Curso 2011/12
CIENCIAS Y TECNOLOGÍAS/46
I.S.B.N.: 978-84-15910-50-3

JEZABEL MÍRIAM MOLINA GIL

Nuevas herramientas de seguridad cooperativa para redes ad-hoc vehiculares

Directora
PINO CABALLERO GIL

SERVICIO DE PUBLICACIONES
UNIVERSIDAD DE LA LAGUNA

SOPORTES AUDIOVISUALES E INFORMÁTICOS
Serie Tesis Doctorales

A mis padres, Susana y Eberardo
y a mi esposo y amigo Cándido

Agradecimientos

Me gustaría expresar mi gratitud a mi incansable directora, la Dra. Pino Caballero, cuya dedicación, comprensión, paciencia y optimismo, me han sido de inestimable ayuda durante esta experiencia de investigación. Aprecio su enorme conocimiento y habilidad en muchas áreas (como criptografía, matemáticas, mundo académico, ética, interacción con compañeros, etc.), así como su apoyo y consejo para redactar solicitudes de ayudas, artículos, y, sobre todo, esta tesis.

Un agradecimiento muy especial va para la Dra. Candelaria Hernández, por sus sugerencias y materiales proporcionados, así como por su comprensión y amabilidad.

Quiero agradecer también al Dr. Alexis Quesada, a quien debo gran parte de mi formación como ingeniera.

Doy las gracias en general a mis colegas del departamento, y en especial a los doctores José Moreno, Belén Melián y Marcos Moreno, por el entorno de trabajo tan agradable.

Me gustaría expresar mi sincero agradecimiento al Dr. Wladimir Bodrow y al Dr. Otokar Grosek por darme la oportunidad de conocer las líneas de investigación desarrolladas por los grupos que ellos dirigen, y porque las estancias en Berlín y Bratislava fueron una gran inspiración para mi tesis, y aún más beneficiosas para mi desarrollo personal.

Además me gustaría agradecer a mi familia el apoyo y cariño que me ha brindado a lo largo de toda mi vida. En particular, debo dar las gracias a mi marido y amigo, Cándido, sin cuyo amor, aliento y motivación, no habría terminado esta tesis.

Finalmente, reconozco que esta investigación no habría sido posible sin la ayuda financiera de la Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias (Ayuda a la Formación de Personal Investigador).

Acknowledgments

I would like to express my gratitude to my tireless supervisor, Dr. Pino Caballero, whose dedication, understanding, patience and optimism have helped me during this research experience. I appreciate her vast knowledge and skill in many areas (e.g., cryptography, maths, ethics, interaction with peers, etc.), and her assistance in writing grant applications, papers, and especially this thesis.

A very special thanks goes out to Dr. Candelaria Hernández, for her suggestions and provided materials, as well as her understanding and kindness.

I also want to thank Dr. Alexis Quesada because I owe him a lot for my education as engineer.

In general I thank my colleagues from the department, and specially Dr. José Moreno, Dr. Belén Melián and Dr. Marcos Moreno, for the nice working environment.

I would like to express my sincere thanks to Dr. Wladimir Bodrow and to Dr. Otokar Grosek for giving me the opportunity to know the research lines developed by the groups they lead, and because the research stay in Berlin and Bratislava has been a great inspiration for my thesis and even more beneficial for my personal development.

Furthermore, I would like to thank my family for the support they have provided me through my entire life. In particular I must acknowledge my husband and best friend, Cándido, without whose love, encouragement and motivation, this thesis would not have ended.

Finally, I acknowledge that this research would not have been possible without the financial help of the Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias (Training Scholarship for Research Staff).

Contenido

Prólogo

La seguridad vial es un tema de interés general. Pese al éxito de las medidas aplicadas hasta el momento, las cifras de siniestralidad vial en la Unión Europea siguen siendo inaceptables. Si a la fundamental cuestión de los accidentes se le suman los atascos, la conclusión es que las carreteras conllevan una compleja problemática que requiere una solución urgente, tanto por las consecuencias en las pérdidas de vidas como por su negativo efecto en la economía y en el medioambiente. No es raro pues que el problema sea objeto de creciente atención, y que exista un Programa de Acción Europeo de Seguridad Vial que se centra en: mentalizar a los usuarios para que tengan un comportamiento más responsable (mayor cumplimiento de la normativa y menor tolerancia ante los comportamientos peligrosos), aumentar la seguridad de los vehículos mediante el apoyo a los avances técnicos y mejorar las infraestructuras viales gracias a las Tecnologías de la Información y la Comunicación.

Como propuesta para este último objetivo, surgen las redes ad-hoc vehiculares o VANETs (Vehicular Ad-hoc NETworks), en las que los vehículos se comunican entre sí para prevenir y/o evitar circunstancias adversas en las carreteras y lograr una gestión más eficiente del tráfico. Para que este tipo de redes llegue a convertirse en una tecnología real que garantice la seguridad pública en las carreteras, son necesarias diversas herramientas de seguridad de las comunicaciones que las protejan de los posibles tipos de ataques, entre los que destacan: ataques a la red que pongan en peligro su conectividad, ataques a la privacidad y anonimato de los usuarios, y ataques a la información modificando su contenido. El propósito fundamental de la presente memoria es la propuesta de nuevas herramientas que permitan proteger las VANETs frente a dichos ataques, asegurando en la medida de lo posible que la información generada en ellas, así como su retransmisión se realice correctamente.

Como resultado principal de esta investigación, y en colaboración con otras investigaciones, destaca VAiPho (VANET in Phones), que es una herramienta para la asistencia a la conducción que permite crear una red ad-hoc vehicular real y segura, utilizando únicamente teléfonos móviles inteligentes. En su estado actual, VAiPho hace factible el despliegue de las VANETs principalmente en entornos urbanos y con aplicación que permite la detección de atascos y plazas de aparcamiento libres y la localización de vehículo aparcado. Dicha herramienta es el origen de una patente de la Universidad de La Laguna ya licenciada a una

empresa.

Esta Tesis incluye un análisis de diversos mecanismos de seguridad necesarios para desplegar una VANET confiable y funcional en la que los nodos sean totalmente autónomos e independientes, proponiéndose una serie de nuevos protocolos, algunos de los cuales hacen uso de soluciones basadas en Criptografía.

La primera parte de esta memoria presenta un estudio de diferentes técnicas de fomento de la cooperación para animar a los usuarios a participar en las funcionalidades básicas de la red como por ejemplo la retransmisión de paquetes de información, ya sea de valor añadido o de seguridad vial. Nuestro principal objetivo en esta parte es que los nodos sean capaces de desplegar la VANET con un alto nivel de productividad, permitiendo el intercambio de información sobre los eventos que surgen en la carretera, e incrementando el alcance del rango de cobertura de los nodos dentro de la red. En concreto se presentan diferentes estrategias para motivar a los nodos a participar en la retransmisión correcta de paquetes, y asegurar una mayor disponibilidad y calidad de la red.

En la segunda parte del documento se discute la necesidad de proteger el contenido de la información retransmitida en la VANET mediante técnicas de agregación de datos. Cuando los nodos reciben un paquete no son capaces de determinar si el nodo que lo generó tenía buena o mala intención, por lo que no les resulta fácil descubrir si la información que contiene es fiable o no. Por tanto, es esencial para poder desplegar y asegurar el funcionamiento de este tipo de redes un protocolo que garantice la veracidad de la información sin suponer un retardo importante. En este trabajo se propone la verificación probabilística para mejorar la eficiencia del proceso. Además, con el objetivo de disminuir el tiempo necesario para generar la información, se utiliza una estructura de grupo que permite manejar de forma eficaz los paquetes generados.

Finalmente se presenta la implementación de algunos de los mecanismos propuestos en teléfonos móviles inteligentes. Los resultados obtenidos en entornos reales son usados para perfeccionar simulaciones NS-2 a gran escala, proporcionando una visión de los protocolos de seguridad que mejor se adaptan a las características de las VANETs. Además dicha implementación ofrece la posibilidad de desplegar una VANET real y segura de manera rápida y económica.

Preface

Road safety is a world-wide problem. Despite the success of several measures implemented by the governments, figures of road accidents in the European Union remain unacceptable. If traffic jams are considered together with the fundamental question of accidents, the conclusion is that roads involve a very complex problem that requires an urgent solution, as the consequences are losses of lives and negative effect on the economy and on the environment. Thus, it is natural that the problem is receiving increasing attention, and that a European Road Safety Action Programme exists, which is focused on: stimulating road users towards a more responsible behaviour (better respect of existing rules and better enforcement against dangerous behaviour), making vehicles safer through improved technical performance standards, and improving the road infrastructure through Information and Communication Technologies.

Closely related to the latter issue, Vehicular Ad-hoc NETworks or VANETs have been proposed as a solution where vehicles communicate with each other in order to prevent adverse circumstances on the roads and to achieve more efficiency in traffic management. In order to turn this type of networks into a reality that helps to improve road safety, several security communication tools are necessary to protect them from many possible types of attacks, such as: attacks on the network that jeopardize their connectivity, attacks on user privacy and anonymity, and attacks on the information that modify its content. The main goal of this Thesis is the proposal of new tools that allow the protection of VANETs against such attacks, ensuring as far as possible that the information generated in them and its retransmission is done correctly.

A remarkable result of this research, obtained in collaboration with other research, is VAiPho (VANET in Phones), a tool for driving assistance that allows creating a real and safe vehicular ad-hoc network by using only smartphones. In its current state, VAiPho makes it feasible the deployment of VANETs primarily in urban settings and with application in detection of traffic jams, free parking spaces and parked vehicle. This tool is the origin of a patent by the University of La Laguna that has been already licensed by a domestic company.

This Thesis includes an analysis of various security mechanisms needed to deploy a reliable and functional VANET where the nodes are fully autonomous and independent,

proposing a series of new protocols, some of which make use of solutions based on Cryptography.

The first part of this Thesis presents a study of different techniques for promoting cooperation to encourage users to participate in basic network functions such as retransmission of packets of information, whether value-added or safety-related. Our main goal in this part is that nodes are able to deploy the VANET with a high level of productivity, enabling the exchange of information about events that occur on the road, and increasing the coverage of the node within the network. Specifically, different strategies to motivate nodes to participate in relaying packets properly, and to ensure greater availability and quality of the network are presented.

The second part of the Thesis discusses the need to protect the content of the information relayed in the VANET by using data aggregation techniques. When a node receives a packet, it can not determine the intentions of the node who generated it, so it is not easy to discover whether the information the packet contains is reliable or not. Thus, a protocol that ensures the accuracy of the information without assuming a significant delay is essential to deploy and ensure the operation of such networks. This paper proposes probabilistic verification of signatures to enhance the efficiency of such a process. Furthermore, in order to reduce the time required to generate the information, a group structure is proposed to handle more efficiently the generated packets.

Finally, the implementation of some of the proposed mechanisms in smartphones is presented. The results obtained in real environments are used to enhance large-scale NS-2 simulations, providing an overview of the security protocols that best suit the characteristics of VANETs. This implementation also offers the possibility of deploying a real and secure VANET quickly and economically.

Capítulo 1

Introducción

Una red ad-hoc vehicular o VANET (acrónimo del inglés, Vehicular Ad-hoc NETwork) se define tradicionalmente como una red inalámbrica entre un conjunto de dispositivos móviles instalados en los vehículos llamados unidades a bordo u OBUs (acrónimo del inglés, On Board Units), y unidades de carretera o RSUs (acrónimo del inglés, Road Side Units), desplegadas a lo largo de la carretera. Según esta definición tradicional, en las VANETs los vehículos pueden intercambiar mensajes entre sí mediante comunicaciones vehículo a vehículo o V2V (acrónimo del inglés, Vehicle-TO-Vehicle) y también vehículo a infraestructura o V2R (acrónimo del inglés, Vehículo-TO-Road side). En una VANET todos los vehículos que participan en las comunicaciones se comportan por una parte como enrutadores o routers inalámbricos, retransmitiendo información en la red, y por otra como nodos móviles, emitiendo y recibiendo información. Por tanto, los dispositivos permiten que los vehículos establezcan conexiones multi-hop generando una red de largo alcance.

Una serie de posibles aplicaciones muy interesantes para los llamados sistemas de transporte inteligente o ITS (acrónimo del inglés, Intelligent Transport Systems) tanto de seguridad vial como de valor añadido han estimulado el estudio de este nuevo tipo de redes ad hoc. Entre las potenciales aplicaciones más destacadas están: aviso de colisión de vehículos, aviso de distancia de seguridad, asistencia al conductor, conducción cooperativa, difusión de información sobre el estado de las carreteras, acceso a Internet, mapa de localización, anuncios de aparcamiento libres, conducción sin conductor, etc.

Los servicios proporcionados por este tipo de redes estarán dirigidos a un grupo general de población, sin ningún tipo de vinculación. Esta situación da lugar a un escenario muy complejo porque se espera que los usuarios cooperen entre sí para constituir la infraestructura de la red. Sin embargo, ya que los nodos son autónomos e independientes, es difícil asumir una participación desinteresada. Por lo tanto, en este tipo de redes cabría esperar que las amenazas de seguridad provengan tanto desde fuera como desde dentro de la red.

La motivación principal de esta Tesis consiste en el desarrollo de mecanismos y tecnologías para combatir algunas de las vulnerabilidades de este tipo de redes. Por una parte se centra en la necesidad de cooperación por parte de los nodos internos para la retransmisión de paquetes dentro de la red, analizando las razones que les llevan a comportarse de manera egoísta, y proponiendo herramientas de fomento de la cooperación que eviten dicho comportamiento mediante premios o castigos. También esta Tesis intenta evitar, mediante el cifrado de la información, tanto que nodos externos puedan beneficiarse de la red accediendo a la información transmitida, como que nodos internos puedan generar información falsa, mediante la configuración en modo promiscuo. Finalmente se estudia la necesidad de asegurar que el contenido de toda la información que se emite en la red es correcta y auténtica, y que su integridad no ha sido modificada durante la retransmisión ni por nodos internos ni por nodos externos a la red, todo ello mediante la agregación de datos. Estos objetivos han sido estudiados en la teoría y las herramientas propuestas han sido implementadas en la práctica tanto en simulaciones como con implementaciones reales en teléfonos móviles jugando el papel de OBUs, de manera que hemos comprobado que con ellas el funcionamiento y la confianza en las VANETs mejora significativamente.

1.1. Historia de las VANETs

Los Comienzos

La historia del uso de las comunicaciones vía radio e infrarrojos entre vehículos y elementos situados en las carreteras está fuertemente ligada a la evolución de los ITS. El uso de comunicaciones y tecnologías para el tránsito vehicular seguro, eficiente y respetuoso

con el medioambiente se abordó por primera vez en la feria mundial de 1939 anunciada con el nombre de *Futurama* y organizada por General Motors. Dicha exposición intentaba dar una visión de futuro, mostrando diversos conceptos innovadores y previendo cómo avanzaría la tecnología en este campo.

Más tarde, hacia finales de la década de los 60, se empezaron a desarrollar los actuales sistemas basados en comunicaciones vía radio en las carreteras demostrando sus beneficios. Desde entonces la meta de la investigación ha ido cambiando constantemente a lo largo del tiempo. Tanto es así que al principio las investigaciones se enfocaban hacia sistemas que guiaban la ruta que seguía el conductor pero luego se cambió el sentido de los trabajos y se optó por poner más énfasis en los sistemas de peajes. Con el tiempo se volvió a cambiar el rumbo y se popularizaron las investigaciones sobre conducción autónoma. Cabe resaltar que el principal fin de las investigaciones, mejorar la seguridad y eficiencia en la conducción, nunca se ha visto alterado.

Las investigaciones y desarrollos en estos temas principalmente se llevaron a cabo en Estados Unidos. Sin embargo, Japón y Europa, y otras partes del mundo, no se han quedado atrás y también han hecho aportaciones importantes en este campo, llegando a hacerse bastante popular esta línea de investigación.

Un concepto clave e invariable en la investigación ha sido la búsqueda de soluciones a la financiación del sistema. Desde el principio muchos investigadores se preguntaban cómo podían rentabilizar económicamente los conceptos y aplicaciones de las VANETs. Se propusieron soluciones de diversa índole: peajes automáticos, pago de tarifas por congestión de tráfico, etc. Sin embargo, estas propuestas han sido recibidas con polémica en muchos casos. Por un lado se sitúan las industrias del automóvil y de la electrónica, que dudan de si llegará algún día en que se materialicen las infraestructuras públicas que se requieren para implementar sistemas de comunicaciones en las carreteras que permitan aprovechar la circulación de vehículos inteligentes según el modelo tradicional de VANET. Por otro lado, los organismos públicos dudan de si la tecnología en los sistemas de carreteras podría llegar a ofrecer soluciones prácticas a los problemas que ocurren actualmente en la carretera, y si merece la pena el desembolso que requerirán.

Desde los 70 a los 90 Se puede decir que las primeras propuestas concretas sobre lo que hoy en día denominamos VANETs se produjeron en la década de los 70. Por tanto, ese año puede considerarse de alguna forma el nacimiento de las VANETs como concepto. Ese año en Estados Unidos, Rosen en [123] propuso un sistema electrónico de ruta guiada llamado ERGS (acrónimo del inglés, Electronic Route-Guidance System) que según sus propias palabras: *"Es un sistema basado en el destino del conductor. Él mismo introduce en el vehículo el destino al que quiere llegar. A medida que el vehículo se acerca en cada tramo a instrumentos adicionales situados en la misma carretera, el vehículo comunica a estos instrumentos cuál es su destino, y estos instrumentos le devuelven la información de la ruta hacia ese destino".* El sistema de comunicación correspondiente operaba a 170 kHz y usaba antenas instaladas en las intersecciones de las carreteras y en la parte trasera de los vehículos para su comunicación. La tasa de transmisión de datos se situaba alrededor de los 2000 bits por segundo. De acuerdo con lo que dicen [59] [60], los esfuerzos llevados a cabo por Rosen para que su sistema ERGS fuese una realidad se vieron truncados por el alto coste que suponía la instalación en las carreteras de toda la infraestructura que conllevaba. Por otra parte, en Japón un proyecto para el control de tráfico de automóviles denominado CACS (acrónimo del inglés, Comprehensive Automobile traffic Control System) fue desarrollado entre 1973 y 1979 por la Agencia de Ciencia Industrial y Tecnología del Ministerio de Comercio Internacional e Industria del país nipón. Los objetivos de dicho proyecto, enunciados por Kawashima en [81], siguen siendo válidos después de más de 30 años:

- Reducción de la congestión de tráfico.
- Reducción de los gases producidos por los vehículos a causa de las congestiones.
- Prevención de accidentes.
- Mejora del rol público y social del automóvil.
- Dar prioridad a los vehículos de emergencia.
- Proporcionar información con rapidez a los vehículos en caso de emergencia.

El proyecto CACS incluyó una operación piloto, con 98 unidades de equipos en carretera y 330 vehículos, tal como informan Nakahara y Yumamoto en [110]. La velocidad de transmisión de las comunicaciones fue de 4,8 Kbps. Más tarde, el programa europeo PROMETHEUS (acrónimo de PROgraMme for European Traffic with Highest Efficiency and Unprecedented Safety) se puso en marcha en 1988 para estimular las actividades de investigación y desarrollo en el ámbito de la tecnología de la información y las comunicaciones móviles entre vehículos y carretera. PROMETHEUS fue apoyado por 19 países del continente europeo y por la Comisión Europea, como comenta Walker en [136] y Williams en [142]. Gillan en [62] explica la organización de PROMETHEUS, desglosándola en una serie de subprogramas:

- PRO-CAR: asistencia a la conducción mediante sistemas electrónicos.
- PRO-NET: comunicaciones entre vehículos.
- PRO-ROAD: comunicaciones entre vehículos y los sistemas de carretera.

El informe de Dabbous y Huitema [45] acerca del subprograma PRO-NET todavía se puede considerar a día de hoy aceptable en muchos aspectos. Analizaron en él las necesidades de comunicación basándose en escenarios típicos, como por ejemplo, una simple maniobra de cambio de carril. Suponiendo una estrategia de difusión periódica y exactitud en la distancia de colisión, mostraron que con una estimación conservadora de la tasa de transmisión periódica de mensajes de estado entre vehículos, cada vehículo debería realizar 20 de estas transmisiones por segundo. También indicaron que con la introducción de métodos de predicción se podría reducir significativamente el número de transmisiones necesarias. Para lograr reducir la tasa de comunicación la atención se centró en el estudio de los sistemas que operan en la banda de frecuencia de 60 GHz, como expuso Fischer en [55]. El interés en la comunicación entre vehículos continuó en Japón y Estados Unidos. Kawashima en 1990 [81] cita dos informes técnicos de 1986 y 1988 publicados por la Asociación de Tecnología Electrónica para la Conducción y el Tráfico de Vehículos, que plasmaban los resultados experimentales de la comunicación vehículo a vehículo. En Estados Unidos, como indica Shladover en 1991 [127], la principal línea de investigación estaba relacionada con la conducción autónoma del automóvil. Jurgen en 1991 [80] relataba la situación de

estos proyectos en aquella época en Estados Unidos, Japón y Europa, así como una visión de futuro sobre el papel de los sistemas que permitirán los vehículos inteligentes en las carreteras. Sachs y Varaiya en 1993 [124] presentaron los requisitos y prestaciones necesarias para poder llevar a cabo una comunicación vehículo a vehículo y una comunicación entre el vehículo y los instrumentos adicionales en la carretera necesarios para poder hacer realidad el sistema.

De los 90 a la actualidad La segunda mitad de la década de 1990 implicó hitos notables y algunos cambios importantes referentes al paradigma en este campo. De hecho el concepto de VANET se ha visto considerablemente afectado por el enorme avance de la tecnología y de la normalización desde mediados de la década de los 90. En San Diego en 1997 dentro de las actividades de demostración que organizaron los integrantes de la Asociación de California sobre Tráfico y Carreteras conocido como PATH (Partners for Advanced Transit and Highways), se mostraron diversos avances en la conducción autónoma de los vehículos de forma cooperativa. No fue el único congreso con una demostración de VANETs. En la ciudad de Tsukuba en Japón, en el año 2000 se presentó algo similar durante la fase de demostración del programa sobre avances en seguridad vehicular llamado ASV (Advanced Safety Vehicle). Europa no iba a ser menos, así que desde el proyecto europeo PROMOTE CHAUFFEUR, también se presentaron resultados. El objetivo marcado hasta ese momento de repente cambió y se pasó de pretender la conducción autónoma mediante cooperación a intentar crear un sistema cooperativo para la asistencia a la conducción.

