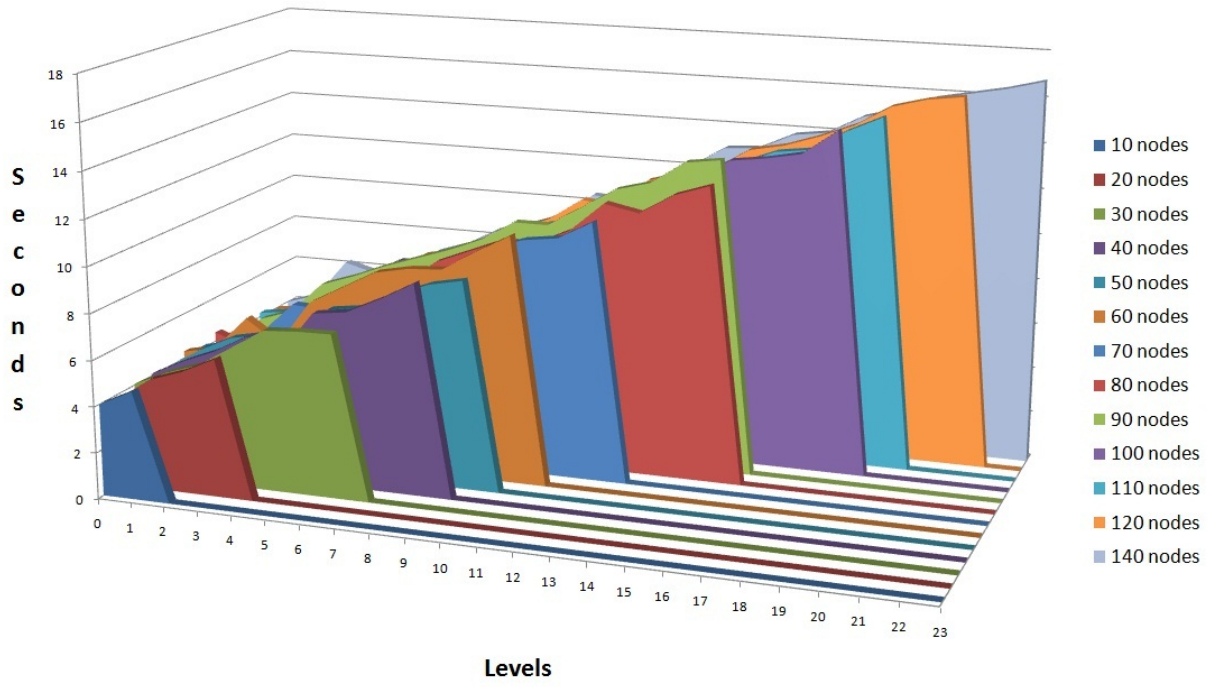


Figura D.13: Delay Merging SubNetworks

because that would imply that all nodes would restart their interfaces, and in that case the process would not be optimal.

The time has been calculated by using real values that nodes take for merging in another sub-network (see D.15). The time that each node takes to join another sub-network depends on the number of nodes in its sub-network, which was computed by using real devices. In particular, such a time can be linearly approximated by the following formula, where S_r is the residual standard deviation, which measures the average variability of the data about the regression line and x denotes the number of nodes in the network:

$$W(x) = 0.107x + 3.63 \pm S_r \; if \; x > 0 \tag{D.1}$$

From this formula, it is possible to estimate the optimum time before repeating the check procedure about whether there are multiple *vaipho* instances.

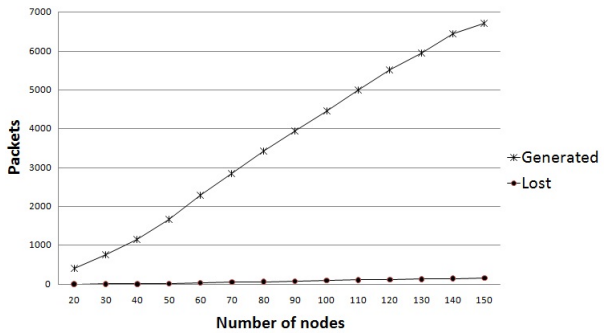


Figura D.14: Number of Generated and Lost Packets

This work proposes the use of such a linear expression by removing the initial constant time 3.63 s as starting point for the fuzzification. In this way, a node detecting that there are multiple instances of the *vaipho* network will wait for a time determined by the number of nodes in its sub-network characterized by the formula $W(x) = 0.107x$, where x is the number of nodes in the sub-network.

Therefore, a node that detects multiple instances of the *vaipho* network where its sub-network is composed of two nodes, will wait for 0.214 seconds before rechecking whether there are more than one *vaipho* instance. Another node detecting the same but belonging to a sub-network of 20 nodes will wait for 2.14 seconds before rechecking. In this way, nodes give time enough for the smaller sub-networks to join the largest sub-network.

Apart from taking into account the number of nodes in the network in case of sub-networks with similar size, we can consider the interference between channels so that the sub-network that is in a location with less interference will prevail over the others.

A sub-network operating on a channel m has no interference if no other sub-network is operating on the same channel or in less than two channels away from it.

The channel interference is measured in dBm, which is a measurement unit used in telecommunications to express the absolute power of a network signal with a power P using the channel c in a moment t through a logarithmic relationship, so that it takes negative values.

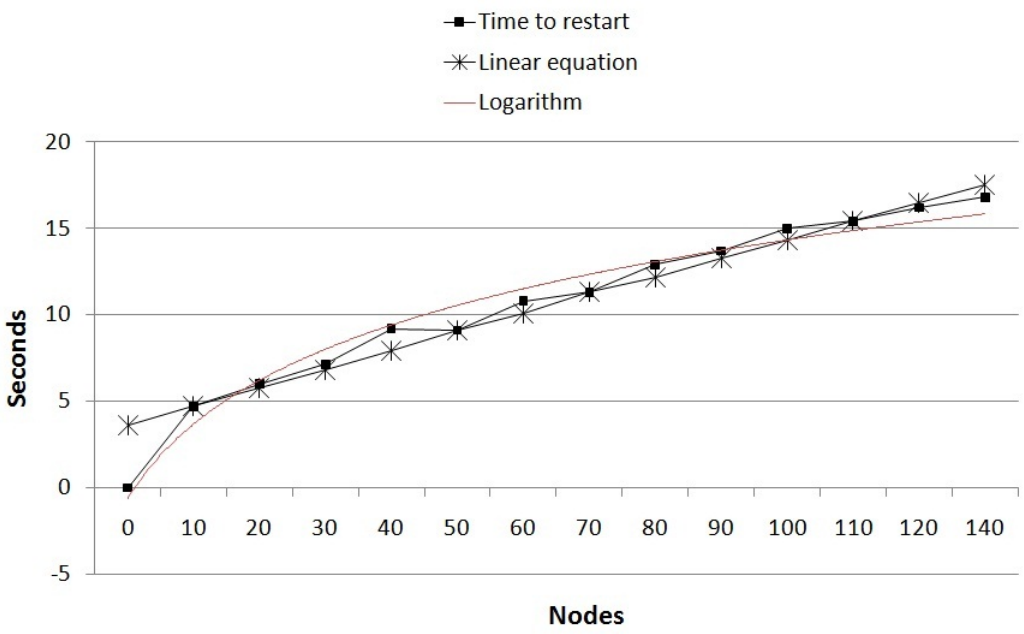


Figura D.15: Time Relay for the Merging Sub-Network

$$dBm(c(t)) = 10 * log\frac{P}{1mW} \tag{D.2}$$

Using this metric, the interference that a channel has, is calculated from the sum of the channels on the same channel as the source channel, plus the sum of the networks on adjacent channels that are multiplied by $\frac{2}{3}$, plus the sum of networks that are within two channels of distance multiplied by $\frac{1}{3}$. This calculation is because if two networks operate in the same channel, the overlap is total. If they are in adjacent channels, the overlap is $\frac{2}{3}$ of the frequency they use. And if they are two channels away, the overlap is $\frac{1}{3}$ of the frequency they use.

By using this metric, the interference that a network using a channel m has can be estimated through the following expression where N denotes the number of sub-networks using the same channel c (here denoted c_i), M and Q denote respectively the numbers of sub-networks within 1 or 2 channels of distance from channel c (here denoted c_j and c_k, respectively), and a takes is obtained from the expression $\frac{2*(-a)}{dBc_{high}} = 1$ where dBc_{high} corresponds to the channel with high interference.

$$dBm_c(t) \approx \sum_{i=1}^{N} \frac{-a}{dBm(c_i(t))} + \sum_{j=1}^{M} \frac{2 * (-a)}{3 * dBm(c_j(t))} + \sum_{k=1}^{Q} \frac{-a}{3 * dBm(c_k(t))} \tag{D.3}$$

This expression results from the fact that if two sub-networks operate on the same channel, the overlap is total, if they are in adjacent channels, the overlap is $\frac{2}{3}$ of the frequency they use, and if they are two channels away, the overlap is $\frac{1}{3}$ of the frequency they use. Note that the value of a could vary depending on the place where vehicles are because if they are in a place where there are many wireless networks, the possibility of having more than two sub-networks using the same channel is higher.

At this point we propose the use of a fuzzy logic based approach for the application of the above interference between channels of sub-networks to estimate waiting times for resetting.

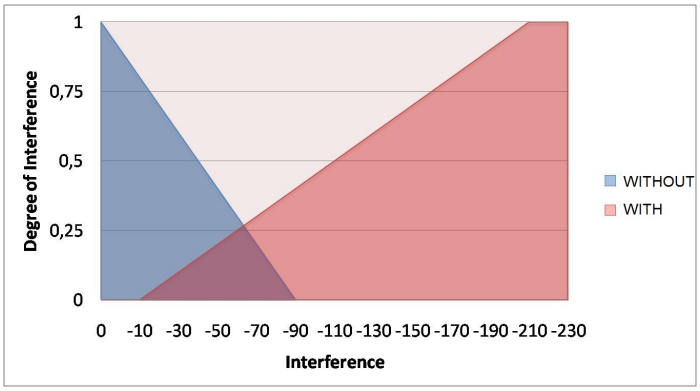


Figura D.16: Fuzzification Function for Interference

The equation that defines the interference degree y for the current *vaipho* sub-network, obtained from its interference value x, is given by the equation D.4

$$y = \begin{cases} \not\exists & si \quad x > 0 \\ 1 + 0.0\hat{1}x & si \quad x \in (0, -90) \\ 1 & si \quad x < -90 \end{cases} \tag{D.4}$$

On the other hand, the equation that defines the value of interference produced by other *vaipho* sub-networks with interference value x, is given by the equation D.5

$$y = \begin{cases} \nexists & si & x > 0 \\ 0 & si & x \in (0, -10) \\ -0,005(x+10) & si & x \in (-10, -230) \\ 1 & si & x < -210 \end{cases} \tag{D.5}$$

This method allows to consider the degree of interference found between two different instances of the *vaipho* network to estimate the waiting time for the reconnection of one or both sub-networks. Thus, Fig. D.16 shows an estimation of such a degree of interference depending both on the power of the network and the existence of interferences with other networks.

In the design of the rules of fuzzy control, denoted below by Rule 1 and Rule 2, DI is the Degree of Interference between the two analyzed sub-networks, the network of the node performing the test is denoted by $vaipho(\alpha)$, the other sub-network is denoted by $vaipho(\beta(l))$ where $l \geq 1$ and correspond with the other instances of *vaipho* network. $F(x)$ is used to store the time that the x nodes of the sub-network $vaipho(\alpha)$ wait before checking again if the other *vaipho* instance exists. In case the other *vaipho* instance already exists after the waiting time, the node which detected some *vaipho* instances at first, starts the merging protocol, in other case, the node does not start the merging protocol and continue normally.

Rule 2 is always tested after Rule 1, and they determine the time that is added or subtracted to the time depending on the number of nodes.

Rule 1 Own instance of the *vaipho* network

---

if $(DI(vaipho(\alpha))$ is $WITHOUT)$ **then**

$F(x) = W(x) + \frac{DI(vaipho(\alpha))}{2}$

if not

$F(x) = W(x) - \frac{DI(vaipho(\alpha))}{2}$

Rule 2 Other instance of the *vaipho* network

---

if $(DI(vaipho(\beta(l)))$ is $WITHOUT)$ **then**

$F(x) = F(x) - \frac{DI(vaipho(\beta(l)))}{2}$

if not

$F(x) = F(x) + \frac{DI(vaipho(\beta(l)))}{2}$

According to the above proposal, each sub-network $vaipho(\alpha)$ with x nodes will be able to estimate its own waiting time $F(x)$ having into account not only its own data but also another sub-network $vaipho(\beta(l))$' data such as its channel and the remaining channels that affect it.

Fig. D.17 shows the waiting time that a node has to wait before recheck the *vaipho* interfaces. This time depends on the number of nodes of the sub-network and it is affected by the interference in channel where $vaipho(\alpha)$ and $vaipho(\beta(l))$ are.

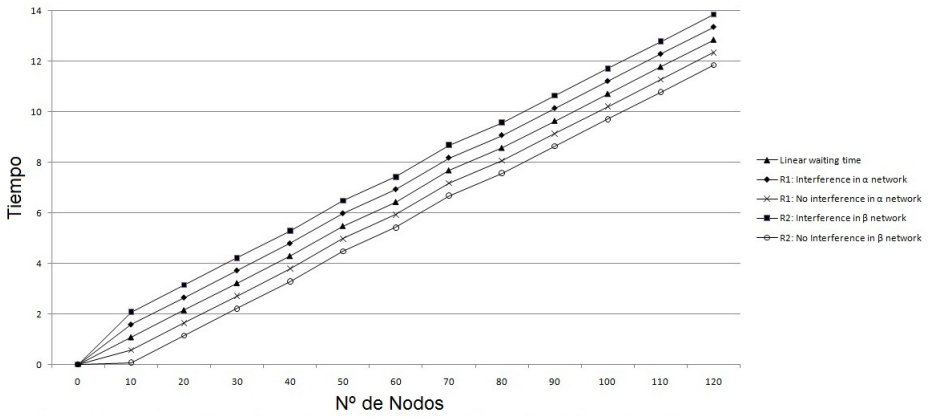


Figura D.17: Waiting Time Depending on the Interference

D.2.6. Security Analysis

It can be distinguished two types of attacks on wireless networks, the attacks produced by nodes that have not been authenticated, therefore they are outside attacks, and those produced by authenticated nodes that are into the network, therefore, inside attacks. In this paper the authentication protocol used is transparent. Therefore, it is assumed in advance that the authentication protocol prevents inside attacks. The topology of the VANET or MANET can not ensure the absolute security because the channel is open to

everyone. There are several types of attacks that can perform unauthenticated nodes by using the vulnerabilities in a distributed and self-managed wireless network.

The attacks carried out against the IEEE 802.11 protocol can be classified into passive and active attacks. Among the most important can be classified Sniffing and passive traffic analysis and in active: impersonation, re-enactment, amendment and Denied of Service (DoS). Most of both passive an active attacks can be avoid with a good authentication protocol, but there are some possible attacks to the proposed schema.

DoS attacks can be performed by any malicious user who can make noise in the channel and in this way, devices can not communicate with each other in these range of emission. The only possible solution to this attack is to change the channel and therefore, the frequency of broadcast, however, a new channel would have the same vulnerabilities as the previous one.

The scheme proposed in this paper has a big drawback. A malicious user could create an instance of *vaipho* in a different channel from the original sub-network *vaipho* and do not reset its interface.With this, devices in the real network would restart its wireless interfaces and would connect to the false network created, then, the malicious user changed again when it becomes to reconnect, if the malicious user do this continuously, the network would be reconnecting all the time.

To partially prevent this attack, it is define a timestamp where nodes can not reconnect to other sub-networks, instead of this, the device has to wait 6 minutes before reconnecting to another network. This measure does not solve the problem completely, but the network will run normally since the events that occur in the VANET are stored in the databases of devices until they expire, and these events can be exchanged and relayed in this range of time. So the attacker's goal will not bear fruit.