Un año antes, en 1999 se había producido un giro bastante importante cuando la Comisión Federal de Comunicaciones de los Estados Unidos asignó 75 MHz de ancho de banda de los 5,9 GHz de la banda total al conjunto de protocolos y estándares para la comunicación inalámbrica de corto alcance entre vehículos e infraestructura conocido como DSRC (Dedicated Short-Range Communication) desarrollado para aportar neutralidad tecnológica en el área.

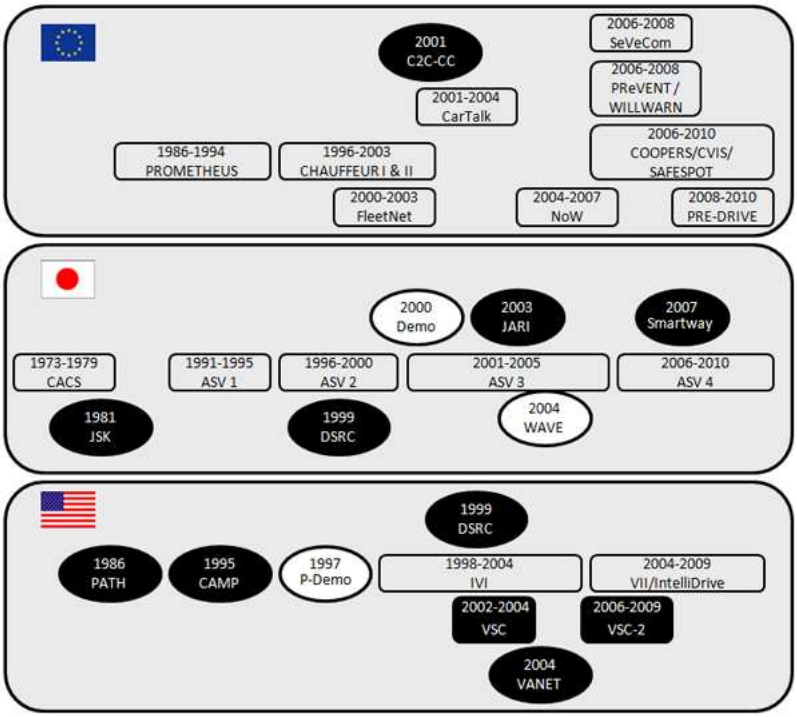
Un año después, ya metidos en pleno siglo XXI, la asociación ASTM (American Society for Testing and Materials), que es una de las organizaciones de desarrollo de normas internacionales más grande del mundo, estableció un grupo de trabajo para crear los

requisitos correspondientes al nuevo estándar DSRC. En 2001, el comité de normas 17.51 de ASTM seleccionó IEEE 802.11a como la tecnología de radio subyacente para DSRC. El estándar correspondiente se publicó finalmente en el año 2002 como ASTM E2213-02 2003 y se revisó en el año 2003 como ASTM E2213-03 2003. La presión por usar los canales asignados y la disponibilidad tanto de las tecnologías IEEE 802.11a como de los estándares, ha aumentado significativamente la investigación y el desarrollo en este campo. En particular, el interés de la comunidad de redes móviles en el tema de las redes vehiculares se revitalizó en estos años. En 2004 la asociación IEEE comenzó a trabajar en el estándar 802.11p sobre el acceso inalámbrico en entornos vehiculares WAVE (Wireless Access in Vehicular Environments), basado en el estándar de ASTM, tal como describen Jiang and Delgrossi en [79]. El proyecto sobre comunicación segura vehicular conocido por el acrónimo VSC (Vehicle Safety Communications), respaldado por la asociación de prevención de accidentes CAMP (Crash Avoidance Metrics Partnership), la administración federal de carreteras de los Estados Unidos FHWA (Federal HighWay Administration) y la administración nacional de seguridad del tráfico en las carreteras NHTSA (National Highway Traffic Safety Administration), investigó también la tecnología DSRC entre los años 2002 y 2004, llegando a la conclusión de que el enfoque basado en el estándar IEEE 802.11a sería capaz de soportar la mayoría de aplicaciones de seguridad que el proyecto VSC había probado. El informe final de dicho proyecto analizaba también los puntos débiles del estándar, como la baja latencia de la comunicación, la alta disponibilidad del canal de radio o temas generales relacionados con la capacidad del canal. En 2004, en el primer encuentro internacional organizado por la ACM sobre redes ad-hoc vehiculares que tuvo lugar en Filadelfia, se acuñó el término VANET.

Desde 1998 hasta 2005, en Estados Unidos, dentro del seno de la Iniciativa sobre Vehículos Inteligentes IVI (Intelligent Vehicle Initiative), como explican Hartman y Strasser en [68], se investigó para lograr seguridad mediante cooperación activa. En el viejo continente, los proyectos CarTalk y FleetNet, analizados por Franz en [58], investigaron las tecnologías y aplicaciones que podrían ayudar a lograr la asistencia a la conducción de forma cooperativa. Mucho más al este, en 2006 en Japón, la tercera fase del proyecto ASV reconoció el papel fundamental que debería tomar la comunicación entre vehículos en la

asistencia a la conducción de forma cooperativa.

En la Figura 1.1 a modo de resumen podemos ver en una línea de tiempo algunos de los principales proyectos y actividades llevadas a cabo en Japón, Estados Unidos y Europa a lo largo de la historia de la investigación en VANETs. Se reflejan ahí los principales proyectos de investigación y desarrollo de las distintas zonas mundiales a lo largo de su historia.



ASB	Advanced Safety Vehicle Program	JARI	Japan Automobile Research Institute	Smart wary	Smartway Public Road Test
CACS	Comprehensive Automobile Traffic Control System	JSK	Association of Electronic Technology for Automobile Traffic and Driving	VANET	ACM VANET Workshop
C2C-CC	Car-to-Car Communication Consortium	NoW	Networks on Wheels	VII	Vehicle Safety Communication Sonsortium
CAMP	Crash Avoidance Metrics Partnership	PATH	California Partners for Advanced Transit and Highways	VSC-2	Vehicle Safety Communications Two Consortium
IVI	Intelligent Vehicle Initiative	P-Demo	PATH-Demo	WAVE	Wireless Access in Vehicular Enviroments

Figura 1.1: Cronología de Proyectos sobre VANET

1.2. Objetivos de la Tesis

El fomento de la cooperación en redes vehiculares es un área que presenta grandes retos sobre todo debido a algunas características específicas de las VANETs con las que hay que lidiar especialmente a la hora de proponer alguna solución específica. Entre ellas destacan:

- Alta movilidad y topología dinámica. Los vehículos en las carreteras viajan a velocidades potencialmente muy altas. Por lo tanto, el período de comunicación entre vehículos puede ser muy corto y a la vez las altas velocidades de los nodos causan cambios frecuentes en la topología, que tiene como resultado que la información acerca de los nodos vecinos quede obsoleta en un corto período de tiempo. Así, cuando un protocolo se basa por ejemplo en tablas con información de vecinos, es bastante probable que las entradas sean antiguas e inútiles o bien inexistentes.

- Densidad variable. La densidad de nodos en las VANETs suele ser variable ya que por ejemplo, en carreteras rurales o por la noche apenas presentan nodos y sin embargo en horas punta los atascos de tráfico forman redes muy densas. Por tanto, el número de vehículos vecinos puede variar en el tiempo y/o la distancia, desde cero hasta cientos de nodos. Por otro lado tenemos la implicación de esa variabilidad de la densidad de tráfico en la consecuente variabilidad de la movilidad de los nodos, que pueden pasar de ser nodos estáticos en los atascos de tráfico, a moverse a velocidades muy altas en autopistas despejadas. Esto implica que los protocolos propuestos tienen que adaptarse tanto a escenarios dispersos y particionados como a escenarios densos, sujetos en ambos casos a cambios rápidos en el tiempo y/o distancia.

- Ancho de banda limitado. La difusión de la información en una VANET se realiza mediante un medio inalámbrico, que representa un recurso propenso a errores y con limitaciones de capacidad. Especialmente en los escenarios densos donde muchos vehículos compiten por el medio inalámbrico, el ancho de banda limitado constituye un grave problema para el protocolo de retransmisión. Por lo tanto, se requiere de un protocolo de transmisión eficiente y cooperación entre vehículos para optimizar dichas

comunicaciones y permitir el despliegue de las VANETs.

- Restricciones de tiempo. La mayoría de las aplicaciones de seguridad vial tienen un factor crítico de tiempo. Esto significa que requieren mecanismos de difusión de información que impliquen una demora mínima. Por lo tanto, cualquier mecanismo propuesto de difusión y de cooperación entre nodos tiene que transmitir la información crítica de seguridad vial sin introducir ningún retraso.

- Patrones de movimiento. Otro aspecto importante de las VANETs es que los movimientos de los vehículos se ven limitados por la topología de la carretera. Esto significa que los movimientos de los nodos obedecen a pautas de movilidad impuestas por la red de carreteras. Por lo tanto, la movilidad de los nodos es predecible y puede ser utilizada por los protocolos de enrutamiento para mejorar el rendimiento de la difusión de información.

- Privacidad. Al transmitir información, los vehículos revelan información propia que puede traducirse en un problema de privacidad. Para resolver este problema, los vehículos pueden comunicarse usando pseudónimos que deben cambiar con cierta frecuencia para evitar posibles seguimientos. El cambio de pseudónimos podría suponer un reto para el control de la cooperación de los nodos.

- Escalabilidad. El protocolo de transmisión tiene que hacer frente a grandes redes en continuo crecimiento y funcionar en redes de distintas magnitudes.

La falta de cooperación de los nodos puede hacer que la red no pueda proporcionar servicios tan básicos como la retransmisión de la información, de ahí el interés en este tema. En esta Tesis se analizan las necesidades básicas de cooperación de las VANETs. La investigación se centra principalmente en estudiar y clasificar las diferentes amenazas que llevan a los nodos a no cooperar en el reenvío de la información, evaluando su influencia en el buen funcionamiento de la red y diseñando protocolos de cooperación robustos y adecuados para estas redes, que no generen una sobrecarga o consumo excesivo de sus recursos ni perjudiquen su funcionamiento. La utilización de mecanismos para animar a los nodos en la retransmisión de paquetes en beneficio de otros nodos es necesaria para construir una red

totalmente funcional. De lo contrario, los nodos podrían simplemente apagar su interfaz de conexión con la red evitando su participación y el consumo de sus recursos. Teniendo en cuenta todos estos aspectos, en esta memoria se analiza la cooperación entre nodos en la retransmisión de mensajes según su topología, distinguiento los mensajes de valor añadido, como paquetes de Internet o publicidad, de los mensajes con información sobre eventos en la carretera, como accidentes, atascos o plazas de aparcamiento.

Además, en esta Tesis se ha diseñado un mecanismo de cifrado que intenta evitar la existencia de nodos externos, motivando a los usuarios a entrar a formar parte de la red y a la vez protegiendo la fiabilidad de la información sobre eventos frente a posibles falsificaciones de nodos internos. Para afrontar esta problemática se propone un nuevo cifrado simétrico de forma que solo los nodos pertenecientes a la red pueden beneficiarse de la información transmitida en la misma ya que requieren de la clave secreta compartida. Además la transmisión cifrada de la información permite la detección de transmisión de información falsa gracias al modo promiscuo de las comunicaciones en grupo. Teniendo en cuenta todos los aspectos mencionados, el cifrado utilizado en VANETs debe ser rápido y ligero, razón por la cual el cifrado diseñado en este trabajo es un cifrado en flujo.

Otro objetivo fundamental de esta Tesis surge de la investigación sobre la necesidad de asegurar la fiabilidad e integridad de la información generada y retransmitida dentro de la red mediante herramientas de agregación de datos basadas en la cooperación entre nodos. La mayoría de las propuestas sobre VANETs no se centran en la generación de información, siendo éste uno de los temas más sensibles en este tipo de redes. Cualquier información falsa o alterada podría influir sobre la confianza en la información retransmitida por la red. Además la retransmisión de información falsa podría tener una repercusión nefasta en el funcionamiento de la red de carreteras llegando a sobrecargar vías que no están preparadas para ello, lo que podría llegar a ocasionar accidentes de tráfico. Si los nodos crean los paquetes individualmente, es fácil que se pueda generar información incorrecta. Sin embargo, en un entorno cooperativo eso sería más complicado. Por este motivo otra de las principales investigaciones presentadas en esta Tesis es la generación de información fidedigna de forma contrastada gracias a la formación de grupos cooperativos y la correspondiente agregación de datos. Además, las topologías basadas en grupos permiten organizar el flujo

de información para reducir la sobrecarga de la red en entornos densos donde el número de paquetes generados colapsaría el canal, proporcionando una transmisión más rápida de la información. Este tipo de arquitecturas proporciona una manera intuitiva de ajustarse a la escalabilidad de este tipo de redes. Sin embargo, requieren de la existencia de un nodo conocido como líder del grupo que es responsable de manejar todo el tráfico de información generado en su grupo. Por lo tanto, será crucial elegir el nodo apropiado para este cargo de manera que cumpla con los requisitos de la red y presente un comportamiento honesto. Otro aspecto crítico en esta investigación y en concreto para esta propuesta es la comprobación de la veracidad de la información. Una vez llegue la información a un nodo, éste debe disponer de un esquema de comprobación rápido permitiendo proporcionar la información al usuario en una franja de tiempo suficiente para la toma de decisiones frente a la información recibida. Para esto se desarrolla un esquema de comprobación probabilista, rápido y eficaz. Por lo tanto, el objetivo es proporcionar un mecanismo que permita garantizar la transmisión de información correcta, utilizando para ello un esquema de agregación de datos que permita minimizar los retardos tanto en generación como en comprobación de la información.

El último objetivo de esta Tesis es poder llevar a cabo la implementación real de los mecanismos de seguridad aquí propuestos en lo que podría ser la primera VANET real del mundo. Para ello es necesario que se puedan usar dispositivos que estén al alcance del mayor número de usuarios posibles y que no supongan un gasto de instalación en los vehículos. Con estos requisitos se propone aquí el uso de teléfonos móviles inteligentes para ser usados como unidades a bordo que permitan el intercambio de información entre vehículos. De esta manera es posible implementar las propuestas no solo en simuladores sino también en dispositivos que permiten la instalación en un entorno real, pudiendo modificarlas y perfeccionarlas para lograr la eficiencia necesaria para el buen funcionamiento de los mecanismos propuestos en una VANET real.

1.3. Principales Contribuciones

El resumen de las principales contribuciones de esta Tesis en el campo de la VANETs se puede estructurar de la siguiente forma:

1. **Contramedidas para prevenir comportamientos egoístas y/o maliciosos.**

   Proponemos una serie de mecanismos de cooperación y aislamiento de nodos egoístas y/o maliciosos de la red para intentar evitar y/o contrarrestar posibles comportamientos de nodos egoístas y ataques de nodos maliciosos, haciendo uso de la revocación de dichos nodos con el objetivo de asegurar que las VANETs funcionen adecuadamente proporcionando información confiable y en tiempo real a los usuarios. Para lograr este objetivo los nodos deben cooperar activamente retransmitiendo los mensajes para alcanzar tantos nodos como sea posible. El funcionamiento de las diferentes propuestas ha sido evaluado con diversas simulaciones mostrando que no solo son eficientes sino que sirven para incrementar la seguridad de las comunicaciones. En esta temática se han publicado cuatro trabajos en volúmenes LNCS [12], [38], [70], [98], conteniendo diferentes propuestas, y se han presentado distintas ponencias en cuatro congresos indexados [97], [99], [100], [102], y en cuatro congresos no indexados [37], [71], [96], [104], entre las cuales la última recibió el premio "Best Paper Award" de la conferencia.

2. **Generador pseudoaleatorio para captar usuarios.**

   En este trabajo se propone un nuevo sistema de comunicaciones seguras para VANETs que fomenta la captación de nuevos nodos ya que imposibilita el acceso a la información desde fuera de la red. En particular se propone el uso de un cifrado en flujo que utiliza un nuevo generador de números pseudoaleatorios cuyo diseño se basa en un filtrado no lineal de un registro con realimentación lineal. Dicho generador ha sido descrito en un trabajo que ha sido aceptado para su publicación en una revista indexada [28]. Por otro lado, el cifrado se ha incluido en un volumen de LNCS [106], y presentado en un congreso [27].

3. **Agregación de datos para su autenticación.**

   Una de las principales dificultades en las redes vehiculares, donde la información de tráfico es generada por las más diversas y variadas fuentes y retransmitida a numerosos destinos, es la necesidad de un mecanismo de autenticación de datos para detectar cualquier posible comportamiento malicioso de usuarios que realicen ataques de modificación o repetición de la información. En esta Tesis se propone un nuevo protocolo de agregación de datos para VANETs, que utiliza verificación probabilística para detectar esos ataques de una manera eficiente a posteriori con mínima sobrecarga y retardo. La propuesta también contiene un mecanismo de seguridad adicional que utiliza la idea de grupos reactivos creados bajo demanda, para garantizar a priori que los vehículos generan información confiable. De acuerdo con un análisis integral de la propuesta incluyendo numerosas simulaciones, se demuestra que es robusta. Los resultados más importantes con respecto a este tema se incluyen en un trabajo que se encuentra en tercera fase de revisión en una revista con un alto factor de impacto [103]. Versiones preliminares han sido publicadas en un volumen de LNCS [36], y presentadas en un congreso indexado [108] y en un congreso nacional [107].

4. **Implementación en dispositivos reales.**

   Si bien el resto del trabajo es aplicable a todo tipo de VANETs, tal y como se definen tradicionalmente en la bibliografía, con OBUs y RSUs, durante el desarrollo de la investigación identificamos como problema la implementación de las propuestas en un entorno real con el objetivo de analizar su funcionamiento ya que el entorno planteado en la bibliografía no se ha puesto en marcha aún. Por una parte, las OBUs son dispositivos instalados en los vehículos incluyendo diferentes partes como por ejemplo sensores, y por lo tanto suponen un coste imposible de cubrir para la realización de una Tesis en un entorno académico. Por otro lado, y aunque se esperaba estuviesen implementadas en 2011, actualmente no se cuenta con la presencia de RSUs en las carreteras que permitan la constitución de una VANET. Con el fin de poder continuar con el presente trabajo se planteó la posibilidad de hacer realidad estas redes sin que supusiera ningún coste pero que permitiera continuar con la presente investigación y

llegar a su implementación real. Teniendo en cuenta que mucha bibliografía propone comunicar los vehículos mediante conexion Wi-Fi utilizando el protocolo 802.11, y dado que los teléfonos inteligentes actuales proporcionan dicha conexión, se decidió en esta Tesis programar los teléfonos móviles para desplegar una VANET real haciendo uso de dichos dispositivos. De esta manera se lograron implementar algunos de los protocolos aquí propuestos en dispositivos móviles, utilizándolos como OBUs dentro de los vehículos con el fin de obtener datos reales y analizar su funcionamiento. Esta idea de puesta en marcha de una VANET real y segura con teléfonos móviles ha sido denominada VAiPho (acrónimo del inglés, VAnet in Phones). Es de destacar que con VAiPho hablamos de una red completamente distribuida y descentralizada sin ninguna autoridad central ni RSUs a diferencia de las propuestas hasta el momento encontradas en la bibliografía. Son muchos los problemas de implementación en teléfonos móviles que se han tenido que solucionar para lograr este desarrollo. Esta Tesis recoge las soluciones implementadas relacionadas con la cooperación, la agregación y el cifrado. Podemos decir que VAiPho es la principal contribución de esta Tesis ya que esta invención fue patentada en la Universidad de La Laguna en 2010 en su fase nacional, encontrándose actualmente en fase de internacionalización [26], y en 2011 la licencia de dicha patente ha sido adquirida para su explotación por la empresa nacional radicada en Madrid, DETECTOR S.A. Asimismo las ideas de VAiPho fueron objeto del primer premio del concurso de emprendedores “Conocer es Valer” de la Universidad de La Laguna [23], así como de comunicaciones en tres congresos [25], [33], [34], y de un proyecto fin de carrera dirigido que obtuvo la máxima calificación [44].

1.3.1. Revistas Indexadas y LNCS

- [12] 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.

- [16] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Merging subnetworks in VANETs by using the IEEE 802.11xx protocol. Submitted to Eurasip Journal of Wireless Com-

munications and Networking, 2011.

- [17] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organized clustering architecture for Vehicular Ad-hoc Networks. Submitted to Journal on Cluster Computing, 2011.

- [19] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organized Life Cycle Management of MANETs, Accepted by Security and Communication Network, 2012.

- [20] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Zero-knowledge authentication in self-organized VANETs. Submitted to IETE Journal of Research, 2011.

- [22] Caballero-Gil, C., Caballero-Gil, P., Peinado-Domínguez, A., Molina-Gil, J. Lightweight authentication for RFID used in VANETs. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6927, 2011.

- [24] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P. Design and implementation of VAiPho, tool for deploying VANETs with phones. Submitted to Computers & Electrical Engineering, 2011.

- [28] 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.

- [32] 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.

- [36] 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.

- [38] 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.

- [69] 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.

- [70] 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.

- [72] 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.

- [98] 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.

- [101] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Prevent Misbehaviour in VANETs. In second round review at Journal of Universal Computer Science, 2011.

- [103] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Probabilistic Aggregation for Data Authentication in VANETs. In third round review at Transportation Research Part C, 2011.

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

- [106] Molina-Gil, J., Caballero-Gil, P., Fúster-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.

1.3.2. Congresos Indexados

- [13] 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.

- [29] 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.

- [30] 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.

- [97] 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.

- [99] 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.

- [100] 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.

- [102] 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.

- [108] 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.

1.3.3. Otros Congresos

- [11] 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.

- [14] 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.

- [15] 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.

- [18] 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.

- [21] 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.

- [25] 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.