Appendix E

Conclusions and Future Work

Due to their wireless decentralized and mobile nature, Mobile Ad-hoc NETworks (MANETs) nowadays represent one of the most promising research areas in the field of communications. However, in order to deploy these networks successfully it is necessary to address several security threats. Particularly important is the resolution of the problem on the authenticity of nodes and of cryptographic keys, which is more difficult to solve in these networks than in traditional wired networks and wireless networks with access points.

A special type of mobile ad-hoc networks is that formed by vehicles, called Vehicular Ad-hoc NETworks (VANETs), whose future applications will imply great benefits for road safety, economics, environment and life quality of users. During the research on these networks, in addition to the security issues, we have the problem of the lack of actual deployed VANETs.

This thesis has proposed several solutions to the aforementioned problems in MANETs and in VANETs. In particular, several new protocols have been specially designed for node authentication, and for life cycle, public-key and sub-network management in such networks. Thus, this thesis contributes to the security research in ad-hoc networks. Also, a novel and practical way to develop secure VANETs with today's tools has been proposed in this work.

The main results and contributions of this thesis are summarized below.

Chapter 2 has presented with full detail the design and simulation of a novel and

complete self-organizing life cycle management scheme for MANETs called SLCM. The SLCM scheme is able to react and adapt to network topology changes without the necessity of any centralized authority. The proposal is balanced because the procedures that legitimate members of the network have to carry out when the network is upgraded (insertion or deletion of nodes) imply identical work for every legitimate node. It also has a lower cost than other methods because it minimizes the number of packets generated for all nodes. An important part for the access control of the SLCM scheme is a new node authentication protocol in MANETs, designed to run in environments without servers. This new protocol is based on the knowledge of each member of the network about a changing piece of information. Its technique is based on zero knowledge proofs, cryptographic paradigm that avoids the transfer of relevant information and that allows to define a strong authentication scheme based on the Hamiltonian Cycle Problem.

The development and evaluation of the SLCM scheme has been based on many NS-2 simulations, which are an important part of this work. The obtained results show the scalability and robustness of the proposal.

The second part of Chapter 2 has been the proposal of a new public key management scheme for MANETs, based on trust chains. In particular, two new algorithms have been proposed for updating public key repositories that improve efficiency and degree of success against the original proposal in different situations, maximizing the probability of communication between any pair of nodes, such as it has been demonstrated in a comparative analysis through Tcl and C++ implementations of the schemes.

Chapter 2 finishes with a novel lightweight mutual authentication scheme for readers and tags, which meets the EPC Gen2 standard of RFID and is immune against most known attacks on authentication schemes. The proposed scheme allows the centralized management of MANETs by adhering tags to mobile nodes and adding new special nodes corresponding to the readers to the network, in order to have total control of the variable topology, as a decentralized alternative to the SLCM scheme.

In Chapter 3 we have proposed a vehicle authentication method for self-managed and decentralized VANETs that does not require deploying any infrastructure on the road or any special equipment on vehicles. The proposed protocol is based on public/private key

pairs and certificate graphs, and a vehicle-discovering scheme with variable pseudonyms to protect node privacy and prevent vehicle tracking. Our proposal allows the use of a hybrid encryption scheme that combines secret-key and public-key cryptography, what can be used to optimize resources.

At the end of Chapter 3 the use of clusters has been proposed as a solution to decrease the number of communications in VANETs under dense traffic conditions, when the overhead of transmitted data causes a considerable drop in communication quality. In particular, a complete description of the proposed scheme for autonomic cluster management in VANETs has been provided, which includes differentiation among possible vehicle states: from the initial state when it does not belong to any cluster, to the choice of an existent cluster to join it or the creation of a new cluster. It has also been shown how to proceed with inter and intra cluster communications. Furthermore, both a cluster-head selection algorithm based on a version of the independent set problem and a secret key agreement scheme based on a generalization of Diffie-Hellman protocol have been here presented.

An analysis of all the proposals included in Chapter 3 has been done through simulations using the open source traffic simulator SUMO and the network simulator NS-2. The results show that our proposal improves the performance and security of VANETs, while ensuring the delivery of messages in real time.

Chapter 4 has included some of the most remarkable issues of a new proposal of mobile application for driver assistance, called VAiPho (VANET in Phones), which is a tool to create self-organized VANETs, mainly to avoid traffic jams, but also for other uses such as detection of free parking spaces, location of parked vehicle, or geolocated advertising. VAiPho does not require deploying any infrastructure on the road, which allows a gradual introduction of VANETs. The main objective of this part of the work has been to implement several of the designed algorithms included in the previous chapter as part of the software developed on current devices like mobile phones equipped with wireless connection and GPS for different mobile platforms. Thus, devices with VAiPho can communicate with each other by sending traffic information, after having mutually authenticated. VAiPho has been designed taking into account the security of both the information, which has to be updated and reliable, and the users, by protecting their privacy. It includes several tools to

prevent fraud and transmission of incorrect information generated in abnormal situations or produced by illegitimate nodes. We have also paid special attention to that VAiPho interfaces are simple in order to ensure that the program will not cause driver distraction. For this reason, VAiPho uses voice warning messages together with icons on the screen. VAiPho operation requires that many users make use of the tool in order to generate and relay enough information. For this reason, it is convenient that VAiPho is initially proposed as a free tool for the user, in which case, it could be sustained with geolocation advertising as a service that could be hired by companies.

The end of Chapter 4 and of the thesis has proposed the development of several practical solutions to a problem that has arised after VAiPho deployment with mobile devices in real circumstances, consisting in the need of merging several mobile wireless sub-networks formed by mobile devices equipped with Wi-Fi by using the IEEE 802.11xx protocol. The main problem arises when sub-networks are created on different channels so that some nodes in a sub-network are not visible to other nodes in another sub-network. The optimal solution would consist in the fact that the smaller sub-networks join the largest sub-network, but there is no possibility for nodes to know the number of devices that integrate other sub-networks. Thus, this work has first proposed a simple and deterministic solution based on the number of nodes in a sub-network. This thesis has also outlined a new fuzzy logic approach to estimate the time that a node has to wait before restarting its network interface, based both on the number of nodes in its sub-network and data about possible interferences from other sub-networks. The deterministic proposal for sub-network merging has been analysed through its implementation in real devices and large scale NS2 simulation, and the obtained results are promising as they show how in a few seconds, all nodes of a sub-network merge into another sub-network.

Among the open problems to be faced in the near future we can mention the study of specific applications and practical limitations of the proposed schemes for node authentication, and life cycle, public-key and sub-network management both in MANETs and in VANETs, and their large-scale implementation in real environments. Also the implementation of the fuzzy logic based proposal for sub-network merging is part of a work in progress.

Furthermore, in the future we intend to extend the current proposals to the so-called next generation networks in the known as the Internet of Things or IoT, which will offer ubiquitous connectivity services to mobile users through heterogeneous wireless networks. These will be used in many application scenarios, including the military, emergency, health, environmental, surveillance, education, business, commercial ones, among others. because they can be quickly built, as they do not need any fixed or centralized infrastructure.

Referencias

[1] C. Adler, S. Eichler, T. Kosch, C. Schroth, M. Strassberger, Self-organized and Context-Adaptive Information Diffusion in Vehicular Ad Hoc Networks, in: International Symposium on Wireless Communication Systems. ISWCS, 2006, pp. 307–311.

[2] I. Alvarez, A. Martin, J. Dunbar, J. Taiber, D. Wilson, J. Gilbert, Voice Interfaced Vehicle User Help, in: Proc. of the Second International Conference on Automotive User Interfaces and Interactive Vehicular Applications. (AutomotiveUI 2010), Pittsburgh, Pennsylvania, USA, 2010.

[3] C. Anagnostopoulos, V. Loumos, E. Kayafas, A license plate-recognition algorithm for intelligent transportation system applications, IEEE Transactions on Intelligent Transportation Systems 7(3) (2006) 377–391.

[4] ANFAC, Sistemas ITS en el vehículo y la coordinación con la infraestructura, in: Jornada Sobre ITS para la Interacción entre el Vehículo y la Infraestructura, Madrid, 2009.

[5] F. Arani, R. Smietana, B. Honary, Real-Time Channel Estimation Using Fuzzy Logic, in: Proc. of the IEEE International symposium on Information Theory. IEEE, Whistler, BC, No. 0-7803-2453-6, 1995, p. 289.