- [27] Caballero-Gil ,P., Caballero-Gil, C., Molina-Gil, J. An EPC Gen2 compliant authentication scheme based on a new pseudorandom number generator. FTRA International Workshop on Strategic Security Management for Industrial Technology, 2011.

- [31] 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, 2009.

- [33] 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. SALT, 2011.

- [34] 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. TRANSNOVA, 2011.

- [37] 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, 2009.

- [71] 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.

- [91] Martín-Fernández, F., Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Implementación de comunicaciones seguras en plataformas móviles para asistencia a la conducción. Submitted to X Congreso de Ingeniería del Transporte. 2012.

- [96] 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.

- [104] 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.

- [107] Molina-Gil, J., Caballero-Gil, P., Hernández-Goya, C., Caballero-Gil, C.: Agregación de datos para autenticar información en VANETs XI Reunión Española sobre Criptología y Seguridad de la Información RECSI. 2010.

1.3.4. Otras Contribuciones

- [23] 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.

- [26] 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., en Junio de 2011.

- [44] Yanes-García D. End-term Project Directed by Caballero Gil P., Molina Gil J. Implementación de comunicaciones seguras en la plataforma Android para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), June 2011.

1.4. Estructura de la Memoria

El resto de esta memoria se organiza como sigue:

El Capítulo 2 comienza con un breve estado del arte con los principales trabajos existentes sobre cooperación en VANETs. A continuación se analizan la importancia de la cooperación en este tipo de redes así como la necesidad de la misma y algunos parámetros importantes. Además se presentan las nuevas propuestas de cooperación entre nodos para la retransmisión de información tanto de valor añadido, como sobre eventos en la carretera, incluyendo simulaciones que permiten analizar las propuestas. Finalmente incluye una descripción detallada de la propuesta de un nuevo generador de números pseudoaleatorios que tiene como objetivo captar el mayor número de usuarios posibles de manera que mejore la conectividad de la red, evitando la transmisión de información falsa. Además contiene un breve estudio de las secuencias que produce.

El Capítulo 3 empieza también con un breve estado del arte, esta vez relacionado con la agregación en VANETs. En cada sección se detallan cada una de las fases corres-

pondientes a un nuevo esquema de agregación propuesto. Acaba el capítulo con un análisis teórico y diversas implementaciones y evaluaciones de simulaciones.

El Capítulo 4 presenta la implementación real de algunos de los protocolos aquí propuestos en dispositivos móviles reales, detallándose la nueva herramienta VAiPho, que proporciona soluciones para la detección automática de eventos en la carretera tales como atascos de tráfico y plazas de aparcamientos libres así como el intercambio seguro y eficiente de información.

Finalmente, algunas conclusiones, una discusión acerca de la investigación realizada y posibles trabajos futuros se incluyen en el Capítulo 5.

Capítulo 2

Cooperación

La cooperación en términos generales consiste en el trabajo en común llevado a cabo por parte de un grupo de entidades hacia un objetivo compartido, generalmente usando métodos también comunes, en lugar de trabajar de forma separada en competición, logrando incrementar la productividad y el trabajo en equipo. En este capítulo se presenta un estudio de diferentes necesidades de cooperación en VANETs, así como un conjunto de soluciones propuestas para afrontar dichas necesidades.

La mayoría de operaciones en redes vehiculares requieren la cooperación de los nodos para asegurar su funcionamiento. Un ejemplo de ello es la retransmisión de paquetes dentro de la red ya que si los nodos no cooperaran en dicha operación, el intercambio de información a través de la red no sería posible. Además, tal y como se verá a continuación, la cooperación permite resolver problemas diversos tales como la garantía de la calidad de las comunicaciones, la detección del mal comportamiento y la prevención contra la inyección o retransmisión de información falsa. El propósito de la investigación realizada ha sido proponer un conjunto de contramedidas para evitar comportamientos egoístas o ataques activos diversos, haciendo uso de la cooperación entre nodos, considerando para ello diferentes factores que hacen que los nodos tiendan a comportarse de manera egoísta. El funcionamiento de las técnicas propuestas se ha evaluado en numerosas simulaciones, y los resultados muestran que las medidas descritas en este capítulo no solo aumentan la eficiencia en el funcionamiento de la red sino también la seguridad de las comunicaciones. El contenido

de este capítulo ha sido publicado en las referencias [37], [38], [70], [71], [97], [98], [99], [100], [102], [104].

2.1. Planteamiento del Problema

Desde las primeras investigaciones realizadas sobre redes ad-hoc, se planteó el fomento de la cooperación entre nodos como un requisito indispensable para su buen funcionamiento. Siempre se puede suponer que cada nodo tiene como objetivo maximizar su propio beneficio, disfrutando de los servicios de red y al mismo tiempo reduciendo al mínimo su contribución en la misma, lo que implica la posible presencia de nodos hostiles y egoístas que dañan de forma pasiva la red, degradando su rendimiento y poniendo en peligro su conectividad.

Las VANETS, como una evolución de las redes móviles ad-hoc o MANETs (Mobile Ad-hoc NETwork), también presentan esta necesidad de cooperación. Sin embargo, y tras analizar su funcionamiento y aplicaciones potenciales concluimos que los requerimientos de cooperación no son exactamente los mismos que en otras redes ad-hoc ya que por sus características propias, además de las necesidades de cooperación existentes en redes ad-hoc en general, presentan otras nuevas. La alta movilidad de los nodos, limitaciones de operatividad en tiempo real y de conectividad, escalabilidad, despliegue gradual y privacidad son algunos de los retos que presentan estas redes. Por tanto, en la presente memoria se propone el uso de herramientas para estimular la cooperación entre nodos de las VANETs que se adaptan a estas características haciendo uso de la tecnología actual para de este modo lograr alcanzar el máximo potencial de dichas redes.

Existen diversas situaciones donde la comunicación entre vehículos ayuda a prevenir accidentes y evitar colapsos, pero requieren que los nodos colaboren en la retransmisión de la información. Sin embargo, el comportamiento de nodos egoístas podría llegar a dividir la red en varias subredes de manera que dejara de funcionar, produciendo de forma pasiva lo que se conoce como ataque por denegación de servicios o DoS (acrónimo del inglés, Denial of Service). Además, es razonable asumir que los nodos tengan como objetivo maximizar su propio beneficio, disfrutando de los servicios proporcionados por la red, y al mismo tiempo

minimizar su contribución a la misma, disminuyendo de este modo el consumo de sus recursos. Por tanto, queda más que justificada la necesidad de motivar de algún modo a los nodos a participar en la red en beneficio de otros. En este trabajo se trata de lidiar con esta problemática haciendo uso herramientas de fomento de la cooperación. En concreto nos centramos en dos casos de fomento de la cooperación en la retransmisión de paquetes, según el tipo de información que contiene:

- Cooperación en Retransmisión de Paquetes de Valor Añadido.
- Cooperación en Retransmisión de Paquetes de Eventos en Carretera.

Una forma adicional de fomento de la cooperación se analiza también aquí al proponer el uso de cifrado que permite la captación de usuarios.

2.2. Estado del Arte

El fomento de ls cooperación ha estado presente desde las primeras investigaciones en redes ad-hoc en general y en MANETs en particular. Buttyan y Hubaux proponen en [7] y [8] el uso de monedas virtuales como incentivos para estimular el reenvío de paquetes en MANETs. En [125] también se realiza una propuesta de fomento de la cooperación en MANETs en la que se motiva a los nodos a retransmitir la información en beneficio de otros nodos haciendo uso de un sistema de cargos y recompensas. El objetivo de los protocolos basados en crédito es implicar al mayor número de usuarios posibles en la realización de las tareas básicas de la red. Para ello utilizan la idea de monedas virtuales con el fin de estimular un comportamiento cooperativo de los nodos. Así los nodos que generan paquetes pagan por que otros nodos los retransmitan. Otros ejemplos son CASHnet [141], Sprite [149], Ad-hoc VCG [2], Pricing for Forwarding [76], PIFA [146], etc. Estos protocolos requieren la existencia de una autoridad externa que mantenga la información del sistema, permitiéndole recordar la recompensa que se debe pagar a cada nodo de la red.

Además de los protocolos basados en crédito podemos encontrar protocolos basados en reputación, los cuales para incentivar la cooperación consideran la colaboración como un requisito obligatorio. El principal objetivo de estos protocolos consisten en detectar,

identificar, castigar y aislar de la red a todos los usuarios malintencionados. Estos protocolos permiten distinguir los usuarios que presentan mal comportamiento intencionadamente frente a los que no transmiten por temas de niveles de batería bajos, pérdida de conexión, etc. Para implementar esta idea se han propuestos dos tipos de soluciones: monitorizar la red o utilizar un hardware resistente a modificaciones. Algunos de estos protocolos basados en reputación son: CONFIDANT [5], OCEAN [3], INRIA [87], Context Aware scheme [117], CineMa [57], etc.

Sin embargo, ya en la bibliografía se comenta que las soluciones propuestas para MANETs no son directamente aplicables a las VANETs. Por ello, Li et al. en [85] describen algunas características únicas deseables para los sistemas de incentivos en VANETs, y proponen un esquema de recompensas basado en recibos con el objetivo de incentivar a los nodos en la retransmisión de paquetes. Sin embargo, este esquema basado en recibos tiene un importante problema de consumo incontrolado. De hecho, la mayoría de las soluciones existentes basadas exclusivamente en mecanismos de recompensa, sufren de falta de garantía de imparcialidad y dependen normalmente de la existencia de un hardware costoso a prueba de manipulación o bien de una tercera parte de confianza.

Otros autores han propuesto enfoques más relacionados con el nuestro al analizar el tema de la cooperación en VANETs desde el punto de vista de la confianza y la reputación. La propuesta llamada VARS [51] plantea el uso de la confianza directa e indirecta, así como una puesta en común de diferentes opiniones de nodos sobre si confiar o no en el contenido de los paquetes recibidos. El principal problema de esa propuesta es que utiliza un conjunto de evaluaciones basadas en la reputación de los nodos durante un largo período de tiempo. Para promover la cooperación entre los nodos y proteger los paquetes durante su propagación en VANETs, [138] propone utilizar un token de confianza dinámico para mejorar los mecanismos de cooperación. El objetivo es no solo hacer cumplir la cooperación entre nodos en VANETs sino proteger la integridad de los paquetes en cada salto durante la propagación. Para eso el modelo utiliza un WatchDog encargado de generar un token de confianza basado en la observación de primera mano. Otro sistema basado en reputación fue descrito en [137], donde las relaciones de confianza y las decisiones de aceptación de paquetes se basan en la observación inmediata y el comportamiento de los nodos durante la

retransmisión. [126] describe un mecanismo para la detección de posibles nodos maliciosos mediante el uso de tres módulos diferentes, cuya suma determina la reputación de los nodos. Sin embargo, todos estos sistemas de vigilancia tienen como inconveniente que no descartan la posibilidad de propagar paquetes manipulados.

Los nodos maliciosos pueden llegar a dividir una VANET en subredes, impidiendo que pueda proporcionar servicios tales como el establecimiento de rutas o el envío de paquetes entre usuarios legítimos. En este sentido, el comportamiento de los nodos maliciosos puede causar una DoS. [77] discute algunas de las principales amenazas a la seguridad y ataques como ese, que se pueden explotar en las VANETs. Del mismo modo, [109] utiliza el enrutamiento de las comunicaciones, e introduce la cooperación basada en una estructura de grupos como un servicio más de la red. [56] ofrece una visión general de los enfoques existentes que tratan de garantizar la seguridad de enrutamiento para MANETs y analiza si estos métodos pueden ser aplicados para asegurar las VANETs. Dichos documentos se centran principalmente en el diseño y análisis de esquemas de enrutamiento.

[145] propone un esquema utilizando grupos de vehículos, que se centra en la coordinación descentralizada de manera que los nodos puedan cooperar en entornos complejos para aplicaciones concretas. Un buen ejemplo de aplicación de VANETs que requiere de cooperación se describe en [83], que propone un marco para la difusión de publicidad en VANETs donde los vehículos reciben incentivos por la transmisión y almacenamiento de los anuncios. A diferencia de las propuestas anteriores, que funcionan sobre mecanismos de recompensa, otros trabajos, que también estudian aplicaciones concretas de las VANETs, se basan en esquemas de castigo y reputación los cuales se fundamentan en la supervisión y análisis del comportamiento de los nodos en la red. [88] utiliza un sistema para evitar que los nodos retransmitan mensajes de tráfico falsos, que determina si los mensajes recibidos son importantes y dignos de confianza para el conductor.

Casi todas las herramientas mencionadas anteriormente basadas en reputación, utilizan básicamente el sistema propuesto en [121], el cual exige la existencia de una Autoridad de Certificación o CA (acrónimo del inglés, Certificate Authority) que es la responsable de entregar un par de claves pública/privada y el correspondiente certificado a cada nodo. En particular, [129] propone que esa función la realice una autoridad regional de trans-

porte, que puede ser el Estado, la provincia, etc., mientras que otros autores proponen la existencia de un "departamento de vehículos motorizados" [86]. Por lo tanto, ninguna de estas soluciones pueden ser consideradas aplicables para redes totalmente distribuidas y descentralizadas, como las que se discuten en este trabajo. Otras referencias cercanas a la idea de grupo aquí presentada proponen el uso de grupos cooperativos para coordinar acciones entre varios vehículos con objeto de tomar una decisión óptima. [4] utiliza la idea de grupos cooperativos para evitar obstáculos o accidentes basándose en la predicción y en la definición de umbrales de forma que cuando la distancia mutua entre dos objetos cae por debajo de cierto umbral se detecta una situación de peligro. [10] también propone la idea de utilizar grupos cooperativos, donde los coches intercambian información proporcionando una visión común que les permite detectar situaciones críticas. Cuando se enfrentan a un obstáculo toman las decisiones óptimas para el grupo, que derivan y distribuyen en forma de secuencias de acciones individuales. Al igual que las propuestas aquí presentadas, ambos trabajos utilizan la idea de formación de grupos para la detección de eventos en la carretera. Sin embargo, en este trabajo el objetivo de los grupos es reducir el número de paquetes generados y aumentar la eficiencia en la generación de los eventos, y no la conducción cooperativa.

2.3. Necesidad de Cooperación

En las redes ad-hoc las comunicaciones se realizan a saltos (multi-hop) de forma que emisor y receptor hacen uso de nodos intermedios como retransmisores de paquetes para poder llevar a cabo su conexión. Esto supone un consumo de recursos para los nodos intermedios, los cuales generan un gasto de su propia batería en beneficio de otros nodos. De ahí la importancia de la cooperación en el enrutamiento en redes ad-hoc. Sin embargo, en VANETs por ejemplo cuando se detecta un evento de seguridad vial, el funcionamiento adecuado consiste en enviar esta información al mayor número de vehículos posibles con el fin de disminuir la posibilidad de accidente. Por tanto, y haciendo uso de la red para este fin, no es necesario utilizar un protocolo de enrutamiento que realice conexiones origen-destino como en las redes ad-hoc en general. El objetivo será sin embargo propagar los eventos al

mayor número de usuarios posible, intercambiando los eventos recibidos o detectados por cada nodo con todos los vehículos que encuentran en su ruta. Por tanto, el intercambio de mensajes en este trabajo consiste en conexiones punto a punto con todos los vehículos que se encuentren en la carretera dentro de un rango de interés con objetivos tales como disminuir los accidentes o el número de vehículos concentrados en un atasco. En ningún caso, como ocurre en las redes ad-hoc en general, los nodos serán simples retransmisores de un mensaje, sino que actuarán como receptores de la información que se envía por la red. Sin embargo, para que dicha información se propague por la red, sigue siendo necesario que los usuarios retransmitan los paquetes recibidos.

A la hora de establecer conexiones entre vehículos, se debe considerar que en muchas ocasiones los nodos se mueven a grandes velocidades en comparación con los nodos de las MANETs en general, o por ejemplo de las redes de sensores. Además, los enlaces en estas redes son intermitentes, ya que los vehículos sólo se conectan entre sí cuando se encuentran y por tanto no existen conexiones fijas entre ellos. Por este motivo, en este trabajo se utiliza el paradigma descrito en [86] conocido como "Almacenar y Llevar" (store and carry), donde los nodos almacenan los eventos durante cierto período de tiempo, y cuando se encuentran y conectan con otros nodos realizan un intercambio de información. Esta propuesta acarrea varios problemas que pueden conducir a un comportamiento egoísta y/o pasivo por parte de los nodos a la hora de cooperar en la red. Uno es la limitación del espacio de almacenamiento necesario para guardar todos los eventos que se generan y/o reciben. Otro es el requerimiento de tiempo de cómputo para la gestión de dicha información, y por último está el consumo de batería producido durante el envío de paquetes para informar a los nodos vecinos.

En este trabajo, con el fin de determinar si realmente es necesaria una herramienta para fomentar la cooperación en estas redes en general, se han realizado inicialmente diversas simulaciones NS-2 básicas para analizar cómo afectaría al comportamiento de la red la existencia de nodos egoístas no retransmisores. Para las simulaciones se han utilizado redes de 100 nodos en un área de 500 m^2 donde un 5 % se comporta de forma egoísta y no retransmiten paquetes ajenos. El movimiento de los nodos en la red así como las comunicaciones entre ellos se ha realizado de forma totalmente aleatoria. El número de conexiones que se

lleva a cabo oscilado entre 20 y 50 por cada simulación y el experimento se ha realizado 100 veces.

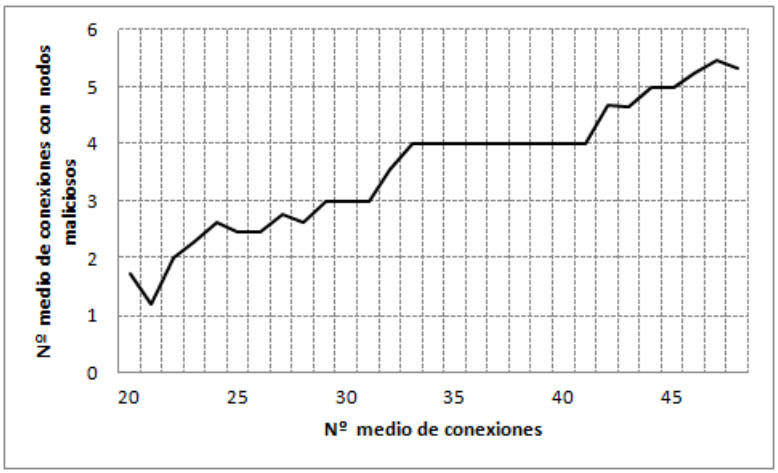


Figura 2.1: Conexiones con un 5 % de nodos maliciosos

En la Figura 2.1 se muestran los resultados obtenidos de las simulaciones. Se puede estimar que la probabilidad de realizar una conexión con uno de estos nodos maliciosos en el peor de los casos es de un 12 % del total de las comunicaciones que se realizan, lo cual supone una probabilidad bastante baja. En estas condiciones los nodos maliciosos no pueden ser considerados una amenaza seria para el buen funcionamiento de la red.

Sin embargo, se podría esperar un porcentaje mayor de nodos maliciosos, lo que sí supondría una gran amenaza, según se observa en la Fig 2.2 ya que con un 10 % de nodos maliciosos en redes con las mismas condiciones anteriores, se observa que el porcentaje esperado de conexiones con nodos maliciosos es de una media de 19,6 % lo cual aún sería admisible en algunas condiciones. Cuando el porcentaje de nodos maliciosos aumenta a un 30 % ya hablamos de que más de la mitad de las conexiones sería con algún nodo malicioso y de ahí en adelante empeora a medida que aumenta el número de nodos maliciosos en la red. En estas condiciones se observa que es necesario motivar a los usuarios para que cooperen en las comunicaciones dentro de la red.

La mayor utilidad de las VANETs es proporcionar a los usuarios información a tiempo real de lo que está sucediendo en la carretera, permitiéndoles evitar zonas de

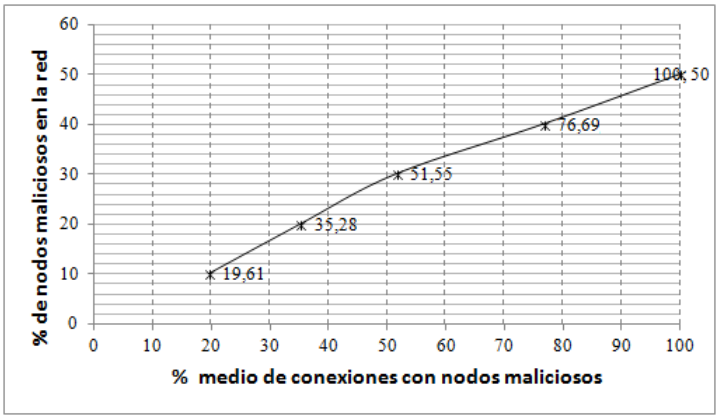


Figura 2.2: Conexiones con un % variable de nodos maliciosos

congestión y tomar decisiones con suficiente antelación con el fin de evitar accidentes. Si un vehículo no coopera en la transmisión de información a otros vehículos, sería conveniente que dicho vehículo no fuera capaz de recibir información sobre eventos en las carreteras ya que de esa forma eso sería una motivación suficiente para cooperar en la retransmisión. Por tanto habría que evitar la posibilidad de que los nodos puedan actúar de forma pasiva recibiendo información de la red y no almacenando ni retransmitiendo los paquetes. Por esta razón en este trabajo se define una herramienta que permite determinar si los nodos están cooperando en la retransmisión de paquetes dentro de la red, y tomar medidas en caso de que no sea así de forma que se les impida por ejemplo aprovecharse de la información que viaja por la red.

Otro aspecto fundamental para el buen funcionamiento de las VANETs es garantizar que la información proporcionada a los usuarios es fiable y actual. Para alcanzar este objetivo los nodos deben cooperar en el envío activo inmediato y continuado de anuncios de eventos totalmente contrastados a todos aquellos nodos que van encontrando durante su vida en la red. También resulta útil la cooperación entre nodos para la generación de anuncios de eventos reales confirmados. Estos son algunos de los aspectos, junto con la limitación de espacio de almacenamiento y batería, que se han tenido en cuenta al diseñar las herramientas propuestas en este capítulo.