[6] H. Asaeda, M. Rahman, H. Manshaei, Y. Fukuzawa, Implementation of Group Member Authentication Protocol in Mobile Ad-hoc Networks, in: Proc. of IEEE Wireless Communications and Networking Conference. WCNC, Las Vegas, USA, 2006.

[7] G. Avoine, www.avoine.net/rfid/ (2010).

[8] S. Basagni, K. Herrin, E. Rosti, D. Bruschi, in: Secure pebblenets, 2001.

[9] S. Bellovin, M. Merritt, Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, in: IEEE Symposium on Security and Privacy, 1992.

[10] Berg-Insight, Mobile Navigation Services. 3th edition. Berg Insight, http://social.telematicsupdate.com/files/BER44_Mobile NavigationServicesEdition.pdf (2009).

[11] Berg-Insight, GPS and Mobile Handsets, http://www.berginsight.com/ShowReport.aspx?m_m=3&id=97 (2010).

[12] J. Blum, A. Eskandarian, L. Hoffman, Mobility management in ivc networks, in: In Proc. of IEEE Intelligent Vehicles Symposium, 2003, pp. 150–155.

[13] P. Bonnefoi, D. Sauveron, MANETS: An Exclusive Choice Between Use and Security?, Computing and Informatics 22, 2003 (2007) 1001–1013.

[14] R. Bose, J. Brakensiek, K. Park, Terminal Mode – Transforming Mobile Devices into Automotive Application Platforms, in: Proc. of the Second International Conference on Automotive User Interfaces and Interactive Vehicular Applications. (AutomotiveUI 2010), Pittsburgh, Pennsylvania, USA, 2010.

[15] BuscaCar, http://es.androidzoom.com/androidapplications/tools/buscacarovyl.html.

[16] L. Buttyán, T. Holczer, I. Vajda, On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs, in: ESAS'07 Proc. of the 4th European conference on Security and privacy in ad-hoc and sensor networks, 2007, pp. 129–141.

[17] C. Caballero-Gil, P. Caballero-Gil, J.Molina-Gil, Self-organized Life Cycle Management of MANETs, Aceptado en Security and Communication Networks, 2012.

[18] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Merging Sub-networks in VANETs by using the IEEE 802.11xx protocol, Eurasip Journal of Wireless Communications and Networking. 2011. (Enviado).

[19] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Self-Organized Clustering Architecture for Vehicular Ad-Hoc Networks, Journal on Cluster Computing. 2011. (Enviado).

[20] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Zero-Knowledge Authentication in Self-Organized VANETs, IETE Journal of Research. 2011. (Enviado).

[21] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Solución Global para la Autenticación de Nodos en MANETs, in: Actas del II Simposio sobre Seguridad Informática. Congreso Español de Informática CEDI, 2007.

[22] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Group Formation through Cooperating Nodes in VANETs, Cooperative Design, Visualization and Engineering. Lecture Notes in Computer Science 6240 (2010) 105–108.

[23] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Knowledge Management Using Clusters in VANETs. Description, Simulation and Analysis, in: KMIS is part of IC3K, the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2010.

[24] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Tool to Simulate Groups in Vehicular Networks Using NS-2 and TraceGraph, in: 5th European Conference on Circuits and Systems for Communications. ECCSC, 2010.

[25] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Using Groups to Reduce Communication Overhead in VANETs, in: The Second International Conference on Advances in P2P Systems. AP2PS. Florencia, Italia, 2010.

[26] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Self-organizing Life Cycle Management of Mobile Ad hoc Networks, in: FTRA International Symposium on Advances in Cryptography, Security and Applications for Future Computing. ACSA, 2012.

[27] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, C. Hernández-Goya, A. Fúster-Sabater, Gestión de Grupos en VANETs: Descripción de Fases, in: XI Reunión Española sobre Criptología y Seguridad de la Información. RECSI, 2010.

[28] C. Caballero-Gil, P. Caballero-Gil, A. Peinado-Dominguez, J. Molina-Gil, Lightweight Authentication for RFID used in VANETs, Lecture Notes in Computer Science. 12th International Conference on Computer Aided Systems Theory EUROCAST 2011, Springer-Verlag.

[29] C. Caballero-Gil, J. Molina-Gil, Primer Premio del Concurso de Emprendedores "Conocer es Valer", http://emprendeull.ning.com/profiles/blogs/entrega-de-premiosdelconcurso- conocer-es-valer, universidad de La Laguna. Importe: 3.000 Euros (2011).

[30] C. Caballero-Gil, J. Molina-Gil, P. Caballero-Gil, Design and Implementation of VAiPho, Tool for Deploying VANETs with Phones, Computers & Electrical Engineering. 2011. (Enviado).

[31] C. Caballero-Gil, J. Molina-Gil, P. Caballero-Gil, F. Martín-Fernández, D. Yánes-García, Introducing Secure and Self-Organized Vehicular Ad-Hoc Networks, in: International Conference on Computer Systems and Technologies. CompSysTech'11, 2011.

[32] P. Caballero-Gil, Zero-Knowledge Proof for the Independent Set Problem, IEICE Transactions on Fundamentals of Electronics Communications and Computer 88-A(5) (2005) 1301–1302.

[33] P. Caballero-Gil, C. Caballero-Gil, A Global Authentication Scheme for Mobile Adhoc Networks, Advances in Information and Computer Security, Lecture Notes in Computer Science 4752 (2007) 105–120.

[34] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, RFID Authentication Protocol Based on a Novel EPC Gen2 PRNG, Information-An International Interdisciplinary Journal, 2012. (Por aparecer).

[35] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, Sistema de Comunicaciones Seguras en una Red Ad-hoc Vehicular Espontanea y Autogestionada. Patente No: P201000865. Universidad de La Laguna. Tenerife. España. Fecha de prioridad: 29 de Junio de 2010.

International Patent No. PCT/ES 2011/000220. 29 June 2011, Licencia de Comercialización Adquirida por Empresa DETECTOR, S.A. en Junio de 2011.

[36] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, An EPC Gen2 Compliant Authentication Scheme Based on a New Pseudorandom Number Generator, in: The 2011 FTRA International Workshop on Strategic Security Management for Industrial Technology. SSMIT, 2011.

[37] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, A. Fúster-Sabater, On Privacy and Integrity in Vehicular Ad hoc Networks, in: The 2010 International Conference on Wireless Networks. ICWN'10, 2010.

[38] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, C. Hernández-Goya, Self-Organized Authentication Architecture for Mobile Ad-hoc Networks, in: 6th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. Wiopt, 2008.

[39] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, C. Hernández-Goya, Flexible Authentication in Vehicular Ad Hoc Networks, in: Proc. of the 15th Asia-Pacific Conf. Communications APCC, 2009, pp. 576–879.

[40] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, A. Quesada-Arencibia, A Simulation Study of New Security Schemes in Mobile Ad-hoc NETworks, Computer Aided Systems Theory EUROCAST 2007, Lecture Notes in Computer Science 4739 (2007) 73–81.

[41] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, D. Yánes-García, F. Martín-Fernández, Detecta Atascos y Aparcamiento en tu Móvil, in: Salón atlántico de logística y transporte. SALT2011, Las Palmas de Gran Canaria, 2011.

[42] P. Caballero-Gil, J. Caballero-Gil, C.and Molina-Gil, D. Yánes-García, F. Martín-Fernández, VAiPho: Una Herramienta para la Asistencia a la Conducción, in: VIII Foro de innovaciones tecnológicas para el transporte. TRANSNOVA2011, Las Palmas de Gran Canaria, 2011.

[43] P. Caballero-Gil, C. Hernández-Goya, Strong Solutions to the Identification Problem, Proc. of COCOON. Lecture Notes in Computer Science 2108 (2001) 257–261.

[44] P. Caballero-Gil, C. Hernández-Goya, Zero-Knowledge Hierarchical Authentication in MANETs, IEICE - Trans. Inf. Syst. E89-D(3) (2006) 1288–1289.

[45] P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Data Aggregation Based on Fuzzy Logic for VANETs, in: Computational Intelligence in Security for Information Systems, Lecture Notes in Computer Science, Vol. 6694, 2011, pp. 33–40.