2.4. Parámetros de Cooperación

2.4.1. El Espacio de Almacenamiento

A fin de asegurar un uso óptimo del espacio de almacenamiento, se definen aquí dos parámetros de decisión sobre si cooperar o no en la retransmisión de un paquete almacenado. En función del *tiempo* y la *distancia* se establece la permanencia o borrado de paquetes en memoria evitando que los eventos permanezcan indefinidamente almacenados.

El parámetro tiempo implica que el nodo origen añada a cada evento de advertencia una *Marca de Tiempo* que indica el momento en que se detecta y/o genera un paquete sobre el evento, de modo que permita el borrado del mismo una vez expire el plazo estipulado. El nodo origen fija este valor teniendo en cuenta que el tipo de evento puede variar, ya que un evento de aparcamiento no es igual ni tiene la misma importancia que un evento de atasco. Por tanto, se debe considerar una definición de plazo de tiempo diferente T_E para cada tipo de evento E. Del mismo modo, el tipo de vía V también influye en el tiempo total que ha de ser considerado para el almacenamiento de eventos. Así, por ejemplo los factores f_V para carretera convencional y autopista o autovía son diferentes, porque el tiempo de conexión en autopista es generalmente más corto que en carretera convencional debido a que la velocidad de los vehículos en este tipo de vías es mayor. Por lo tanto, el tiempo de almacenamiento $T_{E,V}$ para cada evento E en una vía V se puede calcular según la fórmula:

$$T_{E,V} = T_E \cdot f_V$$

Por ejemplo, en la implementación real de este trabajo descrita en el último capítulo se utilizan como tiempo básico para un atasco $T_E = 5min$ y para la detección de un aparcamiento $T_E = 90s$, y $f_V = 2$ para carretera convencional mientras que para autopista o autovía proponemos $f_V = 1$. En cualquier caso, los valores óptimos de estos parámetros se obtendrán y ajustarán después de ver su funcionamiento en dispositivos y en situaciones reales con implementaciones a gran escala, siendo considerados hasta el momento los valores indicados, valores óptimos para las simulaciones.

Por otro lado, la función de distancia permitirá descartar paquetes que se han generado más allá de cierto rango porque en muchos casos la información generada en una

cierta localización deja de ser interesante fuera de cierto radio. Este radio depende del tipo de evento E y del tipo de vía V y debe ser fijado por el nodo encargado de generar el anuncio del evento. Por ejemplo si se está retransmitiendo un anuncio de evento de parking libre en el centro de una ciudad, para un vehículo que está circulando a dos kilómetros de distancia esta información no será de su interés. Sin embargo si en dos kilómetros este mismo vehículo se va a encontrar con un atasco, recibir esta información le será muy útil para poder evitarlo. Esto significa que el valor del radio R_E, que define el almacenamiento y retransmisión de un evento, debe ser más grande o más pequeño, dependiendo del tipo de evento E. Además, la función de distancia para establecer el valor de expiración también depende del tipo de vía V. Por ejemplo, en el mismo espacio de tiempo, un vehículo recorre una distancia mayor en una autopista que en una carretera convencional, por lo que el factor f_V también debería ser considerado pero en este caso como un divisor. En conclusión, el radio fuera del cual un evento debe ser borrado $R_{E,V}$ para cada evento E en la vía V puede ser expresado en función del radio R_E definido para cada evento E según la siguiente fórmula:

$$R_{E,V} = R_E / f_V$$

Teniendo en cuenta ambos parámetros $T_{E,V}$ y $R_{E,V}$ para la expiración de eventos almacenados en memoria, logramos hacer un uso óptimo del espacio de almacenamiento, ayudando a fomentar la cooperación. Estos parámetros son configurables, por lo que podrán ajustarse a valores óptimos a medida que se analice su funcionamiento en la implementación real.

2.4.2. El Consumo de Batería

Se proponen aquí tres niveles de batería para la decisión de si cooperar o no en la retransmisión de un paquete. Dichos niveles pueden ser considerados y establecidos por el nodo y en el caso de la implementación realizada han sido definidos en: alto (66 %), medio (46 %) y bajo (26 %). Cuando la batería alcance el nivel seleccionado por el nodo, que llamamos Nivel de Batería Límite (NBL) se estipula que el nodo deja de cooperar en la retransmisión de paquetes y demás operaciones de la red para preservar ese nivel de batería para operaciones propias. Con esto se pretende que el número de nodos que utilicen la red

sea máximo, pues cuanto mayor es el número de nodos mejor será el funcionamiento de la misma.

Se han realizado simulaciones con el fin de obtener datos a gran escala de cómo afecta el nivel de batería de los nodos a la red. A continuación se muestran los resultados de tres tipos de simulaciones diferentes donde todos los nodos presentan en el primer caso niveles de batería altos establecidos aleatoriamente entre el 75 % y el 100 % de la batería, en el segundo caso niveles de batería medios, entre el 46 % y el 75 %, y finalmente niveles de batería bajos entre el 25 % y el 46 %. Además los nodos presentan diferentes NBLs distribuidos uniformemente acorde a los tres niveles definidos anteriormente. Durante su vida en la red, los nodos pierden batería de forma que cuando los nodos alcancen su NBL dejan de cooperar. El objetivo de esta simulación es analizar el comportamiento de la red en diferentes casos. En el mejor de ellos, la red está formada por nodos que tienen sus niveles de batería muy altos y en el peor, los nodos están a punto de quedarse sin batería, considerando también un estado intermedio.

La Figura 2.3 muestra la relación entre el tiempo de simulación sin recargar batería y el porcentaje medio de éxito en el intercambio de paquetes teniendo en cuenta los tres niveles iniciales de batería detallados anteriormente. Los principales parámetros de estas simulaciones fueron 100 nodos, los cuales realizan 1000 conexiones de manera aleatoria cada 15 minutos. Para cada caso se han distribuido uniformemente los niveles de batería fijados por el nodo de la forma siguiente:

- 33 nodos fijan su NBL en alto, es decir, dejan de cooperar cuando tengan el 66 % de batería,
- 33 nodos fijan su NBL en medio, es decir, dejan de cooperar cuando tengan el 46 % de batería,
- 34 nodos fijan su NBL en bajo, es decir, dejan de cooperar cuando tengan el 26 % de batería.

Como se puede observar en la Figura 2.3, el número de conexiones exitosas disminuye a medida que el nivel de batería de los nodos decrece, siendo el peor caso, aquel en

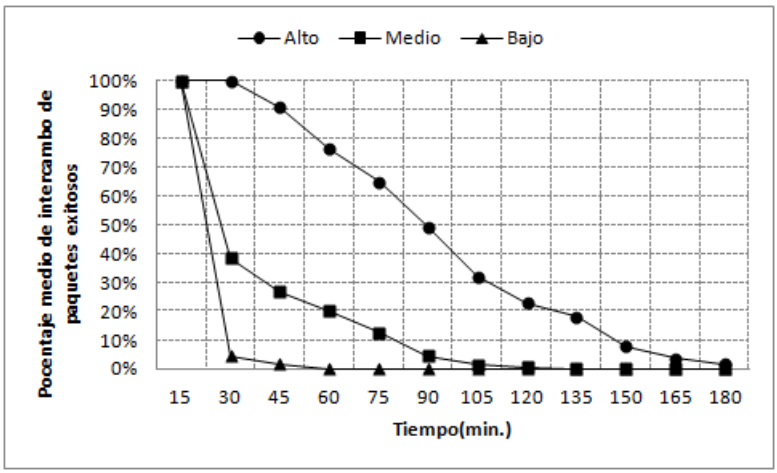


Figura 2.3: Consumo de Batería

el que todos los nodos presentan niveles iniciales de batería muy bajos. Teniendo en cuenta en ese caso que 66 nodos tienen los niveles de energía por debajo del NBL establecido por lo que directamente dejan de cooperar, los resultados son bastante buenos. Por tanto en caso de que los usuarios detecten un consumo excesivo de batería, pueden fijar sus niveles límite evitando comportamientos pasivos o egoístas y por tanto mejorando la cooperación en la retransmisión de paquetes.

2.5. Estructuras de Retransmisión

2.5.1. Estructura de Árbol

El primer sistema propuesto para la retransmisión de paquetes se basa en una estructura de árbol que permita definir los incentivos para premiar el comportamiento de cada nodo en función de su nivel de participación en el proceso de retransmisión. La Figura 2.4 muestra un ejemplo de retransmisión de paquetes en VANETs según una estructura de árbol, aquí llamado *árbol de retransmisión*. En esta figura se presentan esquemáticamente varias características importantes del árbol de retransmisión de un paquete:

1. El nodo raíz se corresponde con el origen que genera y envía el paquete.

2. Cada vehículo intermedio que retransmite el paquete se corresponde con un nodo

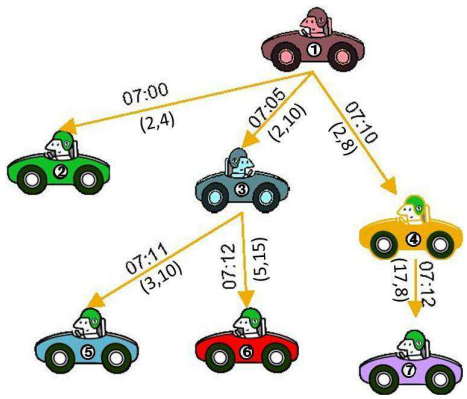


Figura 2.4: Árbol de Retransmisión de un Paquete

intermedio en el árbol.

3. Los nodos ignoran todos los paquetes que ya hayan recibido con anterioridad. En consecuencia, cada vehículo está presente sólo una vez en cada árbol de retransmisión.

4. Cada arista del árbol se corresponde con un encuentro de dos vehículos en la red en el que realizan la transferencia de información, y por ello se le asocia una marca de tiempo (timestamp) y las coordenadas espaciales indicando la posición de los vehículos en el momento del encuentro.

De acuerdo con el paradigma "almacenar y llevar", si un vehículo almacena un paquete por un largo periodo de tiempo y lo retransmite activamente a otros vehículos, el paquete tendrá más probabilidades de llegar a la destino previsto, o llegar a más destinos, en función del objetivo específico del paquete. Por lo tanto, combinando el tiempo de almacenamiento y el número de retransmisiones, podemos definir una métrica de la contribución de los vehículos intermedios de cada árbol de retransmisión. A fin de estimular a los vehículos intermedios en la retransmisión de paquetes, el vehículo o nodo origen o bien los nodos finales del árbol de retransmisión deben recompensar a cada vehículo intermedio de acuerdo con su contribución y según el tipo de paquete retransmitido. En el caso de paquetes de seguridad vial, deben ser los nodos finales quienes incentiven al resto mientras que en paquetes de valor añadido, como por ejemplo de publicidad, debe ser el nodo raíz el que recompense a los nodos retransmisores.

El modelo básico en el que cada nodo intermedio i recibe una recompensa constante Q presenta un problema de gasto excesivo ya que no se puede adivinar de antemano el gasto total porque el número de nodos en el árbol no se puede predecir fácilmente. Tal problema podría resolverse manteniendo constante la recompensa total Q y calculando la recompensa asociada a cada nodo intermedio Q_i de acuerdo con la siguiente fórmula en la que C_i es la contribución de cada nodo i:

$$Q_i = \frac{Q \cdot C_i}{C} \; donde \; C = \sum_i C_i \tag{2.1}$$

En un principio, la contribución C_i de un nodo i en el reenvío de un paquete puede ser modelada como una combinación lineal convexa entre el número de reenvíos n_i y el período de almacenamiento del paquete t_i siendo α un factor entre 0 y 1:

$$C_i = \alpha t_i + (1 - \alpha) n_i. \tag{2.2}$$

Cada nodo i que haya participado en la retransmisión de un paquete debe informar a los beneficiarios finales de la información sobre su contribución C_i según la fórmula 2.2. La contribución final C se calcula mediante la suma de la contribución parcial de cada nodo en el árbol de transmisión de forma que cada nodo intermedio recibe Q_i como recompensa por el reenvío.

Como se explicó anteriormente, la contribución C_i se calcula básicamente en función del número de reenvíos y el tiempo de almacenamiento del paquete. Teniendo solo esto en cuenta, el modelo propuesto no puede ser considerado una buena solución porque los nodos egoístas podrían preferir mantener el paquete en lugar de retransmitirlo ya que no se sabe de antemano cuánto se puede ganar por la retransmisión, y/o retransmitirlo puede implicar tener que compartir más la recompensa.

En la propuesta descrita a continuación, introducimos varios parámetros a considerar a la hora de ofrecer una recompensa. A fin de evitar que los nodos prefieran mantener el paquete en lugar de retransmitirlo, se propone considerar como nuevo parámetro el tiempo límite de vida del paquete en función de las características de la información que contiene. Si la información que se envía es de valor añadido como por ejemplo de acceso a Internet o de publicidad, el plazo será más largo, mientras que si la información está relacionada con

la seguridad vial, el plazo en general será más corto. A continuación se presenta la notación utilizada para describir los parámetros utilizados en el cálculo de la recompensa en este nuevo modelo:

- T_j: Tiempo límite de entrega del paquete j.
- t_{ij}: Tiempo durante el cual el nodo i almacena el paquete j.
- n_{ij}: Número de retransmisiones realizadas por el nodo i del paquete j antes de su tiempo límite T_j.
- Factor de equilibrio α

Si la tasa entre el tiempo t_{ij} y el tiempo límite T_j, t_{ij}/T_j, se considerara como factor multiplicador, la contribución aumentaría a medida que t_{ij} se acerca a T_j, y una vez el tiempo t_{ij} superara el tiempo límite T_j, la contribución continuaría creciendo, incluso más rápido dado que el factor proporcional sería mayor que 1. Este efecto se debe corregir de forma que cuando el tiempo durante el cual se almacena el paquete supera el tiempo límite, la contribución del usuario deje de crecer. Por ese motivo, usando nociones funcionales de límites y asíntotas a continuación se propone el uso de la función exponencial $f(x) = e^{-x}$ que tiene una asíntota horizontal en $y = 0$ cuando x tiende a ∞. Adicionalmente, se propone la incorporación de un nuevo parámetro que también se verá afectado por la utilidad del paquete dependiente de la distancia tanto para el nodo de origen como para el nodo retransmisor. En particular, se usará la siguiente notación adicional en el cálculo de la recompensa según la nueva propuesta:

- d_{ij}: Distancia entre el nodo origen y el nodo destino del paquete, cuando el paquete j es transmitido a través del nodo i.
- D_j: Distancia máxima del radio donde la información del paquete j es considerada de interés por los receptores.

Cada uno de los tres parámetros considerados en la nueva función convexa tiene tres factores de equilibrio entre 0 y 1, representados por α_1, α_2 y α_3. El valor que se asigna a

cada α depende de la relevancia que el nodo origen asigne a cada componente representado en la función de la contribución del nodo i en la retransmisión del paquete j. Así mismo en la contribución se incorpora una constante k positiva que determina la velocidad de acercamiento a la asíntota.

$$C_{ij} = \alpha_1(1 - (k \cdot e)^{-t_{ij}/T_j}) + \alpha_2 n_{ij} + \alpha_3((k \cdot e)^{-d_{ij}/D_j}) \; donde \; \sum_{k=1}^{3} \alpha_k = 1 \tag{2.3}$$

En la Figura 2.5 se observan superpuestas las gráficas correspondientes a la aportación a la contribución por parte del tiempo (gráfica creciente) y de la distancia (gráfica decreciente). Esto se debe a que la contribución crece cuanto más tiempo (dentro del límite) sea almacenado el paquete, pero decrece cuanto más se aleje el nodo retransmisor del lugar de origen del paquete.

A continuación se detalla cada parte de esta función y se justifica la utilización y repercusión que tienen en la nueva función de contribución propuesta.

Como se mencionó anteriormente, el tiempo es uno de los parámetros más importantes cuando se trata de asegurar que un paquete llegue al destino deseado. Si un vehículo almacena un paquete por un largo tiempo, se podría considerar que dicho paquete podría alcanzar a un gran número de vehículos. Sin embargo, este parámetro puede producir un comportamiento egoísta ya que los nodos podría preferir no retransmitir el paquete y de esta manera no compartir la recompensa final con los nodos de transmisión. Este efecto se puede evitar considerando en la función de contribución la siguiente fórmula:

$$(1 - (k \cdot e)^{-t_{ij}/T_j}).$$

Esta función tiene un comportamiento asintótico. Permite establecer el tiempo máximo T_j que un nodo debe almacenar un paquete ya que como se puede observar, el valor de la contribución se incrementa cuando aumenta el tiempo y cuando t_{ij} alcanza el umbral T_j el valor de contribución deja de aumentar con tanta rapidez ya que se encontrará más cerca de la asíntota $y = 1$.

Por otra parte, el objetivo del tercer término de la función de contribución es la evaluación del efecto de la distancia en la recompensa compartida según una fórmula similar

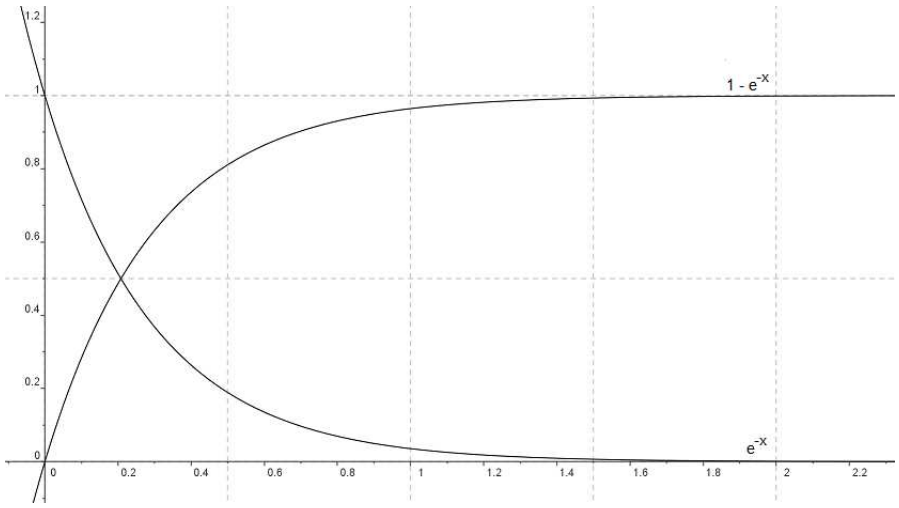


Figura 2.5: Contribución Versus Tiempo y Distancia

a la del tiempo pero con un efecto inverso. Este término se reduce cuando los vehículos se alejan demasiado del nodo origen, de forma que a partir del radio de interés la contribución es prácticamente nula ya que la gráfica decreciente se acerca a la asíntota $y = 0$.

Obsérvese que, dado que la contribución C_{ij} se calcula solo cuando $t_{ij} \leq T_j$ y $d_{ij} \leq D_j$, el efecto asintótico de la fórmula produce que los vehículos más cercanos al evento y que han almacenado el paquete por más tiempo, dentro de esos límites, son los que más recompensa reciben.

Finalmente, el segundo término de la métrica de contribución propuesta está relacionado con el objetivo prioritario de esta sección. Se trata de la medición del número de retransmisiones del paquete por cada nodo intermedio, y por lo tanto de su cooperación. En este caso no se fija ninguna restricción ni límite en cuanto a valores máximos o mínimos ya que la contribución del nodo i aumenta directa y proporcionalmente con el número de retransmisiones del paquete j. De acuerdo con este factor, cuanto más colaboren los vehículos en el envío de paquetes, más grande será su contribución final. En la función propuesta, este parámetro es el que más rápido incrementa la contribución antes de alcanzar el tiempo y distancia límite. En consecuencia, el factor de equilibrio α_2 debe ser mayor que los otros dos factores con el fin de fomentar la cooperación en la transmisión de paquetes.

Con el fin de hacer un estudio de la propuesta de fomento de la retransmisión de paquetes según la estructura de árbol con medida de contribución dada por la fórmula 2.3, se han realizado varias simulaciones de VANETs implementadas en NS-2. Los parámetros de las simulaciones realizadas son los siguientes: 15 nodos colocados al azar en un área de 800m x 800m. El radio de acción de cada nodo es de 100 metros. En cada simulación se elige un nodo al azar, que envía un paquete a todos los nodos vecinos que estén dentro de su radio de acción. En la Figura 2.6 se analiza la relación obtenida entre las recompensas y los parámetros tiempo y espacio de la función de contribución. De acuerdo con la gráfica del tiempo, el esquema asigna mayores recompensas a aquellos nodos que almacenan los paquetes por más tiempo. Sin embargo, hay que tener en cuenta que las recompensas están influenciadas por el número de retransmisiones y la distancia entre el nodo origen y los nodos que retransmiten el paquete. Por otra parte, en la gráfica de distancia el esquema parece dar recompensas más bajas a aquellos nodos cuya distancia es mayor con respecto a la posición inicial del nodo origen. Sin embargo pueden observarse casos donde algunos nodos a una gran distancia, obtienen un promedio de recompensa mayor. Esto se debe a la influencia de otros parámetros como el número de retransmisiones debido a que en el mismo momento, distancia y número de retransmisiones pueden ser mayores por haberse encontrado con más nodos al alejarse del evento. En particular, el parámetro nº de retransmisiones es el más influyente, tal como se observa en a Figura 2.7, pues los nodos con mayores contribuciones, son generalmente los nodos que tienen más descendientes en el árbol de retransmisión. Como era de esperar en las retransmisiones, el promedio de recompensa aumenta en función del número de retransmisiones con lo que se obtiene una gráfica muy parecida a la del tiempo. Se puede concluir que la fórmula 2.3 propuesta asigna una mayor recompensa a los nodos que efectivamente retransmiten más paquetes durante mayor tiempo.