[46] P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, C. Hernández-Goya, Stimulating Cooperation in Self-Organized Vehicular Networks, in: 15th IEEE Asia-Pacific Conference on Communication. APCC, 2009, pp. 346–349.

[47] P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, C. Hernández-Goya, Security in Commercial Applications of Vehicular Ad-Hoc Networks, in: Financial Cryptography. Lecture Notes in Computer Science, Vol. 6052, 2010, p. 427.

[48] K. Cai, Design and analysis of a connected dominating set algorithm for mobile ad hoc networks, in: Master's thesis, The University of British Columbia, 2004.

[49] G. Calandriello, P. Papadimitratos, J. Hubaux, A. Lioy, Efficient and Robust Pseudonymous Authentication in VANET, in: VANET '07: Proc. of the 4th ACM international workshop on Vehicular Ad Hoc networks. ACM, New York, NY, USA, 2007, pp. 19–28.

[50] G. Calinescu, P.-J. Wan, Range assignment for high connectivity in wireless ad hoc networks, in: Proc. International Conference on Ad hoc and Wireless Networks, 2003, p. 235–246.

[51] J. Cano Reyes, E. Burgoa, C. Calafate, J. Cano, P. Manzoni, An autoconfiguration method for IEEE 802.11 based MANETs using Bluetooth, in: XVII Jornadas de Paralelismo. Albacete, Spain, 2006.

[52] S. Capkun, Location verification and key management in wireless networks, MSc thesis, University of Split, Croatie. EPFL. 2004.

[53] S. Capkun, L. Buttyan, J. Hubaux, Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph, in: Proc. of The ACM New Security Paradigms Workshop, 2002.

[54] S. Capkun, L. Buttyán, J. Hubaux, Self-Organized Public-Key Management for Mobile Ad-hoc Networks, in: Laboratory for Computer Communications and Applications (LCA). School of Information and Communication Sciences (I&C). Swiss Federal Institute of Technology Lausanne (EPFL). CH-1015 Lausanne, Switzerland, 2003.

[55] Car-Finder, http://es.androidzoom.com/ androidapplications/transportation/car-findersmsf.html.

[56] Car-Spotter, http://itunes.apple.com/us/app/car-spotter/id293089131?mt=8.

[57] E. Case, Advance driver information system concept for North America: A Mobility 2000 report, in: Vehicle Navigation and Information Systems Conference, 1989.

[58] I. Catling, The DRIVE programme in the European Community, in: Driver Information, IEE.Colloquium on, London, UK, 1988.

[59] D. Cavin, Y. Sasson, A. Schiper, On the Accuracy of MANET simulators, ACM Workshop on Principles of Mobile Computing (POMC 02), ACM Press, Nueva York (2002) 38–43.

[60] G. Chang, J. Sheu, C. Chung, Zooming. A Zoom-Based Approach for Parking Space Availability in VANET, in: Vehicular Technology Conference (VTC 2010-Spring), IEEE 71st, 2010.

[61] M. A. Chowdhury, A. Sadek, Fundamentals of Intelligent Transportation Systems Planning, in: Boston London: Artech House., 2003.

[62] T. Clausen, P. Jacquet, Optimized Link State Routing Protocol (OLSR), in: RFC 3626, 2003.

[63] A. E. F. Clementi, P. Penna, R. Silvestri, The power range assignment problem in

packet radio networks in the plane, in: Proc. 17th Annual Symposium on Theoretical Aspects of Computer Science. STACS, 2000, p. 651–660.

[64] COMeSafety, COMeSafety Project under European Commission within the Sixth Framework Programme priority FP6-2004-IST-4, http://www.comesafety.org/ index.php?id=9 (2006).

[65] COOPERS, CO-OPerative SystEms for Intelligent Road Safety - COOPERS - belong 6th Framework Programme by the European Commission - Information Society and Media, http://www.coopers-ip.eu/index.php?id = project (2006).

[66] C. Csehi, J. Toth, Search for Hamiltonian Cycles, The Mathematical Journal 13 (2011) 2.

[67] CVIS, Cooperative Vehicle-Infrastructure System) subproject COMO(Cooperative Monitoring, http://www.cvisproject.org/en/cvis_subprojects/ applications/como/como.htm (2006).

[68] M. Dado, J. Spalek, A. Janota, Present and future challenges of ICT for intelligent transportation technologies and services, in: Proc. of the 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE, 2009.

[69] D. Dasgupta, Immunity-Based Intrusion Detection System: A General Framework, in: Proc. of 22nd National Information Systems Security Conference NISSC, 1999.

[70] Y. Desmedt, M. Burmester, An Efficient Zero-Knowledge Scheme for the Discrete Logarithm Based on Smooth Numbers, in: ASIACRYPT '91: Proc. of the International Conference on the Theory and Applications of Cryptology. Springer-Verlag, London, UK, Vol. 739, 1993, pp. 360–367.

[71] DETECTOR, http://www.grupodetector.com/.

[72] S. Dornbush, A. Joshi, StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's, in: Proc. of the VTC2007-Spring Vehicular Technology Conf. IEEE 65th, 2007, pp. 11–5.

[73] H. Doumenc, Estudio Comparativo de Protocolos de Encaminamiento en Redes VANET, 2008.

[74] EIA-U.S, Growth of carbon dioxide emissions slows in the projections, http://www.eia.doe.gov/oiaf/aeo/pdf/trend_6.pdf (2010).

[75] T. Eissa, S. Razak, M. Ngadi, Towards providing a new lightweight authentication and encryption scheme for MANET, Wireless Networks 17 (2011) 833–842.

[76] M. Elhdhili, L. Ben Azzouz, F. Kamoun, CASAN: Clustering algorithm for security in ad hoc networks, Computer. Communications vol.31, no.13 (2008) 2972–2980.

[77] L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks, in: 9th ACM conference on Computer and Communications Security, ACM Press, 2002, pp. 41–47.

[78] European-Commission, Galileo - The European programme for Global Navigation Services. Community Research and Development Information Service (CORDIS), http://cordis.europa.eu/search/index.cfm?fuseaction=prog.document&PG_RCN=9726313 (2007).

[79] European-Commission, Proposal from the Commission to the European Parlament and Council for a regulation to reduce CO2 emissions from passenger cars, http://ec.europa.eu/environment/air/transport/co2/pdf/sec_2007_1723.pdf (2007).

[80] European-Commission, Action Plan for the deployment of Intelligent Trasnport Systems in Europe, http://ec.europa.eu/transport/its/road/action_plan_en.htm (2008).

[81] European-Commission, European Energy and Transport Trends to 2030, http://ec.europa.eu/dgs/energy_transport/figures/trends_2030_update_2007/energy_transport_trends_2030_update_2007_en.pdf (2008).

[82] European-Communities, Dedicated Road Infrastructure for Vehicle Safety in Europe (DRIVE) Final Report on Performance and Results - DRIVE, in: DG XIII Telecommunications, Information Market and Exploitation of Research. Community pro-

gramme in the field of road transport informatics and telecommunications - DRIVE, 1994.

[83] EVITA, evita-project.org/.

[84] K. Fall, K. Varadhan, Ns Notes and Documentation, The VINT Project. UC Berkeley, LBN, http://www.isi.edu/nsnam/ns/.

[85] P. Fan, J. Haran, J. Dillenburg, P. Nelson, Cluster-based framework in vehicular ad-hoc networks, Lecture Notes in Computer Science 3738 (2005) 32–42.

[86] P. Fan, P. Sistla, P. Nelson, Theoretical analysis of a directional stability-based clustering algorithm for VANETs, in: Proc. of the fifth ACM international workshop on VehiculAr Inter-NETworking, New York, USA, ACM. VANET'08, 2008, p. 80–81.

[87] FHA, Federal Highway Administration, http://www.fhwa.dot.gov/ (2007).

[88] A. Fiat, A. Shamir, How To Prove Yourself: Practical Solutions to Identification and Signature Problems, Proc. of Crypto '86, Lecture Notes in Computer Science. Springer-Verlag 263 (1986) 186–194.

[89] L. Figueredo, J. Isabel, J. Machado, J. Ferreira, J. Martins de Carvalho, Towards the Development of Intelligent Transportation Systems, in: IEEE Conference on Intelligent Transportation Systems, Proc., ITSC, Oakland (CA) USA, 2001.

[90] G-Park, http://itunes.apple.com/es/app/g-park/ id284943236?mt=8.

[91] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, in: Freeman, San Francisco, CA, 1979.

[92] GEONET, Geographic addressing and routing for vehicular communications belongs 7th Framework Program, ICT for intelligent vehicles and mobility services, http://www.geonet-project.eu (2008).

[93] V. R. Ghorpade, Y. V. Joshi, R. R. Manthalkar, Efficient public key authentication in MANET, in: Proc. of the International Conference on Advances in Computing,

Communication and Control, ICAC3 '09, ACM, New York, NY, USA, 2009, pp. 106–112.

[94] R. Ghosh, S. Basagni, Mitigating the impact of node mobility on ad hoc clustering, Wireless Communications and Mobile Computing 8, no. 3 (2008) 295–308.

[95] O. Goldreich, S. Micali, A. Wigderson, How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design, Proc. of Crypto '86, Lecture Notes in Computer Science Vol. 263. Springer-Verlag 263 (1986) 171–185.

[96] GoogleMaps, Traffic option, http://maps.google.com/.

[97] GSA, GNSS is becoming the next big technology wave, http://www.gsa.europa.eu/go/the-market/studies-and-forecasts (2009).

[98] GSA, Location-based services market ready for takeoff, http://www.gsa.europa.eu/go/news/location-based-servicesmarket-ready-for-takeoff (2009).

[99] M. Guerrero Zapata, N. Asokan, Securing ad hoc routing protocols, in: WiSE '02 Proc. of the 1st ACM workshop on Wireless security, 2002.

[100] Y. Gunter, B. Wiegel, H. Grossmann, Cluster-based medium access scheme for VANETs, in: Intelligent Transportation Systems Conference. ITSC. IEEE, 2007, pp. 343–348.

[101] J. Haas, Y. Hu, K. Laberteaux, Design and Analysis of a Lightweight Certificate Revocation Mechanism for VANET, in: VANET '09: Proc, of the sixth ACM international workshop on VehiculAr InterNETworking. ACM, New York, NY, USA, 2009, pp. 89–98.

[102] S. Hahm, Y. Jung, S. Yi, Y. Song, I. Chong, K. Lim, A Self-Organized Architecture in Mobile Ad-Hoc Networks, Proc. of ICOIN, Lecture Notes in Computer Science 3291 (2005) 689–696.

[103] C. Hernández-Goya, P. Caballero-Gil, O. Delgado-Mohatar, J. Molina-Gil, C. Caballero-Gil, Using New Tools for Certificate Repositories Generation in MANETs, Data and Applications Security XXII, Lecture Notes in Computer Science. Springer 5094 (2008) 175–189.

[104] C. Hernández-Goya, P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Cooperation Enforcement Schemes in Vehicular Ad-Hoc Networks, Computer Aided Systems Theory - EUROCAST 2009: 12th International Conference, Lecture Notes in Computer Science 5717 (2009) 429–436.

[105] C. Hernández-Goya, P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Cooperation Requirements for Packet Forwarding in Vehicular Ad-hoc Networks (VANETs), in: Proc. of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, 2009.

[106] C. Hernández-Goya, P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Extending OLSR Functionalities to PKI Management, 12th International Conference on Computer Aided Systems Theory EUROCAST. Lecture Notes in Computer Science 6928.

[107] HIDO, ITS Toolkit for road transport in countries with developing and transitional economies. Highway Industry Development Organization, http://www.hido.or.jp/itsos/ (2004).

[108] K. Hoeper, G. Gong, Models of authentications in ad-hoc networks and their related network properties, Master's thesis, Department of Electrical and Computer Engineering. University of Waterloo. Waterloo, Ontario, N2L 3G1, Canada (2004).

[109] J. Hubaux, L. Buttyán, S. Capkun, The quest for security in mobile ad-hoc networks, in: MobiHoc '01 Proc. of the 2nd ACM international symposium on Mobile ad hoc networking & computing. ACM New York, NY, USA, 2001.

[110] K. Ibrahim, M. Weigle, Towards an optimized and secure cascade for data aggregation in VANETs, in: Proc. of the fifth ACM international workshop on VehiculAr InterNETworking, VANET'08. New York, USA, ACM, 2008, pp. 84–85.

[111] IEEE-802.11, Standard Specifications for Wireless Local Area Networks, http://standards.ieee.org/wireless/.

[112] M. Imani, M. Taheri, M. Naderi, Security enhanced routing protocol for ad hoc networks, Journal of Convergence 1, No.1 (2010) 43–48.

[113] iTetris, An Integrated Wireless and Traffic Platform for Real- road traffic management solutions, ict-itetris .eu/.

[114] ITSA, Intelligent Trasportation Society of America, http://www.itsa.org/itsa_history/c48/About_Us/ ITS_America_History.html (2009).

[115] IVWSN, Intra-Vehicular Wireless Sensor Networks, lib.bioinfo.pl/projects/view/14362.

[116] D. Jiang, L. Delgrossi, IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments, in: Vehicular Technology Conference. VTC Spring 2008. IEEE, 2008.

[117] A. Juels, RFID security and privacy: a research survey, IEEE Journal on Selected Areas in Communications 24, Is. 2 (2006) 381–394.

[118] O. Kachirski, R. Guha, Intrusion detection using mobile agents in wireless ad-hoc networks, in: Proc. of the IEEE Workshop on Knowledge Media Networking, 2002, pp. 153 –158.

[119] G. Kambourakis, E. Konstantinou, A. Douma, M. Anagnostopoulos, G. Fotiadis, Efficient Certification Path Discovery for MANET, EURASIP Journal on Wireless Communications and Networking 2010 (2010) 16.

[120] L. M. Kirousis, E. Kranakis, D. Krizanc, A. Pelc, Power consumption in packet radio networks, in: Proc. 14th Annual Symposium on Theoretical Aspects of Computer Science, Vol. 1200, 1997, p. 363–374.

[121] J. Kleinberg, The Small-World Phenomenon: An Algorithmic Perspective, in: Proc. 32nd ACM Symposium on Theory of Computing, 2000.

[122] S. O. Krumke, R. Liu, E. L. Lloyd, M. V. Marathe, R. Ramanathan, S. S. Ravi, Topology control problems under symmetric and asymmetric power thresholds, Proc. International Conference on Ad hoc and Wireless Networks 2865 (2003) 187–198.

[123] V. Kumar, R. Sharma, K. Kush, Key Authentication for MANET Security, in: High Performance Architecture and Grid Computing, Vol. 169, Springer Berlin Heidelberg, 2011, pp. 497–504.

[124] L. la Kloul, F. Valois, Investigating unfairness scenarios in MANET using 802.11b, in: Proc. of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, New York, 2005.

[125] K. Laberteaux, J. Haas, Y. Hu, Security Certificate Revocation List Distribution for VANET, in: VANET '08: Proc. of the fifth ACM international workshop on VehiculAr Inter-NETworking, ACM, New York, NY, USA, 2008, pp. 88–89.

[126] L. Lamport, Password authentication with insecure communication,, Communication of the ACM 24, no. 11 (1981) 770–772.

[127] LGParking, http://es.androidzoom.com/androidapplications/lifestyle/lg-parkingueamscreenshots.html.

[128] C. Li, M. Hwang, Y. Chu, A Secure and Efficient Communication Scheme with Authenticated Key Establishment and Privacy Preserving for Vehicular Ad Hoc Networks, Comput. Commun 31(12) (2008) 2803–2814.

[129] H. Li, H. Vincent Poor, Impact of Channel Estimation Errors on Multiuser Detection via the Replica Method, EURASIP Journal on Wireless Communications and Networking 2005 (2005) 12.

[130] Z. Lidong, Z. Haas, Securing ad hoc networks, Network,IEEE Communications Society 13, Issue.6 (2002) 24–30.

[131] M. Liu, T. Lai, M. Liu, Is Clock Synchronization Essential for Power Management in IEEE 802.11-Based Mobile Ad Hoc Networks?, in: The Second IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2005.

[132] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, S. S. Ravi, Algorithmic aspects of topology control problems for ad hoc networks, Mobile Networks and Applications 10(1-2) (2005) 19–34.