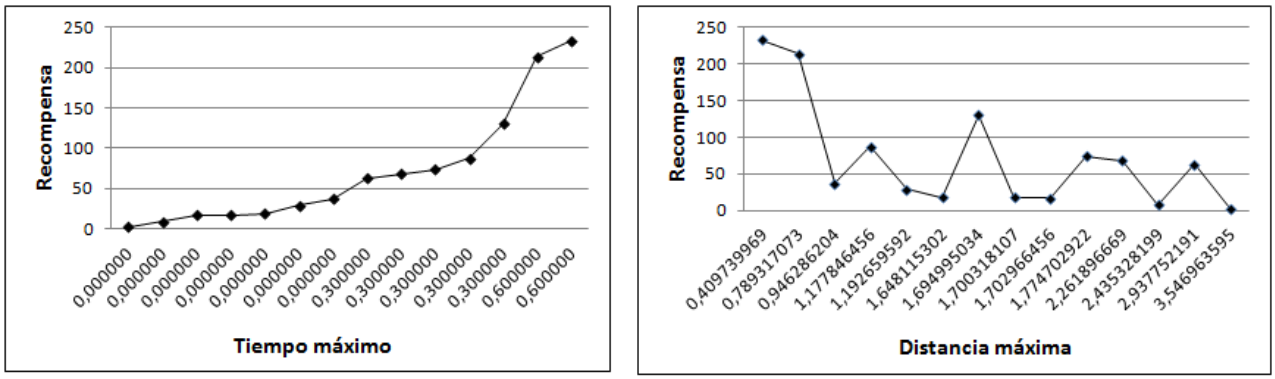


Figura 2.6: Resultados de Simulaciones en NS-2

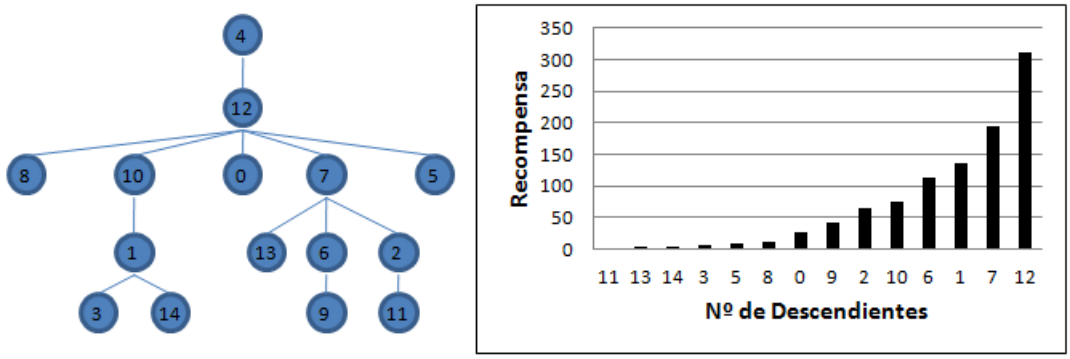


Figura 2.7: Recompensa Versus Descendientes en el Árbol de Retransmisiones

2.5.2. Estructura de Grupo

Llamamos aquí grupo a un conjunto de vehículos que se encuentran en un área geográfica cercana y dentro de la cobertura de al menos uno de ellos. Entendemos que cada grupo tiene un número mínimo de vehículos y es controlado por uno de los nodos, que se conoce como *líder* del grupo, el cual permite hacer más simple y ligera la gestión del grupo y las retransmisiones de los paquetes. El líder de grupo será el encargado de gestionar la información y las retransmisiones que se establecerán dentro de su grupo, delegando trabajo a otros vehículos si fuese necesario. En la versión más reducida de grupo se supone que todos los nodos que forman parte del grupo tienen conexión directa, es decir que la distancia entre ellos es de uno, lo que se corresponde con el número de saltos necesarios para comunicarse. En este trabajo supondremos que cada nodo dispone de un par de claves publica/privada y

que todos los nodos de cada grupo comparten una clave secreta con el líder del grupo. En la fase de creación del grupo, cada nodo envía como respuesta a una petición de formación de grupo la aceptación con su clave pública. El líder del nuevo grupo después del proceso de autenticación envía a cada nodo la clave secreta del grupo cifrada con la clave pública de cada nodo.

La gestión de grupos debe cumplir dos requisitos: minimizar el consumo de recursos y en particular el intercambio de mensajes, y tener en cuenta la topología altamente dinámica de la red, requiriendo actualización de la pertenencia al grupo. En particular, los vehículos forman grupos según celdas dinámicas donde el líder es el vehículo que haya iniciado la formación del grupo. Esta definición será la estructura básica sobre la que se sostiene gran parte del capítulo 3 y será explicada detalladamente más adelante.

No es difícil suponer que ningún nodo querrá ser líder de su grupo porque el número de paquetes que tiene que manejar en ese caso es mayor que siendo un nodo cualquiera del grupo. Sin embargo, luego se propondrán varios mecanismos para fomentar la cooperación según el tipo de paquetes, concluyéndose que en todos los casos, el líder recibe una recompensa por su cooperación mayor que cualquier otro nodo del grupo, lo que debe animarle a jugar el papel de líder.

2.6. Cooperación en Retransmisión de Paquetes de Valor Añadido

El principal objetivo de una VANET es mejorar la seguridad, eficiencia y confort en la carretera, utilizando sistemas que aumentan la seguridad de los pasajeros mediante el intercambio de información entre vehículos. Las aplicaciones relacionadas con la seguridad vial son las más importantes en este tipo de redes. Sin embargo, su estructura permite proporcionar otros servicios como el acceso a Internet o el intercambio de publicidad. En esta sección presentamos propuestas basadas en incentivos según los diferentes tipos de paquetes de valor añadido que se pueden retransmitir en este tipo de redes.

Los esquemas basados en incentivos consisten en recompensar a aquellos nodos que participan activamente en los servicios de la red, siendo el nodo beneficiario de dicho servicio

el encargado de pagar por él. En la propuesta presentada más adelante en esta sección distinguimos dos tipos de paquetes diferentes para los cuales se describen diferentes esquemas de incentivo de manera que se ajusten lo más posible a su objetivo y funcionamiento:

- Paquetes de Internet: Este tipo de paquete proporciona acceso a Internet a través de nodos intermedios.
- Paquetes de Publicidad: Este tipo de paquete proporciona información comercial y publicidad sobre establecimientos como tiendas, restaurantes, etc, normalmente localizadas geográficamente cerca del vehículo.

2.6.1. Paquetes de Internet

Para la retransmisión de paquetes de Internet, se asume que un usuario A ha realizado un contrato para dicho servicio de antemano con una operadora de Internet. Por lo tanto, se asume la existencia de: un nodo destino de la retransmisión que es el usuario A que ha contratado un servicio, un nodo origen que es la RSU a través de la cual la operadora se encarga de proporcionar dicho servicio, y un conjunto de nodos intermedios que son los vehículos que llevan a cabo la conexión entre el origen y el destino.

Cuando el nodo A quiere disfrutar de la conexión a Internet, tiene que establecer una sesión punto a punto con una RSU. Con el fin de establecer dicha sesión, A genera un mensaje de petición y lo retransmite a través de la red. El esquema de cooperación propuesto se basa en que cada nodo que recibe este paquete introduce su pseudónimo en el paquete y lo retransmite según uno de los modelos descritos. El objetivo de que cada nodo intermedio incluya su pseudónimo en el paquete es que la operadora sepa la identidad de los nodos cooperativos y pueda recompensarlos sin que los otros nodos retransmisores puedan descubrirla. Si se usa para la retransmisión la estructura de grupo propuesta en la sección anterior, A envía la petición al líder del grupo quien lo reenvía a los nodos de su grupo que estén en contacto con otros grupos, repitiéndose este proceso hasta que la petición llegue a la RSU. Entonces ésta responde con una confirmación de establecimiento de sesión, que devuelve al nodo A a través de la misma ruta. El proceso definido anteriormente puede verse esquemáticamente en la Figura 2.8. La sesión de Internet entre el nodo A y la RSU se activa

cuando ambos reciben el mensaje de confirmación. En caso de utilizarse una estructura de árbol para la retransmisión, sólo aquellos nodos intermedios correspondientes a la ruta más corta entre A y RSU participarán activamente en el establecimiento de la sesión, por lo que serán los únicos en recibir recompensa.

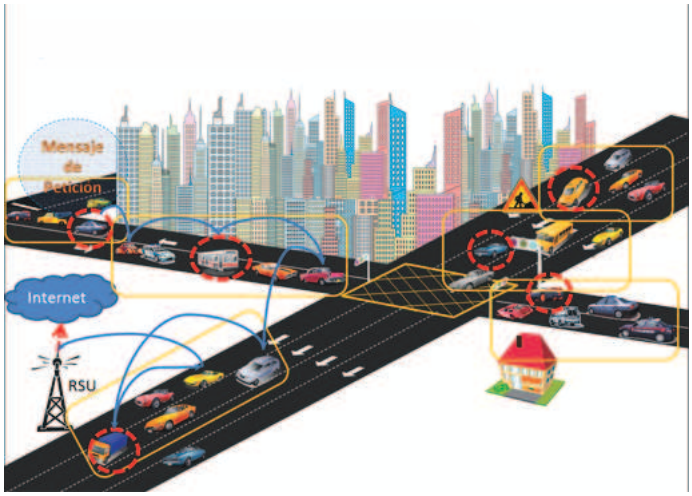


Figura 2.8: Paquetes de Internet

Para llevar a cabo la conexión, se distinguen dos tipos de paquetes diferentes:

1. *Paquetes de sesión:* Enviados cuando un nodo A quiere conectarse a Internet y tiene que encontrar una ruta hasta una RSU que ofrezca este servicio, para establecer una sesión. Cuando el nodo no tiene conexión directa con la RSU, debe encontrar una ruta a través de nodos intermedios.

2. *Paquetes de datos:* Una vez establecida la conexión, los paquetes de datos se retransmiten a través de la ruta descubierta, sin tener que pasar por el nodo líder en el caso de que se use la estructura de grupo. De esta manera, el líder no tendrá que recibir todos los paquetes de datos y puede centrarse en sus acciones como líder.

El operador debe distinguir entre paquetes de sesión y paquetes de datos a la hora de recompensar a los nodos porque el número de paquetes necesarios para el establecimiento de sesión es en general menor que el número de paquetes de datos, pero la recompensa debe ser comparativamente mayor porque sería imposible conectar el nodo origen a la RSU sin

establecer una sesión. En particular, no será rentable para el operador pagar una cantidad que solo dependa del número de paquetes retransmitidos. Para solucionar este problema se propone que los nodos retransmisores de paquetes de datos reciban una recompensa q significativamente menor que 1 ($0 < q << 1$) por cada paquete transmitido. Un nodo puede estimar el número n_d de paquetes de datos que puede transmitir antes de perder la conexión con una RSU y así calcular la recompensa esperada:

$$Q_d = q \cdot n_d \tag{2.4}$$

La recompensa por los paquetes de establecimiento de sesión debe ser un valor Q_s mayor que la recompensa por los paquetes de datos Q_d. Cuando se establece una sesión para realizar según la estructura basada en grupos una conexión a Internet, el líder del grupo es el encargado de encontrar la mejor ruta entre los nodos de su grupo y la RSU. Una vez establecida esta conexión, el nodo líder ya no formará parte de la retransmisión de paquetes de datos. Sin embargo, si el líder decidiera no establecer la sesión de conexión, la comunicación sería imposible. Por lo tanto, la recompensa total recibida por el líder por la retransmisión de paquetes de sesión será Q_s, que como ya se dijo, es superior a la recompensa Q_d establecida para los paquetes de datos. De esa forma, el líder del grupo se sentirá motivado a cooperar y cumplir su función.

2.6.2. Paquetes de Publicidad

La difusión de publicidad a través de la comunicación V2V es la tercera utilidad mencionada en este tipo de red. En este caso, el proveedor envía los anuncios comerciales, y los vehículos cercanos que los reciben comienzan a difundirlos a otros vehículos mientras están en movimiento. Estos anuncios se transmiten durante un cierto período de tiempo y distancia respecto al proveedor origen. Inspirado en un esquema de micro-pagos se propone que cada pago por servicio de retransmisión de paquetes de publicidad sea como un billete de lotería. Al recibirlo, tanto el anunciante como el vehículo ganador de la lotería puede determinar si se trata de un billete ganador o no. El anunciante no sólo pagará al vehículo con el billete ganador, sino también al vehículo que recibe el paquete retransmitido como

veremos. Este esquema permite a los proveedores determinar de antemano la recompensa o gasto por la retransmisión de paquetes, lo que es una ventaja porque de lo contrario podría tener un problema de exceso de gasto.

Para este tipo de paquetes, el proveedor de publicidad envía anuncios comerciales a todos los vehículos que se encuentran en su rango de aplicación. En este caso, el número de paquetes generados es en general mayor que los paquetes de Internet, ya que el objetivo de estos paquetes es ofrecer publicidad a tantos vehículos como sea posible por lo que la retransmisión es vía broadcast pura mientras que en el caso de los paquetes de Internet es necesario enrutamiento multihop. Aunque el número de paquetes generados es mayor en publicidad que en Internet, tanto en el caso de usar retransmisiones según la estructura de árbol como en grupo, se debe lograr minimizar el número de retransmisiones repetidas. En el caso de usar grupos, el líder recibe el paquete de publicidad y es responsable de su difusión dentro de su grupo. El resultado es el mismo que cuando se usa un árbol de retransmisión y es que ningún nodo retransmite el mismo paquete de sus nodos vecinos, porque si el líder recibe un paquete que ya ha sido retransmitido dentro del grupo, no lo retransmite de nuevo a su grupo. Por otra parte, el líder debe buscar dentro de su grupo a los nodos que le permitan enviar el paquete a los grupos vecinos tal como se muestra en la Figura 2.9. Así, se consigue reducir tanto el número de retransmisiones entre los grupos como el número de retransmisiones en el interior del grupo.

El modelo propuesto para el fomento de la cooperación en la retransmisión de paquetes de publicidad es una especie de lotería en la que cada nodo tiene una probabilidad de ser ganador. Cada proveedor de publicidad genera un paquete que contiene un identificador único *PackID* concatenado con la información de publicidad $AdInformation$ y un resumen H:

$$[PackID\,|AdInformation|\,H]$$

Cuando un nodo N recibe el paquete, comprueba la información. Si N decide participar en la retransmisión, envía el mensaje a los nodos i hijos en el árbol de retransmisión, y espera de cada uno el correspondiente recibo rec_{N_i}. Después, el nodo N calcula para cada rec_{N_i} recibido el resumen de *PackID*, $NodeID_N$ y rec_{N_i}, mediante una función hash

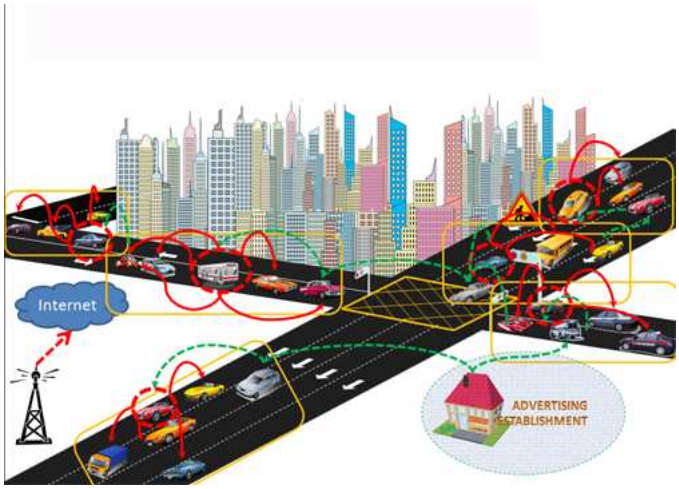


Figura 2.9: Paquetes de Publicidad

criptográfica h con probabilidad de colisión despreciable, y compara el resultado con H.

$$h(PackID\,|NodeID_N|\,rec_{N_i}) = H \tag{2.5}$$

Si la igualdad 2.5 se cumple con alguno de los recibos, entonces el nodo N es un ganador.

Además, un nodo también puede obtener una recompensa si como hijo envía el recibo ganador a su padre, para fomentar la devolución de recibos. Se denota como $Prob_h$ a la probabilidad constante de que el resumen mediante h del paquete concatenado incluyendo el recibo rec_{N_i} colisione con el valor H y por tanto produzca un premio.

$$Prob_h = Prob[h(PackID\,|NodeID_N|\,rec_{N_i}) = H] \tag{2.6}$$

Se asume que cada nodo puede recibir sólo una recompensa de sus hijos, por lo que la probabilidad $Prob_P$ de que un nodo retransmisor cualquiera gane un premio tras la retransmisión de un paquete a N_c nodos hijos, tras haber recibido el paquete de un número N_f de nodos padres, se puede definir como:

$$Prob_P = (N_c + N_f) \cdot Prob_h \tag{2.7}$$

Si analizamos la función de probabilidad en 2.7, se ve que cuando un nodo obtiene de uno de sus hijos un recibo ganador, se puede esperar que ya no transmita más paquetes, porque la función definida restringe a un solo premio debido a que la función hash criptográfica usada tiene probabilidad de colisión despreciable. Este comportamiento no se considera deseable ya que el objetivo es motivar a retransmitir el máximo posible los paquetes de publicidad. Para solucionar este problema se plantea el uso de una función hash con probabilidad de colisión no despreciable, lo que conduce a la posibilidad de que haya más de un recibo premiado por cada paquete:

$$\exists i \neq j : \ h(PackID\,|NodeID_N|\,rec_{N_i}) = h(PackID\,|NodeID_N|\,rec_{N_j}) = H \tag{2.8}$$

En este caso, un nodo podría ganar según la probabilidad anterior la lotería más de una vez con el mismo paquete de publicidad retransmitido y también más de una vez por los recibos que devuelva a sus padres. Por lo tanto, el problema de que una vez que un nodo recibe el premio, deje de retransmitir el mismo paquete queda resuelto con el uso de una función hash con colisiones. En el caso de la estructura de grupo el líder del grupo recibe siempre todos los paquetes de publicidad de su grupo para su redifusión luego tendrá una mayor probabilidad de obtener recibos de lotería premiados. De hecho en cada redifusión de un paquete de publicidad a los G nodos de su grupo, cuando se usan funciones hash con probabilidad de colisión no nula, la probabilidad de que el líder sea premiado sólo por las retranmisiones a su grupo es:

$$G \cdot Prob_h \tag{2.9}$$

Esta expresión supone un incentivo para el líder en la retransmisión de los paquetes de publicidad, porque cuanto mayor sea el número de retransmisiones a su grupo, mayor será la probabilidad de ganar al menos un premio. Por lo tanto el modelo motiva a todos los nodos tanto a cooperar como a querer ser líderes.

Para el anunciante será conveniente intentar reducir al mínimo la ocurrencia de colisiones en la función hash usada, ya que determina la probabilidad de premios. Sin embargo, no debe anularla ya que eso frenaría la retransmisión de sus anuncios. Por tanto,

el anunciante debe escoger cuidadosamente tanto la función hash como la recompensa para motivar a los nodos en su participación en la retransmisión, fijando dicha probabilidad de colisión en un valor lo suficientemente atractivo.

2.7. Cooperación en Retransmisión de Eventos en Carretera

En esta sección se propone un conjunto de contramedidas para prevenir tanto comportamientos egoístas como ataques maliciosos, haciendo uso de revocación de nodos a través de mecanismos de aislamiento de nodos maliciosos de la red.

El objetivo de detectar un ataque en la red es disminuir su efecto y evitar nuevos ataque en el futuro, aislando a los nodos atacantes, proporcionando información real y confiable sólo a aquellos vehículos que participan en el buen funcionamiento de la red. El esquema propuesto consiste en un sistema de revocación descentralizado mediante la cooperación de nodos, y en el consecuente aislamiento de los nodos maliciosos. Para ello se propone una solución basada en el uso de listas de reputación donde cada vehículo guarda unas listas que almacenan información acerca de aquellos vehículos que han presentado mal comportamiento.

2.7.1. Preliminares Criptográficos

Las necesidades criptográficas tales como la autenticación, integridad de datos, privacidad y confidencialidad son algunos de los aspectos más importantes en investigación sobre redes en general y VANETs en particular. Con el fin de alcanzar estos requerimientos, en esta sección se propone el uso de algunos mecanismos criptográficos conocidos, como son la firma digital basada en clave pública y la autenticación a través de pseudónimos para el fomento de la cooperación en la retransmisión de eventos en carretera.

Antes de empezar a formar parte de la red, cada nodo debe obtener de forma descentralizada un par de claves pública y privada. Para ello ejecuta un protocolo de intercambio de claves con uno o varios nodos legítimos de la red. Además, a cada nodo que empieza a formar parte de la red se le proporciona un pseudónimo que se asociará en todo momento con su comportamiento, ya sea cooperativo o egoísta, sin revelar su identidad.

Este alias se creará de manera automática por un generador a partir de su clave pública, mediante una función unidireccional inyectiva, que evitará la posible existencia de dos pseudónimos diferentes para la misma clave pública y por lo tanto la posibilidad de generar falsos pseudónimos o de hacerse pasar por otro nodo dentro de la red.

Por otro lado, cada nodo de la red debe poseer un almacén o repositorio de certificados de claves públicas el cual contiene las claves públicas de otros nodos de la red firmadas por nodos confiables de la misma. Cuando dos nodos se encuentran y quieren comunicarse, realizan un intercambio de sus claves públicas. Entonces, cada uno de estos nodos busca la clave pública proporcionada por su interlocutor en su almacén y si no existe una coincidencia, los nodos intercambian sus repositorios. Una vez intercambiados los almacenes, ambos nodos buscan una ruta común en el conjunto unión de ambos almacenes, tal y como propone [73] con el fin de encontrar una cadena de confianza entre ambos. En el caso de no encontrarla, los nodos no se autentican mutuamente ya que no pueden confiar el uno en el otro y por lo tanto, no pueden intercambiar más información. Es posible que durante el inicio del ciclo de vida de la red, la probabilidad de coincidencia entre dos almacenes sea muy pequeña, en cuyo caso será necesario definir niveles de seguridad más bajos que los supuestos. Sin embargo, cuando la red alcance un tamaño suficiente, teniendo en cuenta el experimento de "el mundo es un pañuelo" descrito en [111], estos niveles de seguridad pueden ir aumentando por la conocida propiedad de los "seis grados de separación" descrita en [111] y basada en la idea de que siempre es posible unir a dos personas cualesquiera en el mundo mediante una cadena de de menos de seis conocidos. Según este principio, la probabilidad de encontrar alguna coincidencia entre dos almacenes será muy alta tras un periodo de vida de la red, por lo que se puede considerar que es posible encontrar una cadena común en el almacén conjunto de dos nodos cualesquiera.

2.7.2. Detección de Comportamiento Egoísta

Un aspecto importante en la retransmisión de eventos en carretera es que siempre debe proporcionar información a tiempo real y confiable a los nodos que forman parte de la red. Para alcanzar este objetivo, por una parte se utiliza el fomento de la cooperación en retransmisión de paquetes mediante recompensas. Sin embargo es necesario proponer

también un mecanismo que permita identificar posibles comportamientos egoístas mediante castigos.

Para ello se propone a continuación el uso de los paquetes de control de recepción asociados a una marca de tiempo (timestamp), los cuales denominamos ACK (acrónimo del inglés, ACKnowledgement) y utilizamos para determinar si los nodos están cooperando o no con otros nodos de la red de forma continuada. La idea de los ACK es similar a la de los recibos rec_{N_i} de lotería descritos en la sección anterior, pero en este caso no hay efecto probabilístico sino probatorio de cuándo se realizó la última retransmisión.

En particular, cuando un nodo A recibe o produce algún paquete de información, antes de enviársela a otro nodo B, le pregunta a B sobre su cooperación en la red. El nodo B responde al nodo A mediante el envío del último paquete ACK que ha recibido. Si la marca de tiempo correspondiente al paquete ACK recibido de B por el nodo A excede de un umbral T definido por el protocolo en función del tamaño de la red, el nodo A decide no enviar la información al nodo B porque considera que el nodo no está cooperando activamente en la red últimamente. Por lo tanto, los nodos están motivados a cooperar para mantener actualizados sus ACKs.

La consecución de ACKs en una retransmisión se observa en el sistema cuyo proceso se detalla esquemáticamente en la Figura 2.10. El nodo A tiene un paquete de información M que desea enviar a B. Antes de enviársela al nodo B, A divide el contenido del mensaje M en dos partes (M_1, M_2) formando dos paquetes diferentes. El objetivo de dividir la información en dos paquetes diferentes es asegurar que el nodo A reciba al menos un ACK como prueba de su cooperación, antes de que B pueda conseguir el contenido completo del mensaje M, con el objeto de evitar el posible comportamiento pasivo de B en cuanto a la devolución del correspondiente ACK a A. Cuando el nodo B recibe de A la primera parte del paquete M_1, cifra con su clave pública $E_B(M_1)$, le envía un primer acuse de recibo cifrado con la clave pública de A y firmado con su clave privada $E_A(D_B(ACK_1))$. Cuando el nodo A, recibe este paquete envía a B la segunda parte del mensaje de nuevo cifrado con la clave pública de B, $E_B(M_2)$, con objeto de que solo B pueda descifrarlo. En este momento es cuando el nodo B puede recuperar el contenido completo del mensaje M. Finalmente, B envía a A el segundo acuse de recibo cifrado y firmado $E_B(M_2)$, confirmando así la recepción de la

segunda parte del paquete. Durante el proceso de intercambio de paquetes es posible que exista mal comportamiento por parte del emisor o/y del receptor, incumpliendo el proceso de intercambio de paquetes definido. Por una parte, es posible que después de recibir el primer ACK_1, el nodo A no envíe M_2. También es posible que el nodo B no envíe el segundo ACK_2 tras la recepción de M_2. En cualquiera de estos casos, el comportamiento egoísta debe ser reportado a través del mecanismo de reputación explicado a continuación.

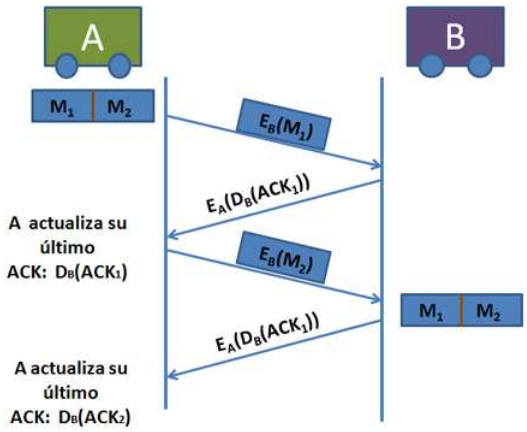


Figura 2.10: Envío de Paquete y Confirmación de Recepción

2.7.3. Aislamiento de Nodos Maliciosos

Como se discutió previamente, un aspecto vital del funcionamiento de la retransmisión de eventos en carretera es que los nodos no solo cooperen en el intercambio de paquetes con sus nodos vecinos sino que se comporten honestamente enviando información fiable a través de la red. Para cubrir esta necesidad, se propone a continuación el uso de listas de reputación. Cada nodo mantiene dos listas de reputación llamadas: Lista Individual de Reputación (LIR), en la cual cada nodo almacena información de sus propias experiencias con otros nodos que encuentra durante su vida en la red; y Lista General de Reputación (LGR), que contiene los pseudónimos revocados correspondientes a vehículos que han tenido mal comportamiento dentro de la red. El propósito de estas listas es excluir de la red a los nodos que presentan mal comportamiento, de forma que ese castigo motiva

a los nodos a contribuir en las operaciones de la red para no quedar aislados.

Las listas de reputación son actualizadas cada vez que se detecta que un vehículo intenta algún ataque y pueden ser modificadas durante la interacción con otros vehículos. También pueden ser actualizadas durante el intercambio de paquetes entre dos vehículos autenticados ya que además de intercambiar sus almacenes de claves también intercambian sus listas de reputación. Dichas listas permiten el aislamiento de los nodos maliciosos porque al tener las listas actualizadas, los nodos legítimos y con buena reputación en la red no intercambiarán información con aquellos nodos que estén presentes en sus listas.

Para actualizar estas listas es importante utilizar un proceso eficiente y basado en algoritmos rápidos de búsqueda. La Tabla 2.1 muestra los cuatro campos que proponemos como mínimo para estas listas. Cada registro de la lista debe contener el pseudónimo del vehículo que presenta mal comportamiento, para evitar toda conexión y transmisión de información con él. La fecha del mal comportamiento que también se incluye en el registro se utiliza para mantener la lista actualizada y borrar registros antiguos, lo que permite la reinserción a nodos que fueron maliciosos hace tiempo. Un tercer campo con la firma del nodo que presenta la denuncia se incluye en el registro para evitar falsas alegaciones. Finalmente, el campo de coordenadas geográficas (X,Y) donde se detectó el mal comportamiento proporciona una solución a algunos problemas que podría presentar esta propuesta, y que se detallan más adelante.

Tabla 2.1: Campos de las Listas.

Pseudónimo del nodo malicioso	Marca de Tiempo	Firma del nodo denunciante	Coordenadas (X,Y)

Lista Individual de Reputación (LIR)

La lista LIR permite a los nodos almacenar información sobre aquellos vehículos que han presentado mal comportamiento en cuanto a la cooperación durante el intercambio de información con acuse de recibo descrito. Además permite a los nodos tomar decisiones directas acerca de si cooperar o no con otros nodos en función de su experiencia personal porque la información que almacenan es completamente fiable. La LIR puede ser actualizada

si un nodo detecta un mal comportamiento por parte de otro nodo, bien porque genera mensajes falsos que no se corresponden con la información de su entorno, o bien, como vimos anteriormente, porque en el intercambio de mensajes con acuse de recibo descrito no cumple con su papel.

Por tanto, durante el intercambio de paquetes, si alguno de los dos nodos decide no retransmitir todos los paquetes necesarios que se detallan en la Figura 2.10 es introducido en la LIR del nodo con el que está haciendo el intercambio. Esto sucede por ejemplo si un nodo A no envía la segunda parte del paquete M_2 una vez el nodo B le ha enviado el primer ACK_1, o bien si el nodo B no envía alguno de los ACKs correspondientes a las diferentes partes del mensaje enviado por el nodo A.

La Figura 2.11 muestra el diagrama de flujo correspondiente al procedimiento ejecutado por un nodo A en el envío del paquete al nodo B mientras que la Figura 2.12 muestra el diagrama de flujo correspondiente al proceso ejecutado por el nodo B que recibe el paquete del nodo A y está a cargo de enviar los ACKs.

Según la LIR, los nodos tienen una visión individual y personal acerca de la cooperación de otros nodos en la red. Sin embargo ésta no les permite tener una visión general sobre la cooperación de otros nodos con los que no han interactuado. Si la VANET no fuera completamente distribuida y descentralizada se podrían utilizar las RSU para este propósito. Sin embargo en esta propuesta planteamos la descentralización y autoorganización de las VANETs, evitando la hipótesis de existencia de RSUs, lo que nos hace proponer una nueva herramienta que llamamos Lista General de Reputación (LGR).

Lista General de Reputación (LGR)

En ausencia de una autoridad central que se encargue de recoger información tanto de los nodos de la red como de la propia observación para generar una lista común que pueda retransmitir a todos los nodos, hay que buscar una propuesta totalmente distribuida y descentralizada. El objetivo de la lista LGR es que los nodos puedan compartir información sobre los nodos maliciosos dentro de la red, con el fin de mantenerlos fuera de la misma y de este modo disminuir la posibilidad de nuevos ataques. La revocación de los nodos maliciosos se debe llevar a cabo haciendo uso de la cooperación entre los vehículos de manera que los

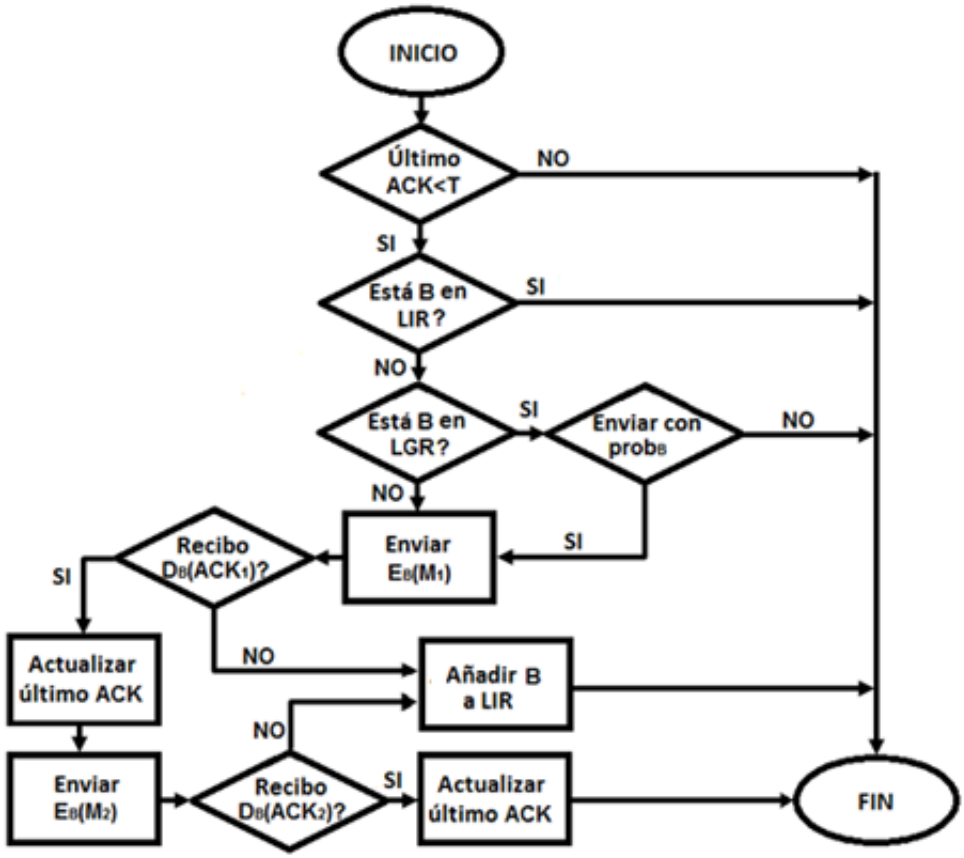


Figura 2.11: Diagrama de Flujo del Emisor

nodos puedan compartir y contrastar la información sobre los usuarios mal intencionados en la red.

La LIR, como vimos anteriormente, es almacenada y actualizada individualmente por cada vehículo. A diferencia de la LIR, el propósito de la LGR es que los nodos puedan hacer una puesta en común de la información almacenada en sus LIRs. En un primer intento por resolver este problema se propone que en cada encuentro los nodos además de intercambiar sus repositorios, intercambien sus LGRs. Por una parte y tras analizar la propuesta, se observó que presentaba varios problemas. El primero es la posibilidad de existencia de denuncias falsas. Con el objetivo de atacar a otro nodo, un nodo malicioso podría introducir en sus listas un nodo con buena reputación con el fin de compartir esta información con otros nodos y revocar al nodo honesto por una denuncia falsa. Para solucionar este problema

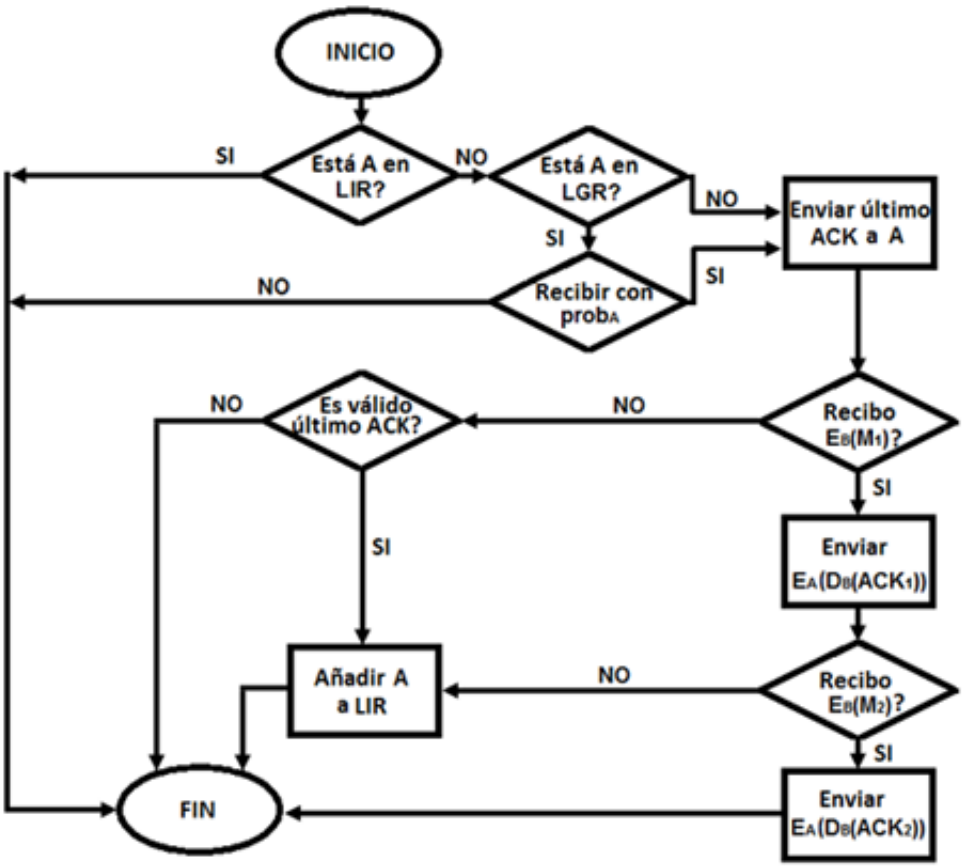


Figura 2.12: Diagrama de Flujo del Receptor

se propone definir un criterio para determinar si un nodo debe ser aislado o no de la red en función de su aparición en las LGRs de otros nodos. En este mecanismo se exigirá un número mínimo de denuncias antes de determinar que un nodo que aparece en una lista es realmente malicioso. Es decir, si un determinado número de nodos diferentes están de acuerdo en que un nodo en particular es malicioso, se entiende que se tienen evidencias suficientes de que la denuncia podría ser cierta. Según el experimento "6 grados de separación", en el caso de un nodo malicioso, al cabo de cierto tiempo de vida de la red no sea difícil encontrar más de una denuncia sobre el mismo nodo, dado que si un nodo intenta un ataque es muy probable que no lo intente en un solo sitio por lo que se recibirán varias denuncias de varios nodos acerca de su mal comportamiento en diferentes momentos y lugares. En este caso, cada nodo podría calcular una tasa de mal comportamiento c_i para cada nodo i

correspondiente al número de denuncias sobre el nodo i procedentes de diferentes nodos de la red. Cuanto mayor sea la tasa de mal comportamiento c_i, menor debe ser la probabilidad de enviar/aceptar paquetes de/para el nodo i. En particular, dicha probabilidad será:

$$prob_i = \begin{cases} 0, & si \quad i \in LIR \\ 1/c_i, & \forall \quad c_i \geq 1 \\ 1, & si \quad c_i = 0 \end{cases} \tag{2.10}$$

Dicha tasa de mal comportamiento $prob_i$ también determina la probabilidad de que el nodo i sea incluido en la LGR de forma que si la tasa supera un umbral predeterminado, el nodo se incluye en la LGR. Por otro lado surge el problema de posibles ataques en grupo donde varios nodos se pueden poner de acuerdo para introducir a un nodo correcto en su LGR con el fin de que otros nodos en la red lo revoquen y quede totalmente aislado. Para solucionarlo no solo será necesario denunciar a un nodo sino que las denuncias deben ser realizadas por los nodos en distintas coordenadas. De este modo se requiere por un lado que diferentes nodos realicen denuncias de un mismo nodo antes de tomarlo como malicioso, y por otro lado las denuncias se deben corresponder con diferentes coordenadas de manera que se demuestre que no se trata de denuncias falsas sino que el nodo realmente ha intentando atacar en diferentes puntos de la red.

Para alcanzar el objetivo de aislar a los nodos maliciosos de la red, se propone como solución combinar la cooperación de los nodos con la información contenida en sus LGRs además de mantener la propuesta de uso de las LIRs. Esta idea permite evitar conexiones con vehículos que no colaboran en el intercambio de paquetes, además de aislar a los nodos que intentan generar información falsa. Como se verá en el capítulo 3, el intento de generación de información falsa puede ser detectado y paralizado mediante un esquema de agregación. Como veremos, el esquema de agregación no permite llevar a cabo con éxito este tipo de ataques porque es fácilmente detectado por los nodos vecinos o por los nodos que reciben el paquete falso. Si los nodos detectan que un nodo está intentando atacar, lo normal es que este nodo quede aislado, para lo cual son útiles las listas de reputación propuestas.

2.7.4. Generación de Eventos en Carretera

En ausencia de una autoridad certificadora central, la revocación según las LGRs se lleva a cabo haciendo uso de la cooperación. Como se detallará en el capítulo 3, cuando un nodo detecta un evento o incidente, éste debe generar un mensaje de detección de evento con el fin de permitir a otros vehículos vecinos firmar si están de acuerdo con la información. Este mensaje M será firmado con la clave privada PrK_i del nodo i que genera el paquete y a continuación lo enviará a sus nodos vecinos. Si el vehículo i es un atacante, podría intentar enviar un mensaje con contenido falso (ver Figura 2.13A). En ese caso, en el supuesto de que se use la estructura de grupos los nodos que están dentro de su grupo y reciben ese mensaje serán capaces de detectar que se trata de un ataque tras comparar la información recibida en el paquete con la información de su entorno, e incluirán al nodo i en su LIR. El tiempo que permanece este registro en la LIR será menor que en el caso de comportamiento egoísta en el intercambio de paquetes, para permitir cierta flexibilidad frente a posibles errores que se mencionarán en la Sección 2.7.5.

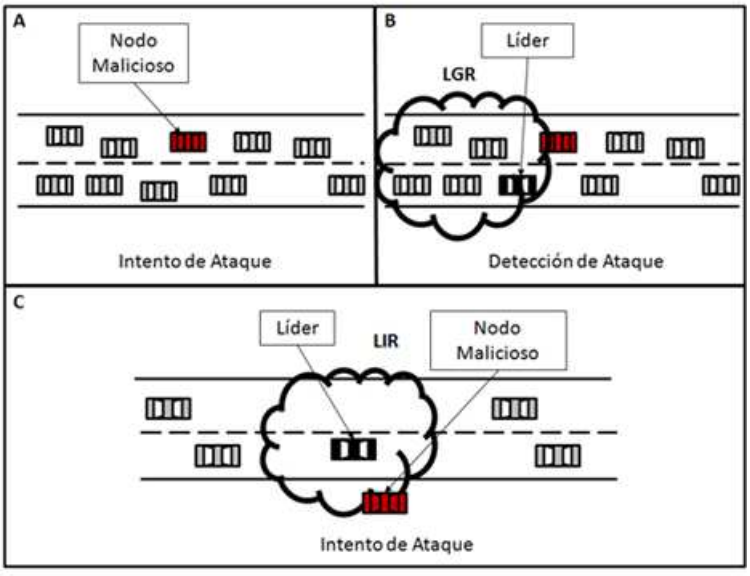


Figura 2.13: Ataque y Detección

Una vez detectado el mal comportamiento, los nodos vecinos que detectaron el ataque, forman un nuevo grupo cooperativo excluyendo al nodo atacante, tal como se

muestra en la Figura 2.13B. El líder de este nuevo grupo envía un mensaje de denuncia conteniendo la información falsa generada y firmada con la clave privada del nodo atacante. Los nodos del grupo que reciban ese paquete comprueban la información, y si están de acuerdo, firman la denuncia indicando que están de acuerdo con el hecho de que el nodo i es un atacante. Finalmente, el líder añade todas las denuncias firmadas recibidas de los nodos de su grupo en un paquete único, que envía a todos los nodos que están a su alcance. En el momento que los nodos reciben este paquete, incluyen al nodo i en su LGR y revocan su certificado borrando sus claves del almacén. Inicialmente los nodos que reciban el paquete falso incluirán al atacante en su LIR por si no se llegase a completar el proceso. De este modo se logra que al menos todos los nodos que recibieron la información tengan constancia directa del intento de ataque por lo que en un futuro no intercambiarán información con el nodo malicioso.