[133] LocalizadorGPS, http://es.androidzoom.com/androidapplications/communication/localizador-gps-aparcar-cochetwda.html.

[134] H. Luo, S. Lu, Ubiquitous and robust authentication services for ad-hoc wireless networks, in: Technical Report 200030, UCLA Computer Science Department, 2000.

[135] A. Mahajan, N. Potnis, K. Gopalan, A. Wang, Modeling VANET deployment in urban settings, in: Proc. of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, MSWiM'07, New York, USA, ACM, 2007, p. 151–158.

[136] S. Maki, T. Aura, M. Hietalathi, Robust membership management for ad-hoc groups, in: Proc. of 5th NordicWorkshop on Secure IT Systems. NORDSEC, 2000.

[137] M. Malekzadeh, A. Abdul Ghani, S. Subramaniam, Design and Implementation of a Lightweight Security Model to Prevent IEEE 802.11 Wireless DoS Attacks, EURASIP Journal on Wireless Communications and Networking 2011 (2011) 16.

[138] MARTA, Movilidad y Automoción con Redes de Transporte Avanzadas. CDTI - Ministerio de Industria Turismos y Comercio, http://www.cenitmarta.org (2007).

[139] F. Martín-Fernández, Proyecto Fin de Carrera Dirigido por Caballero-Gil P., Caballero-Gil C. Implementación de comunicaciones seguras en la plataforma Symbian para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), Junio 2011.

[140] F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, Implementación de comunicaciones seguras en plataformas móviles para asistencia a la conducción, in: X Congreso de Ingeniería del Transporte, 2012, (Enviado).

[141] N. Maslekar, M. Boussedjra, J. Mouzna, L. Houda, Direction based clustering algorithm for data dissemination in vehicular networks, in: Proc. IEEE Vehicular Networking Conf. VNC, 2009, p. 1–6.

[142] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrashekharan, W. Xue, M. Gruteser, W. Trappe, Drive-by Sensing of Road-Side Parking Statistics, in: The ACM/USENIX Annual International Conference on Mobile Systems, Applications and Services. MobiSys, 2010.

[143] Modlok, Rompiendo el protocolo WPA con clave precompartida (PSK) y el protocolo de integridad de clave temporal (TKIP), in: http://www.xombra.com/pdf.php?nota=2451&t=1, 2006.

[144] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Countermeasures to Avoid Non-Cooperation in Fully Self-Organized VANETs, IEICE Transactions on Communications. 2011. (Enviado).

[145] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Countermeasures to Prevent Misbehaviour in VANETs, Journal of Universal Computer Science. 2011. (En Segunda Ronda de Revisión).

[146] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Probabilistic Aggregation for Data Authentication in VANETs, Transportation Research Part C: Emerging Technologies, Elsevier. 2011. (En Tercera Ronda de Revisión).

[147] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Herramientas para la Seguridad Cooperativa en Redes Ad-Hoc, Actas del II Simposio sobre Seguridad Informática. Congreso Español de Informática CEDI.

[148] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Cooperative Approach to Self-Managed VANETs, in: International Conference on Wireless Information Networks and Systems. WINSYS, 2010.

[149] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Enhancing Collaboration in Vehicular Networks, Cooperative Design, Visualization and Engineering, Lecture Notes in Computer Science 6240 (2010) 77–80.

[150] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Group Proposal to Secure Vehicular

Ad-Hoc networks, in: The 2010 International Conference on Security and Management. SAM, 2010.

[151] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, A Vision of Cooperation Tools for VANETs, in: D-SPAN. IEEE International Workshop on Data Security and PrivAcy in wireless Networks. WowMom, 2010.

[152] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Enhancing Cooperation in Wireless Vehicular Networks, in: 8th International Workshop on Security in Information Systems. WOSIS, 2011.

[153] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Reputation Lists and Groups to Promote Cooperation, in: International Conference on Computer Systems and Technologies. CompSystech'11, 2011.

[154] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, C. Hernández-Goya, Agregación de Datos para Autenticar Información en VANETs, in: XI Reunión Española sobre Criptología y Seguridad de la Información. RECSI 2010, Vol. 6927, 2010.

[155] J. Molina-Gil, P. Caballero-Gil, A. Fúster-Sabater, C. Caballero-Gil, Pseudorandom Generator to Strengthen Cooperation in VANETs, 12th International Conference on Computer Aided Systems Theory. EUROCAST 2011, Springer-Verlag. Lecture Notes in Computer Science 6927.

[156] J. Molina-Gil, P. Caballero-Gil, C. Hernández-Goya, C. Caballero-Gil, Data Aggregation for Information Authentication in VANETs, in: Sixth International Conference on Information Assurance and Security. IAS, 2010.

[157] A. Monzón, I. Otero, J. Vassallo, OASIS -Operación de Autopistas Seguras, Inteligentes y Sostenibles, in: Jornada Sobre ITS para la Interacción entre el Vehículo y la Infraestructura, Madrid. OCDE, 2009.

[158] R. Nawaz, S. Sun, Channel Estimation in 802.11G in the Presence of Bluetooth Interference, in: EUSIPCO 2008, 16th European Signal Processing Conference, Lausanne, 2008.

[159] NS-2, The Network Simulator - NS-2, http://isi.edu/nsnam/ns/.

[160] OFAV, OpenAIRE - OpenAIRE, www.openaire.eu.

[161] Oversee, Oversee - Home, https://www.oversee-project.com/.

[162] R. Panayappan, J. Trivedi, A. Studer, A. Perrig, VANET-Based Approach for Parking Space Availability, in: VANET '07: Proc. of the fourth ACM international workshop on Vehicular Ad Hoc Networks. ACM, New York, NY, USA, 2007, pp. 75–76.

[163] S. Panichpapiboon, W. Pattara-Atikom, Connectivity requirements for a self-organizing vehicular network, in: Intelligent Vehicles Symposium, 2008 IEEE. IVS, 2008, pp. 968–972.

[164] P. Papadimitratos, Z. Haas, Secure routing for mobile ad-hoc networks, in: Proc. of SCS Communication Networks and Distributed Systems Modeling and Simulation. CNDS, 2002.

[165] C. Perkings, P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers, in: SIGCOMM '94. Proc. of the conference on Communications architectures, protocols and applications, 1994.

[166] C. Perkins, E. Belding-Royer, S. Das, Ad-hoc On-Demand Distance Vector (AODV) Routing, in: RFC 3561, 2003.

[167] A. Perrig, R. Canetti, J. Tygar, D. Song, The TESLA Broadcast Authentication Protocol, RSA CryptoBytes 5 (2002) 2–13.

[168] K. Plobl, H. Federrath, A Privacy Aware and Efficient Security Infrastructure for Vehicular Ad Hoc Networks, Comput. Stand. Interfaces 30(6) (2008) 390–397.

[169] PowerUp, project homepage, www.power-up.org/.

[170] R. Ramanathan, R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. INFOCOM, 2000, p. 404– 413.

[171] Z. Rawashdeh, S. Mahmud, Media Access Technique for Cluster-Based Vehicular Ad Hoc Networks, in: Proc. VTC 2008-Fall Vehicular Technology Conf. IEEE 68th, 2008, pp. 1–5.

[172] M. Raya, J. Hubaux, The security of vehicular ad hoc networks, in: SASN 05, Proc. of the 3rd ACM workshop on Security of ad hoc and sensor networks, New York, USA, ACM, 2005, p. 11–21.

[173] RFID-Specification, EPCglobal Class 1 Gen 2, www.epcglobalinc.org/standards/.

[174] RITA, Overview of the U.S. DOT Priority ITS Initiative: Mobility Services for All Americans, from http://www.its.dot.gov/ msaa/msaa_paratransit.htm (2008).

[175] R. Rivest, The RC4 Encryption Algorithm, in: RSA Data. Security, Inc, 1992.

[176] D. A. Rosen, F. Mammano, R. Favout, Electronic Route-Guidance System for Highway Vehicles, IEEE Transactions on Vehicular Technology 19(1) (1970) 143–152.

[177] SAFESPOT, INFRASENS Cooperative System for Road Safety belong 6th Framework Programme by the European Commission - Information Society and Media, http://www.safespot-eu.org/sp2.html (2008).