En esta propuesta es casi imposible para un atacante revocar a un nodo honesto pues es imposible que otros nodos tengan acceso a la clave privada de un nodo honesto para firmar la información falsa enviada. Sin embargo, es imposible asegurar que ante la detección del evento de una información falsa haya suficiente número de nodos para revocar al nodo origen debido a que no haya testigos suficientes del mal comportamiento y por tanto el número de denuncias no sea lo suficientemente fiable. Por lo tanto, cuando el número de denuncias no es suficiente pero un nodo detecta un ataque, lo único que puede hacer es introducir al nodo malicioso en su LIR, pero no en su LGR ver Figura 2.13C. Cuando un nodo intenta comunicarse con otro y éste está en su LIR, el nodo honesto rechazará directamente la comunicación.

Igual que en la propuesta descrita, en este esquema basado en grupos un nodo malicioso podría intentar revocar a un nodo correcto tras la generación de información real, alegando junto con otros nodos maliciosos que la información es falsa. A diferencia del caso anterior, donde se podían recibir denuncias de diferentes nodos en momentos diferentes, esta propuesta requiere que todos los nodos estén en las mismas coordenadas en el mismo momento para generar la denuncia, por lo que la estructura de esta propuesta de fomento de la cooperación en la generación de anuncios de eventos complica este ataque. Con el fin de hacer más fuerte el esquema, se requerirán como mínimo dos paquetes de denuncia

firmados por diferentes grupos, además de un mínimo de denuncias en cada paquete. Con esto se reduce la posibilidad de ataques dado que si se trata realmente de un nodo atacante y no legítimo, sería posible que intentara atacar en diferentes puntos por lo que se podría considerar la posibilidad de exigir que los dos paquetes además deban estar en coordenadas diferentes, proporcionando flexibilidad al esquema. Así, cuando un nodo recibe nuevas entradas para su LGR, debe comprobar antes las denuncias mediante comprobación de firmas de grupos de nodos diferentes en coordenadas diferentes. En caso contrario puede almacenar la denuncia en su LIR, hasta recibir una nueva denuncia de otro grupo, lo que determinaría la inclusión también en la LGR.

Cada vez que dos vehículos se encuentran, realizan un intercambio de repositorios y además intercambian sus LGRs. Una vez terminado el intercambio, cada uno actualiza su repositorio en función de la información recibida. De este modo, si un vehículo tiene información útil pero se encuentra con un nodo revocado, no enviará la información porque todo nodo revocado se encuentra aislado de la red.

2.7.5. Análisis de la Propuesta

En esta sección se realiza un breve análisis del funcionamiento de las propuestas descritas para el fomento de la cooperación en la retransmisión de eventos en carretera. En primer lugar hay que ser conscientes de la posibilidad de que un comportamiento honesto sea detectado como comportamiento malicioso por condiciones anómalas del tráfico o la carretera (por ejemplo si un nodo anuncia automáticamente una congestión en una autopista por encontrarse en un carril de desaceleración) o bien del propio vehículo (por ejemplo el caso de que un vehículo debido a un accidente, avería o conversación telefónica se pare en el arcén, y su comportamiento sea detectado automáticamente como posible congestión). En ambos casos los nodos que circulan normalmente por la vía pueden tomar a dichos nodos por atacantes, luego intentarán generar la revocación del vehículo. En este caso los usuarios insertarán a estos nodos en sus LIRs, pero dado que todas las denuncias se generarán en las mismas coordenadas, los nodos no serán revocados a menos que tengan varias denuncias de mal comportamiento en más de un lugar y momento y por parte de diferentes nodos. Sin embargo sí es cierto que dichos nodos aparecerán en las LIRs de los nodos que lo detecten,

luego es importante que el tiempo de expiración de este tipo de denuncias en la LIR sea menor que en las LGRs para que dichos nodos no queden indefinidamente denunciados. Estas y otras situaciones especiales se detallarán en el capítulo 3 centrado en este propósito.

Si bien los mecanismos de fomento de la cooperación descritos en esta sección para la detección de mal comportamiento no pueden resolver del todo este problema, sí logran incrementar el comportamiento honesto o cooperativo de los nodos tal como veremos en la sección de simulaciones. Por tanto, teniendo en cuenta la influencia del porcentaje de nodos egoístas en la operatividad de la red presentado al comienzo de este capítulo, podemos deducir que mantener o reducir el número de nodos maliciosos y egoístas beneficia mucho a la red.

Finalmente, otra situación que puede ocasionar problemas es el del uso de acuses de recibos ACKs como mecanismo de cooperación ya que cuando un nuevo nodo entra a formar parte de la red, no dispone de ningún paquete ACK porque no ha participado en ningún intercambio de información. En este sentido, el nodo no podría justificar su participación activa dentro de la red y por lo tanto no podía beneficiarse de la misma. Una posible solución es que el nodo que autentique a dicho nuevo nodo le entregue durante el control de acceso un ACK que le permita empezar a recibir información de la red. Otra opción es esperar hasta que el nuevo nodo genere su propio paquete de información, y después de compartirlo con otros nodos reciba el ACK por participar activamente en la red. La mejor opción depende de las condiciones específicas de la implementación de la propuesta.

2.7.6. Flexibilidad y Robustez

Un buen mecanismo para la detección del mal comportamiento y fomento de cooperación debe tener dos características: flexibilidad y robustez. En relación con la flexibilidad, es de destacar que un posible fallo en el hardware, software o una mala configuración del dispositivo, podría ocasionar que el sistema propuesto enviara mensajes con información incorrecta. Por lo tanto el sistema no debe ser demasiado estricto, permitiendo a los nodos recuperarse de denuncias erróneas. Además, sería conveniente permitir la reinserción de los nodos maliciosos si no presentan mal comportamiento en un período de tiempo. En este caso, los nodos maliciosos a reinsertar tienen dos posibilidades:

1. Obtener un nuevo par de claves y un nuevo pseudónimo de un nodo legítimo de la red. Cuando un nodo es marcado como atacante, es aislado de la red y no recibe ninguna información de tráfico. En caso de que se trate de un fallo o circunstancia excepcional del sistema y realmente no sea malicioso, el nodo podrá solicitar un nuevo par de claves. Antes de que el nodo los reciba, debe explicar y justificar la situación al nodo legítimo que le va a facilitar el nuevo par de claves si lo creyera oportuno.

2. Solicitar la comprobación del registro correspondiente. Si en las listas de reputación una denuncia es antigua, el nodo denunciado puede solicitar su eliminación para volver a ser admitido en la red. El nodo permanecerá en la listas de reputación por un periodo de tiempo que dependerá del grado de seguridad y tamaño de la red. Una vez este período expire, el nodo será borrado de la lista y podrá unirse nuevamente a la red con sus credenciales. Otro caso en que se debe borrar un registro de la lista es cuando se detecta que un nodo denunciado ha obtenido una nueva credencial, lo cual significa que algún nodo legítimo ha confiado en él, lo que podría interpretarse como prueba de su honestidad.

La robustez del mecanismo propuesto asegura que la información que llega a los nodos es cierta, lo cual evita que los nodos puedan suplantar a otros nodos enviando información falsa de su parte, o modificar información retransmitida. Para asegurarlo, cada vehículo intermedio puede determinar si la información generada por el nodo origen ha sido modificada o no. Esto es posible porque cada vez que un nodo origen genera un paquete, aplica una función hash al contenido del mensaje y lo firma con su clave privada, enviando además su clave pública para permitir a los receptores comprobar y leer el mensaje. Por tanto si el mensaje ha sido modificado en la transmisión, el nodo que lo reciba podrá comprobarlo a través de la firma digital. Además gracias a este mecanismo de detección, los nodos atacantes pueden ser aislados de la red, lo cual asegurará que la mayoría de los nodos que participan en la red son confiables y por lo tanto lo es también la información que ellos envían.

2.7.7. Simulaciones

En esta subsección se presentan los detalles y resultados de las simulaciones e implementaciones de los mecanismos presentados en esta sección en un entorno que se ajusta lo máximo posible a una VANET real, utilizando NS-2 como simulador de redes y SUMO como simulador de movilidad, tomando como punto de partida las simulaciones analizadas en [32]. Se han simulado en particular los mecanismos de listas LIR y LGR en un ambiente totalmente aleatorio, para analizar sus efectos en la red y el rendimiento de la cooperación resultante.

El objetivo de los mecanismos propuestos es detectar y aislar de la red aquellos nodos que presenten comportamientos maliciosos. Una simulación interesante ha consistido en determinar el tiempo necesario en una red para detectar todos los nodos maliciosos con el fin de aislarlos. En la primera simulación se usó un conjunto de 100 nodos con comunicaciones entre ellos generadas de forma totalmente aleatoria. Cada simulación se realizó 100 veces para cada uno de los diferentes porcentajes establecidos de nodos maliciosos. Las figuras muestran a continuación los resultados obtenidos en media en cada uno de los casos.

Cuando un nodo realiza una conexión con un nodo malicioso, éste lo incluye en su LIR evitando futuras conexiones con él. Por el contrario, cuando un nodo realiza una conexión con un nodo que no es malicioso, simplemente realizan un intercambio de sus LGR. Para incluir un nodo en la LGR se estableció en las simulaciones un mínimo de 3 denuncias diferentes sobre el mismo nodo antes de determinar que es malicioso. La Figura 2.14 muestra el tiempo necesario para que todos los nodos determinen quiénes son los nodos maliciosos de la red considerando diferentes porcentajes de nodos maliciosos en la red. Como se puede observar, a medida que aumenta el número de nodos maliciosos dentro de la red, el tiempo requerido para detectarlos decrece. Esto se debe a que cuanto mayor es el número de nodos maliciosos dentro de la red, mayor es la probabilidad de encontrarse con uno de ellos luego el número de denuncias sobre ellos aumenta. Por lo tanto, se puede decir que el mecanismo funciona mejor detectando a los nodos maliciosos, a medida que el número de nodos maliciosos aumenta.

De la simulación relacionada con la Figura 2.14 se deduce también que el caso

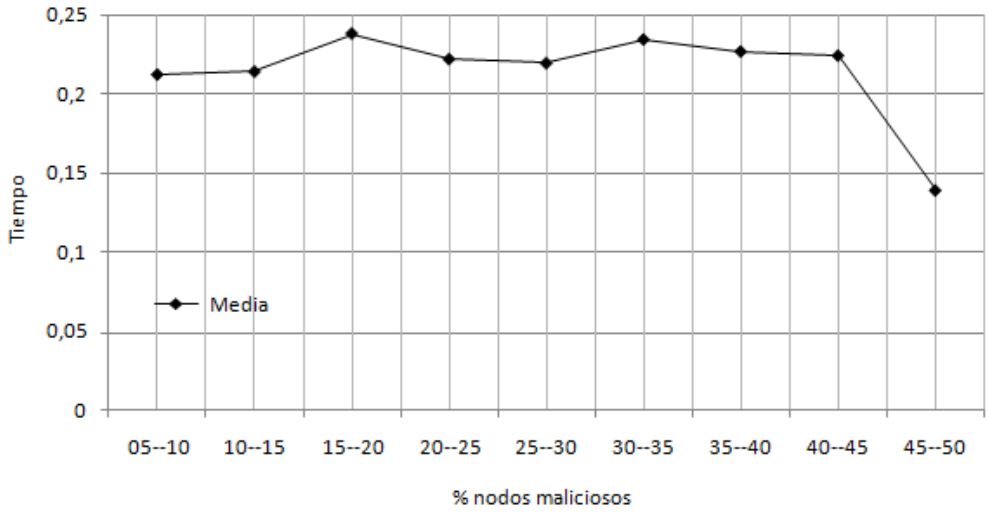


Figura 2.14: Tiempo de Detección de Nodos Maliciosos vs. % Nodos Maliciosos

en el que el método necesita más tiempo es aquél en el que el número de nodos maliciosos oscila entre el 15 % y el 20 % del total de nodos en la red. Tomando este dato como punto de partida se utilizaron para las simulaciones redes donde el número de nodos que la conforman oscila de 100 a 1000 nodos. En este caso el objetivo de la simulación consistió en determinar cómo influye el número de nodos que conforman la red en el tiempo necesario para aislar a todos los nodos maliciosos de la red. En este caso también los resultados son el promedio de 100 simulaciones para los diferentes tamaños de red utilizados. Como muestra la Figura 2.15, el tiempo para alertar a todos los nodos se incrementa a medida que aumenta el tamaño de la red. Sin embargo, los resultados también muestran que es posible aislar a los nodos maliciosos de la red en un tiempo razonable y con independencia del tamaño de la misma ya que el crecimiento observado en la gráfica es lineal.

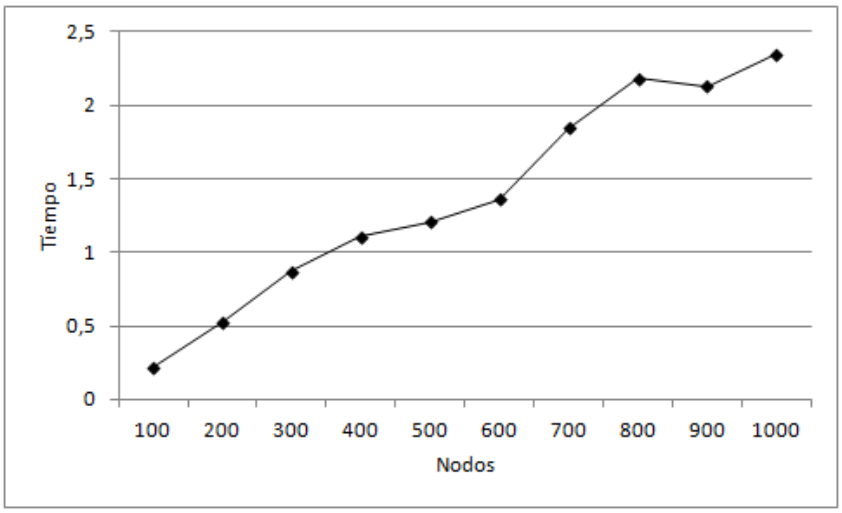


Figura 2.15: Tiempo de Detección de Maliciosos vs. Tamaño de Red

2.8. Captación de Usuarios

En esta sección se presenta una propuesta que tiene como objetivo motivar a los usuarios a entrar a formar parte de la red, evitando comportamientos pasivos de usuarios que tengan la intención de obtener información de la red sin cooperar en su funcionamiento. Como se ha presentado en este capítulo, la cooperación entre nodos es uno de los factores a considerar para el correcto funcionamiento de una red vehicular. Otro aspecto fundamental en las VANETs es que cuanto mayor es el número de usuarios que participan en ella mejor es su funcionamiento dado que permite ampliar la cobertura de la red. Sin embargo, las VANETs, tal y como se presentan en este trabajo son redes abiertas, en las que los nodos pueden en principio conectarse sin necesidad de ninguna clave con el fin de simplificar la conexión a la misma. Algunos usuarios expertos podrían aprovechar esta oportunidad para obtener información de la red actuando como nodos pasivos que solo escuchan los mensajes transmitidos. Sin embargo aunque no les sea posible el intercambio de mensajes, por no aparecer en el repositorio de ningún nodo, podrían facilmente llevar a cabo un ataque "man-in-the-middle" observando e interceptando mensajes entre dos víctimas sin que ninguna de ellas sepa que el enlace entre ellos ha sido violado.

Por lo tanto, en este tipo de redes, además de motivar a los nodos en la retrans-

misión de paquetes en beneficio de sus nodos vecinos, se debe asegurar que sólo los vehículos que pertenecen a la red y ayudan a su funcionamiento, se beneficien de la información que se transmite en ella. Para ello se propone utilizar el intercambio de datos cifrado como método para impedir el acceso a usuarios no autorizados fortaleciendo la entrada legítima de nodos en la VANET. En este trabajo se propone el cifrado simétrico como método más eficaz a la hora de establecer un canal de comunicación seguro para las VANETs porque en estas redes el tamaño de los datos transmitidos es generalmente muy grande. En particular, se propone un nuevo generador pseudoaleatorio para cifrar la información enviada. Este procedimiento evita que los nodos pasivos que no cooperan en la retransmisión de paquetes puedan obtener beneficios de la información enviada a través de la red.

Un generador de números pseudoaleatorios o PRNG (acrónimo del inglés, Pseudo Random Number Generator) es un algoritmo que produce secuencias de números cuyas propiedades se aproximan a las propiedades de los números aleatorios. Con el fin de ser considerado útil para criptografía, la secuencia resultante debe cumplir al menos tres propiedades: poseer un gran período, pseudoaleatoriedad y una complejidad lineal alta. En este trabajo se analizan estas propiedades para el PRNG propuesto.

A continuación se describe un cifrado en flujo que utiliza un nuevo PRNG cuyo diseño se basa en un filtrado no lineal de un registro de desplazamiento con realimentación lineal o LFSR (acrónimo del inglés, Linear Feedback Shift Register).

2.8.1. Preliminares Criptográficos

Cifrado en Flujo

El cifrado en flujo es un procedimiento de cifrado por sustitución que opera sobre el texto en claro símbolo a símbolo (sea éste un bit o un carácter), para obtener el texto cifrado. Habitualmente esta operación es un simple OR-exclusivo entre un símbolo en claro y uno de la clave (secuencia cifrante).

Para garantizar la seguridad perfecta de este tipo de cifrados, la secuencia de clave o secuencia cifrante tendría que ser totalmente aleatoria, tal y como ocurre en el cifrado de Vernam [135]. Sin embargo, debido a la utilidad que se le da a estos cifrados, es necesario

usar algoritmos determinísticos donde las mismas entradas del algoritmo siempre produzcan las mismas salidas. Así, la máquina interna que define el generador siempre pasa por la misma secuencia de estados definida por la entrada, y por tanto la salida nunca puede ser considerada verdaderamente aleatoria. Por lo tanto, para garantizar un nivel suficiente de seguridad criptográfica, a estas secuencias se les exige una serie de características que las hacen parecer aleatorias. Este concepto se formaliza en parte a través de la pseudoaleatoriedad descrita por los postulados de Golomb [66], que son una serie de condiciones necesarias pero no suficientes que debe cumplir una secuencia para parecer aleatoria. Una secuencia binaria que satisface los tres postulados de Golomb se denomina secuencia pseudoaleatoria o PN-secuencia:

- Postulado G1. Debe existir igual número de ceros (0s) que de unos (1s), aceptándose como máximo una diferencia igual a la unidad. Si una secuencia binaria cumple G1, la probabilidad de recibir un bit 1 coincide con la probabilidad de recibir un bit 0, esto es, un 50 %.

- Postulado G2. En un período P, la distribución de las rachas corresponde a una progresión geométrica, de forma que la mitad de las rachas será de longitud 1, la cuarta parte de longitud 2, la octava parte de longitud 3, etc. Si una secuencia binaria cumple G2, la probabilidad de recibir un bit 1 o un bit 0, después de haber recibido un bit 1 o un bit 0, concuerda, esto es, un 50 %.

- Postulado G3. La autocorrelación AC(k) es constante para cualquier desplazamiento de un periodo de la secuencia k bits.

$$AC(k) = \frac{Coincidencias - Diferencias}{P}$$

  Si una secuencia binaria cumple G3, ningún fragmento de secuencia seleccionado por un potencial atacante, proporcionará más información que cualquier otro fragmento, impidiéndose lanzar ataques relacionados con el periodo de la secuencia ni con correlaciones entre desplazamientos de la secuencia.

Por otra parte, además de la pseudoaleatoriedad, existen otros criterios que una secuencia debe cumplir para ser de utilidad criptográfica [92], tales como:

- C1. El periodo P de la secuencia debe ser muy grande (aproximadamente del orden de 10^{50}).
- C2. La secuencia ha de ser fácil de generar.
- C3. El conocimiento de una pequeña parte de la secuencia cifrante no debe permitir a un criptoanalista generar el periodo completo de la secuencia.

Registro de Desplazamiento con Realimentación Lineal

La mayoría de aplicaciones de comunicaciones, tanto comerciales como militares, que hacen uso de secuencias pseudoaleatorias, utilizan hardware que les permite generarlas a gran velocidad de manera simple y con un bajo coste. Para ello suelen utilizarse dispositivos denominados registros de desplazamiento con realimentación lineal o LFSRs (acrónimo del inglés, Linear Feedback Shift Registers) [66].

Un registro de desplazamiento es básicamente un circuito digital secuencial con varias celdas de memoria conectadas entre sí, donde cada celda almacena un bit cuyo valor depende de entradas y valores anteriores. Así, el valor de todas estas celdas conforman el estado del registro. Cuando se cambia el estado del registro (generalmente al ritmo de un reloj), el nuevo estado del registro se forma simplemente desplazando los bits de cada celda a la celda vecina. Así, el bit en un extremo sale del registro, a la vez que un nuevo bit entra en la celda del otro extremo. En la Figura 2.16 podemos ver la representación de la implementación hardware de un registro de desplazamiento de 4 bits con una entrada de datos, representada mediante Q1-Q4, y una señal de reloj que indica en qué momento se debe pasar de un estado al siguiente.

Una vez explicados qué son los registros de desplazamiento es muy sencillo entender cómo funciona un LFSR. Los LFSRs están constituidos por un conjunto de L etapas o celdas de memoria, interconectadas mediante puertas lógicas OR-exclusivas (XOR). Las celdas de memoria están unidas entre sí y avanzan igual que los registros de desplazamiento pero a diferencia de éstos, el nuevo bit ingresado en cada golpe de reloj es el resultado de la operación XOR de realimentación de algunas de las celdas del estado anterior, con lo que se tiene un sistema cerrado y realimentado linealmente como se muestra en la Figura 2.17.

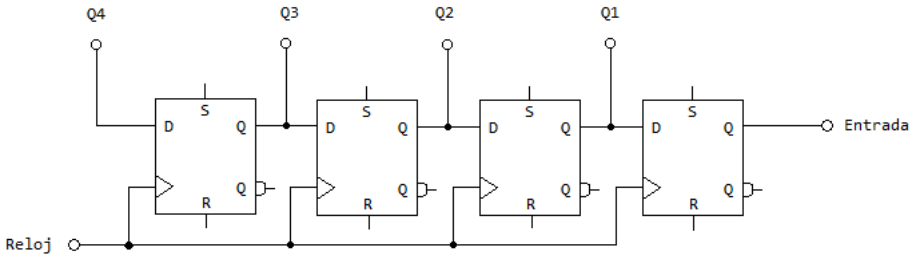


Figura 2.16: Registro de Desplazamiento

Los parámetros que caracterizan un LFSR son el número L de celdas que se denomina longitud del LFSR, y las celdas o etapas que intervienen en la función lineal de realimentación. A cada vector formado por los contenidos de las L etapas $s_j, s_{j+1}, \ldots, s_{j+L-1}$ se le denomina estado del LFSR. Los coeficientes c_i se caracterizan mediante el llamado polinomio de realimentación sobre $GF(2)$ donde $C(x) = 1 + c_1x + c_2x^2 + \ldots + c_Lx^L$, cuyo grado L es la longitud del registro.