[178] K. Sampigethaya, M. Li, L. Huang, R. Poovendran, CARAVAN: Providing location privacy for VANET, Embedded Security in Cars. ESCAR (2005) 8.

[179] R. Santos, R. Edwards, N. Seed, Inter vehicular data exchange between fast moving road traffic using an ad-hoc cluster-based location routing algorithm and 802.11b direct sequence spread spectrum radio, in: PostGraduate Networking Conference, 2003.

[180] P. Schartner, M. Schaffer, Unique User-Generated Digital Pseudonyms, In Computer Network Security 3685 (2005) 194–205.

[181] C. Schnorr, Efficient Identification and Signatures for Smart Cards, in: Proc. of the 1989 Advances in cryptology. CRYPTO '89. Springer-Verlag New York, Inc., New York, NY, USA, 1989, pp. 239–252.

[182] T. Schwartz, S. Castronovo, C. Endres, A GPS-less Method to Find Your Parked Car, in: Proc. of the Second International Conference on Automotive User Interfaces and Interactive Vehicular Applications. (AutomotiveUI 2010), Pittsburgh, Pennsylvania, USA, 2010.

[183] SEVECOM, Secure Vehicular Comunication, The SEVECOM project is part of the eSafety initiative, the Information Society and Media initiative, and the 6th Framework Programme of the European Commission, http://www.sevecom.org (2006).

[184] A. Shamir, Identity-based Cryptosystems and Signature Schemes, in: Advances in Cryptology-CRYPTO '84, G.R. Blakley, D. Chaum (Eds.), LNCS 196, Springer-Verlag, 1984, pp. 47–53.

[185] C. Shea, B. Hassanabadi, S. Valaee, Mobility-Based Clustering in VANETs Using Affinity Propagation, in: Proc. IEEE Global Telecommunications Conf. GLOBE-COM, 2009, p. 1–6.

[186] B. SIG, Specification of the Bluetooth system, Version 1.1, https://www.bluetooth.com (February 2001).

[187] H. Soeren, Cooperative Intelligent Transport Systems providing for sustainable transport, in: ETSI Green Agenda Seminar, Cannes, France, 2009.

[188] F. Stajano, R. Anderson, The resurrecting duckling: Security issues for ad-hoc wireless networks, Proc. of the 7th InternationalWorkshop on Security Protocols. Lecture Notes in Computer Science, Springer-Verlag Issue 4, Part Supplement (1999) 22 – 26.

[189] W. Stalling, Cryptography and network security: principles and practice, Prentice Hall Press Upper Saddle River, NJ, USA ©2010, 2010.

[190] L. G. Stavenhagen, The European initiative EUREKA: The Concept and Aims of EUREKA, Intereconomics 21 (1986) 3–6.

[191] A. Studer, E. Shi, F. Bai, A. Perrig, TACKing together Efficient Authentication, Revocation, and Privacy in VANETs, in: Proc. of the 6th Annual IEEE communica-

tions society conference on Sensor, Mesh and Ad Hoc Communications and Networks. SECON'09. IEEE Press, Piscataway, NJ, USA, 2009, pp. 484–492.

[192] H. Su, X. Zhang, H. Chen, WSN12-6: Cluster-Based DSRC Architecture for QoS Provisioning over Vehicle Ad Hoc Networks, in: Proc. IEEE Global Telecommunications Conf. GLOBECOM '06, 2006, p. 1–5.

[193] J. Sucec, I. Marsic, Hierarchical routing overhead in mobile ad hoc networks, Mobile Computing, IEE Transactions on 3, no. 1 (2004) 46–56.

[194] SUMO, Simulation of Urban MObility, http://sumo.sourceforge.net/.

[195] SYGIC, Real-Time Traffic, http://www.sygic.com.

[196] S. Takaba, Japanese Projects on Automobile Information and Communication Systems - Things Aimed at and Obtained in 20 Years' Experiences, in: Proc. of Society of Automotive Engineers, 1991.

[197] K. Takada, Y. Tanaka, A. Igarashi, D. Fujita, Road/automobile communication system (RACS) and its economic effect, in: Vehicle Navigation and Information Systems Conference, 1989.

[198] H. Tanaka, O. Masugata, D. Ohta, A. Hasegawa, P. Davis, Fast, self-adaptive timing-synchronisation algorithm for 802.11 MANET, Electronics Letters- IEEE 42; Numb. 16 (2006) 932–933.

[199] TEN-T, Projects in the field of Intelligent Transport Systems for Road Traffic (ITS) 2007- 2013. European Comission, http://tentea.ec.europa.eu/download/calls_2009/Call %2009 %20TEN-TEA %20ITS_09_FINAL.pdf (2009).

[200] P. Timmers, Business models for electronic markets, Electronic markets - 8(2). 1998.

[201] TISA, Traveller Information Services Association, http://www.tisa.org/en/welcome.htm (2009).

[202] TOMTOM, HD Traffic, http://www.tomtom.com/services/ service.php?id=2.

[203] Urban-Mobility-Report, in: Texas Transportation Institute, Texas A&M University, 2010.

[204] US.DOT, Transportation: Vision for 2030 Ensuring personal freedom and economic vitality for a Nation on the move, http://www.rita.dot.gov/publications/transportation_vision_2030/pdf/entire.pdf (2008).

[205] US.DOT, "Transportation for a New Generation" 2010 -2015, http://www.dot.gov/stratplan/dot_strategic_plan_10-15.pdf (2010).

[206] VAiPho, Mass Media, http://www.vaipho.com/prensa.

[207] VAiPho, VANET in Phones Web, http://www.vaipho.com.

[208] N. Wang, Y. Huang, W. Chen, A Novel Secure Communication Scheme in Vehicular Ad Hoc Networks, Computer Communications 31(12) (2008) 2827–2837.

[209] WAZE, Real-Time maps and traffic information based on the wisdom of the crowd, http://www.waze.com/.

[210] M. Williams, PROMETHEUS-The European research programme for optimising the road transport system in Europe, in: Driver Information, IEE Colloquium on London, 1988.

[211] H. Wilson So, J. Walr, McMAC: A Multi-Channel MAC Proposal for Ad-Hoc Wireless Networks, in: Proc. of IEEE WCNC 2007, Hongkong, 2007.

[212] L. Wischhof, Self-Organizing Communication in Vehicular Ad Hoc Networks, in: Technische Universitat, 2007.

[213] J. Wu, S. Yang, Small World Model-based Polylogarithmic Routing using Mobile Nodes, Journal of Computer Science and Technology 23(3) (2008) 327–342.

[214] H. Xiong, Z. Qin, F. Li, Secure Vehicle-to-roadside Communication Protocol Using Certificate-based Cryptosystem, IETE Technical Review 27, Is. 3 (2010) 214–219.

[215] K. Yang, X. Wang, Network Planning for Multi-radio Cognitive Wireless Networks, in: XIV European Signal Processing Conference, Florence, Italy, 2006.

[216] Y. Yang, J. Chen, L. Duan, L. Meng, Z. Gao, X. Qiu, A self-configuration management model for clustering based MANETs, in: Proc. International Conference on Ultra Modern Technology, 2009, pp. 1–7.

[217] Z. Yang, A. Høst-Madsen, Routing and Power Allocation in Asynchronous Gaussian Multiple-Relay Channels, EURASIP Journal on Wireless Communications and Networking 2006 (2006) 11.

[218] C. Yeh, Y. Huang, T. Wang, H. Chen, DESCV - A Secure Wireless Communication Scheme for Vehicle Ad Hoc Networking, Mobile Networks and Applications 14(5) (2009) 611–624.

[219] S. Yi, P. Naldurg, R. Kravets, A Security-Aware Routing Protocol for Wireless Ad Hoc Networks, in: Proc. of ACM Symposium on Mobile Ad hoc Networking & Computing. MOVIHOC, 2002.

[220] L. Zhou, Z. J. Haas, Securing ad-hoc networks, IEEE Networks 13, issue 6 (1999) 24–30.

[221] R. Zito, G. D'Este, M. Taylor, Global positioning systems in the time domain: How useful a tool for intelligent vehicle-highway systems?, Transportation Research Part C 3(4) (1995) 193–209.