Una cuestión importante es que la salida del registro de desplazamiento es periódica, lo que significa que cada celda de memoria repetirá su contenido cada cierto tiempo. Es posible asegurar que el periodo producido por el generador sea el máximo posible mediante la utilización de polinomios de realimentación que además de ser irreducibles sean primitivos.

Se dice que un polinomio es irreducible si es un polinomio con coeficientes enteros que no puede ser factorizado en polinomios de grado menor. Por tanto el concepto de polinomio irreducible es similar al concepto de número primo.

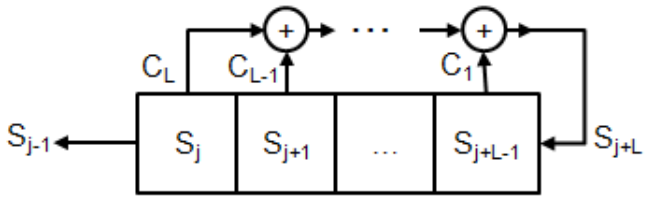


Figura 2.17: Representación Esquemática de un LFSR

Un polinomio de grado L es primitivo si y solo si

- es irreducible
- su orden es 2^L-1 (es decir, divide a $x^{2^{L-1}} + 1$, y es el menor exponente del binomio para el que eso ocurre).

No existe una fórmula sencilla para obtener todos los polinomios primitivos [130] de un grado arbitrario, recurriéndose típicamente a manuales de referencia con tablas o a búsqueda exhaustiva mediante programas informáticos. Sin embargo, la generación de clases específicas de polinomios primitivos, como trinomios por ejemplo, se puede realizar con programas muy eficientes. Además, el número de polinomios primitivos diferentes es $\phi(2^L - 1)/L$, luego crece exponencialmente con respecto a L.

Por tanto, las m-secuencias obtenidas mediante un LFSR de longitud L con polinomio primitivo presentan dos propiedades que las caracterizan y las hacen particularmente interesantes:

1. El período de las m-secuencias es $2^L - 1$.
2. En cada periodo hay 2^{L-k} copias de cada posible formación de k bits, excepto para la de k ceros, de la cual hay $2^{L-k} - 1$ copias, con $1 \leq k \leq L$.

Para ver si una m-secuencia producida con un LFSR es semejante a la que obtendríamos en un experimento totalmente aleatorio debemos también comprobar si cada bit de nuestra secuencia tiene información dependiente de otros bits de la secuencia. Una medida interesante de este tipo de dependencia es la que nos da la función de autocorrelación de la secuencia tal como establecía el tercer postulado de Golomb. En el caso de las m-secuencias, el periodo $P = 2^L - 1$ y la autocorrelación fuera de fase es constante y de valor -1/P. Por tanto, las m-secuencias producidas con LFSRs de polinomio primitivo satisfacen bastante bien las propiedades de pseudoaleatoriedad, pero siempre hasta cierto grado debido a su periodo finito. Estas propiedades, junto con su facilidad de implementación, hacen que los LFSRs sean estructuras muy interesantes para obtener secuencias pseudoaleatorias.

Impredecibilidad y Complejidad Lineal

Uno de los principales problemas que presenta el uso de LFSRs se debe a su linealidad pues el estado inicial o semilla se puede determinar fácilmente con un simple sistema de ecuaciones lineales mediante el uso de la función polinómica $C(x)$ de $2L$ bits consecutivos de salida de secuencia cifrante. Esta característica de la secuencia es un factor crítico en las aplicaciones criptográficas, luego es necesario usar técnicas que permitan resolverla. Una medida que permite determinar el grado de impredecibilidad de las secuencias pseudoaleatorias es la llamada complejidad lineal denotada como $\wedge$ y definida como el menor número de bits de una secuencia necesarios para descubrir el resto [63].

Para que la secuencia producida por un LFSR se pueda considerar impredecible, deberá presentar una complejidad lineal elevada, aunque esto será tan solo una condición necesaria y no suficiente. Una razón para usar la complejidad lineal como parámetro de medida de la impredecibilidad de la secuencia es que el periodo de la secuencia de salida del LFSR es al menos tan grande como la complejidad lineal, por lo que una gran complejidad lineal implicará siempre un periodo grande. Las secuencias pseudoaleatorias muestran un aumento de periodo al aumentar su complejidad lineal. Esto no ocurre siempre a la inversa, pues una secuencia con gran periodo no tiene necesariamente una gran complejidad lineal.

Las secuencias de máximo periodo o m-secuencias obtenidas a partir de LFSRs con polinomio de realimentación primitivo presentan excelentes propiedades respecto a la distribución estadística y el periodo, pero tienen una complejidad lineal mínima, que vale:

$$\wedge = L$$

De hecho conociendo tan solo 2L bits de la secuencia de salida, empleando el algoritmo de síntesis de Massey-Berlekamp [93] se pueden obtener el estado inicial y las conexiones de realimentación del LFSR que los generó, y puede obtenerse el resto del periodo de la secuencia.

Por lo tanto queda claro que cuanto mayor sea la complejidad lineal de la secuencia, mayor será su impredecibilidad. Sin embargo usando solo LFSRs siempre se está limitado en cuando a la complejidad lineal. Si se desean obtener secuencias con mayor impredecibilidad

para aplicaciones criptográficas se debe recurrir a técnicas no lineales que, aplicadas sobre las salidas de los LFSR o sobre sus etapas, produzcan mayores complejidades lineales.

Para intentar conseguir una alta impredecibilidad de las secuencias producidas, los generadores de secuencia cifrante en muchos casos utilizan métodos como los filtrados no lineales. Los filtrados no lineales consisten en funciones que aplicadas sobre las etapas de un LFSR tienen como objetivo incrementar la complejidad lineal de la secuencia de salida, sin modificar las propiedades de distribución estadística obtenidas mediante el uso de conexiones de realimentación lineal con polinomio primitivo.

Lo que se hace al usar funciones no lineales con LFSR es aumentar la dificultad del análisis de la secuencia de salida. La transformación no lineal más simple es el producto de dos dígitos binarios $f(x_1, x_2) = x_1 \cdot x_2$, donde el producto se corresponde con la operación AND. Por tanto, el llamado filtrado no lineal de un LFSR consiste en una función booleana no lineal f que se aplica a las etapas de un LFSR tal y como se muestra en la Figura 2.18 para producir una secuencia cifrante $z_{j \in N}$ llamada en muchos casos secuencia filtrada.

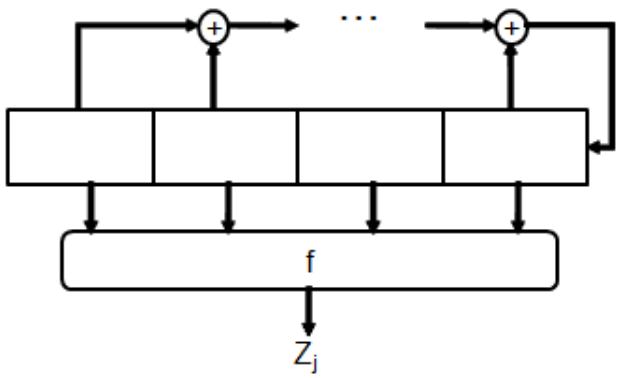


Figura 2.18: Filtrado no Lineal

2.8.2. Generador Propuesto

Como ya hemos visto, el LFSR se usa comúnmente como generador pseudoaleatorio en criptografía debido a las buenas características de las secuencias producidas y también porque su implementación hardware es eficiente y los requerimientos de computación son

simples. Sin embargo, tal como se ha comentado, los LFSRs tienen inconvenientes importantes que deben ser resueltos a fin de ser utilizados de manera segura [35]. La propuesta aquí presentada se basa en un LFSR principalmente porque es un sistema ideal para sistemas con limitaciones tanto en cuestiones de tiempo de generación como en energía y en computación, como podría ser el caso de las VANETs que nos ocupa. En particular, esta propuesta está basada en un registro de desplazamiento con polinomio de realimentación primitivo sobre GF(2), $1+c_1x+c_2x^2+\ldots+c_Lx^L$, de grado L igual a la longitud de la clave usada en cada momento, y alimentado con la semilla formada por dicha clave.

El polinomio de realimentación $C(x)$ del registro propuesto viene dado por el polinomio primitivo de menores coeficientes no nulos y número de dichos coeficientes dado por el menor número posible mayor o igual que $0,1 \cdot L$, para evitar ataques por correlación y garantizar su eficiencia.

El orden de la función de filtrado propuesta es el número primo p más cercano a $L/2$ y menor que $L/2$, para garantizar una complejidad lineal grande. Dicha función incluye un término lineal correspondiente a su orden, además de un número de términos de cada orden $i = 2, 3, \ldots, p$ dado por la parte entera de L/i. Estos términos se obtienen multiplicando sucesivas etapas para lograr pseudoaleatoriedad y confusión.

Para evitar ataques de correlación, la salida de dicho filtrado no lineal se decima irregularmente de manera que el bit más significativo del registro en cada momento determina si la correspondiente salida de la función de filtrado se utiliza o se descarta. Finalmente, con objeto de garantizar una salida estable, se incluye un buffer de tamaño 4.

A continuación se especifican los detalles concretos del diseño, que se muestra en la Figura 2.19. El LFSR utilizado en el diseño propuesto es de longitud $L = 20$. Su contenido se denota como $s_j, s_{j+1}, \ldots, s_{j+19}$. En particular, el polinomio de realimentación propuesto $C(x)$ del LFSR es un polinomio primitivo de grado 20 y se define como $C(x) = 1+x^3+x^{20}$, por lo que la función de realimentación es $s_{j+20} = s_{j+17} + s_j$. El contenido del LFSR de 20 bits es la entrada de la función de filtrado no lineal f. Concretamente, 20 variables correspondientes al estado del LFSR forman la entrada de una función booleana orden 7, escogida por ser equilibrada, con inmunidad a la correlación de orden 1 y con un alto grado de no linealidad.

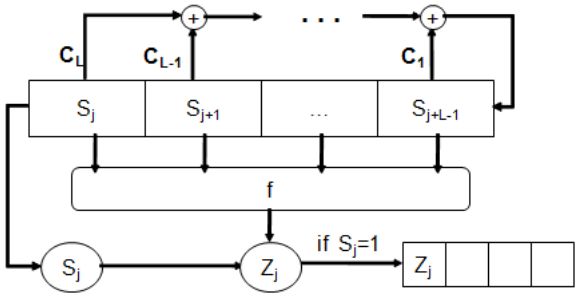


Figura 2.19: Descripción General del PRNG

En particular la primera función de filtrado propuesta se define de la forma siguiente sobre una entrada $x = (x_1, x_2, \ldots, , x_{20})$:

$$f(x) = x_{i+7} + \sum_{i=1}^{10} x_i \cdot x_{i+1} + \sum_{i=1}^{6} x_i \cdot x_{i+1} \cdot x_{i+2} + \sum_{i=1}^{5} x_i \cdot x_{i+1} \cdot x_{i+2} \cdot x_{i+3} + \sum_{i=1}^{4} x_i \cdot x_{i+1} \cdots x_{i+4} + \sum_{i=1}^{3} x_i \cdot x_{i+1} \cdots x_{i+5} + \sum_{i=1}^{2} x_i \cdot x_{i+1} \cdots x_{i+6} \tag{2.11}$$

2.8.3. Análisis del Generador

La calidad de un PRNG se puede medir tomando una muestra de la secuencia de salida y sometiéndola a pruebas. Con el fin de comprobar la pseudoaleatoriedad del generador propuesto en este trabajo se han generado varias secuencias y se han sometido a una batería de test estadísticos. Es imposible dar una demostración matemática de que el generador es un buen generador de bits pseudoaleatorios, porque el hecho de que logre pasar los test de manera exitosa no permite asegurar de forma inequívoca que genere secuencias pseudoaleatorias. Sin embargo, pasar la prueba de la batería de tests sí proporciona evidencias a favor de un buen grado de pseudoaleatoriedad. Por lo tanto, si el resultado de pasar los tests es negativo podemos concluir que el generador falla y es rechazado por no ser pseudoaleatorio. Si por el contrario los pasa significa que es lo suficientemente bueno como para no descartarlo de inmediato, pero no significa que funciona para todo tipo de aplicaciones ni que las secuencias cifrantes que produzca puedan considerarse criptográficamente fuertes.

Tests Estadísticos

En el tema de los PRNGs usados en cifrado en flujo el análisis más importante a realizar consiste en evitar que cualquier posible atacante sea capaz de encontrar alguna regularidad en la secuencia de salida. Si este fuera el caso, podría explotarla para lanzar un ataque y predecir los siguientes bits de salida. Por esta razón, se requiere que el flujo de salida sea indistinguible de una secuencia aleatoria.

A fin de demostrar la pseudoaleatoriedad de nuestro generador, el PRNG propuesto ha sido implementado en el entorno de VANETs donde se propone utilizarlo. Su implementación se detallará en el siguiente capítulo. Los datos generados para comprobar sus propiedades estadísticas constan de 3,9 Gb correspondientes a $2^{20} - 1$ bits producidos con el generador para cada una de las semillas posibles del LFSR. El análisis de secuencias se realiza, principalmente usando dos distribuciones estadísticas, la normal y la X^2, con un intervalo de confianza del 95 %. En concreto, las secuencias se han sometido a la siguiente batería de tests estadísticos:

- Test de Frecuencia. El propósito de este test es determinar si el número de unos y ceros en una secuencia es aproximadamente el mismo que se espera para una secuencia verdaderamente aleatoria. La prueba evalúa la cercanía de la fracción de unos a 1/2, es decir, comprueba que el número de unos y ceros en una secuencia es aproximadamente el mismo, luego se corresponde con el primer postulado de Golomb.

- Test de Series. El objetivo de esta prueba es determinar la frecuencia de aparición del patrón de m-bits a través de toda la secuencia. Se cuenta el número de ocurrencias de las cuatro subsecuencias: 00, 01, 10 y 11, y se examina si la distribución es aproximadamente la misma que se espera para una secuencia aleatoria. Si es así se concluye que la secuencia pasa el test.

- Test de Póquer. En esta prueba se dividen los datos en grupos de cuatro bits consecutivos para determinar cuántas veces aparecen las dieciseis combinaciones posibles de cuatro bits. El cuadrado de cada resultado se suma y se escala para ver si el resultado se encuentra dentro de en un rango determinado. Si es así, la secuencia pasa la prueba.

Test	Pasa	Falla
Frecuencia	100 %	0 %
Series	100 %	0 %
Póquer	99,27 %	0,72 %
Rachas	31,79 %	68,21 %
Autocorrelación	98,94 %	1,05 %

Figura 2.20: Resultados de Test Estadísticos con Primer Filtrado

- Test de Rachas. El objeto de esta prueba es el número total de rachas de ceros y unos existentes en toda la secuencia. El propósito del test de rachas consiste en determinar si el número de rachas de unos y de ceros de diferentes longitudes es el esperado para una secuencia aleatoria. Este test se corresponde con el segundo postulado de Golomb.

- Test de Autocorrelación. El propósito de la prueba de autocorrelación es comprobar que los bits de la secuencia no dependen de otros bits. Para llevarlo a cabo se elige un número d y luego se compara cada bit i con el bit i + d. Si los dos bits son los mismos con demasiada frecuencia, o por el contrario son diferentes con demasiada frecuencia, la prueba se rechaza. Este test se corresponde con el tercer postulado de Golomb.

En la Figura 2.20 se muestra el resumen de los datos obtenidos para los test presentados anteriormente, realizados sobre las secuencias generadas con el PRNG propuesto.

Según la tabla de la Figura 2.20, las secuencias generadas pasan el test de frecuencia con un 100 % de resultados positivos. Los resultados para el test de series son también de un 100 % de resultados positivos para el generador propuesto. Más del 99 % de las secuencias pasan el test de póquer mientras que casi el 99 % de los tests de autocorrelación producen resultados positivos. Como se puede observar, en todos estos tests los resultados son muy satisfactorios para las secuencias generadas. Sin embargo, en particular el test de rachas produce casi un 32 % de resultados positivos, lo cual no es un resultado aceptable. El problema se presenta en el test de rachas que, como se comentó anteriormente tiene como propósito determinar si el número de rachas de unos y de ceros de diferentes longitudes es el esperado para secuencias aleatorias. Esto nos dio una pista de dónde estaba el problema. En concreto, y para resolver el inconveniente nos centramos en la función de filtrado a la

Test	Pasa	Falla
Frecuencia	100 %	0 %
Series	100 %	0 %
Póquer	99,01 %	0,99 %
Rachas	98,55 %	1,45 %
Autocorrelación	98,68 %	1,32 %

Figura 2.21: Resultados de Test Estadísticos con Segundo Filtrado

cual realizamos una pequeña modificación respecto a la primera función de filtrado no lineal propuesta (ecuación 2.11) para evitar la superposición de etapas del LFSR en los términos de cada orden, y así resolver el problemas las rachas.

Por tanto, la nueva función de filtrado mejorada se define como:

$$f(x) = x_{i+7} + \sum_{i=1}^{10} x_{2i-1} \cdot x_{2i} + \sum_{i=1}^{6} x_{3i-2} \cdot x_{3i-1} \cdot x_{3i} + \cdots + \sum_{i=1}^{2} x_{7i-6} \cdot x_{7i-5} \cdot x_{7i-4} \cdot x_{7i-3} \cdot x_{7i-2} \cdot x_{7i-1} \cdot x_{7i} \tag{2.12}$$

Una vez realizados los cambios en la función de filtrado, se han vuelto a generar secuencias con el nuevo generador propuesto y se han sometido a la batería de tests presentada anteriormente. La Figura 2.21 muestra los resultados obtenidos con el PRNG, una vez implementada la nueva función de filtrado:

Según la tabla de la Figura 2.21 las secuencias generadas pasan el test de frecuencia con un 100 % de resultados positivos. Los resultados para el test de series son también de un 100 % de resultados positivos para el generador propuesto. El 99 % de las secuencias pasan el test de póquer mientras que más del 98 % de los tests de autocorrelación producen resultados positivos. Finalmente se puede observar cómo para el test de rachas las secuencias generadas ahora producen más de un 98 % de resultados positivos, lo cual en este caso suponen un resultado aceptable para nuestro generador.

Por tanto los resultados obtenidos muestran que las propiedades de pseudoaleatoriedad de las secuencias producidas por el PRNG propuesto son buenas. Sin embargo somos conscientes de que es necesario probar otros tests importantes como los propuestos por el

grupo NIST en [112], además de diversas características de resistencia a ataques conocidos, antes de poder concluir que el generador es robusto criptográficamente hablando, para su uso como base de la protección de la confidencialidad de las comunicaciones en VANETs. Completar la verificación de dicha robustez es parte de los trabajos futuros de esta tesis.

Capítulo 3

Agregación

En la actualidad son numerosas las investigaciones sobre VANETs pues se espera que en el futuro este tipo de redes permitan reducir e incluso evitar el número de muertes en las carreteras gracias a la información proporcionada en tiempo real sobre el estado de las mismas. Estas redes permitirán a los conductores el intercambio de información sobre eventos potencialmente peligrosos, como podrían ser accidentes, obstáculos en la vía, etc. Además propocionarán la posibilidad de encontrar plazas de aparcamientos libres en una determinada zona y de evitar o reducir congestiones mediante el conocimiento de las condiciones de tráfico en tiempo real.

Para que todas estas aplicaciones funcionen correctamente, es necesario garantizar que la información que circula en la red sea fidedigna por lo que sería conveniente evitar o al menos disminuir el número de advertencias falsas en la misma. Ninguno de los esquemas existentes protegen a las VANETs de ataques sencillos y a la vez dañinos, como son la generación y/o distribución de contenidos falsos y posibles modificaciones o repeticiones de mensajes enviados. Un atacante como por ejemplo un conductor que tenga prisa por llegar a su destino, podría inyectar información que no se corresponda con lo que está observando realmente, diseminando información falsa como por ejemplo la existencia de una congestión en su trayectoria para intentar disminuir el número de vehículos que circulan por la misma.

Es cierto que gracias a la firma digital basada en criptografía de clave pública sería posible determinar y sancionar al vehículo que presenta información falsa como verdadera.

Sin embargo, el tiempo necesario para afrontar este problema a posteriori por parte de las administraciones públicas hace que esta aproximación no sea práctica en una VANET. Lo ideal sería contar con un mecanismo de detección de estos ataques, que sea automático y funcione en tiempo real. Para afrontar esta cuestión se propone en esta tesis utilizar la agregación de datos. Si bien es cierto que la agregación de datos se ha utilizado en muchos trabajos como un mecanismo que permite disminuir el número de paquetes que circulan en la red, se puede utilizar también para aumentar la fiabilidad de la información diseminada, tal como se verá a continuación. En este capítulo utilizamos la idea de agregación de datos basada en un esquema probabilista para detectar intentos de ataque de modificación o repetición de los datos de manera rápida y fiable. Los resultados expuestos en este capítulo se encuentran incluidos en las publicaciones [36], [99], [103], [107], [108].

3.1. Estado del Arte

En diferentes referencias bibliográficas relacionadas con la protección de las comunicaciones en VANETs, se pueden encontrar varios artículos que proponen el uso de firma digital basada en criptografía asimétrica combinada con GPS para verificar el origen y la integridad de los mensajes especialmente en aplicaciones de gestión de flotas [64]. Otros trabajos proponen el uso de cifrado simétrico para proporcionar privacidad de localización y evitar ataques de seguimiento [139]. También se pueden encontrar propuestas basadas en el uso de pseudónimos para garantizar la protección de identidad de los usuarios, como la descrita en [39], que afronta el problema de la gestión de pseudónimos mediante el uso de firmas de grupo. Sin embargo, ninguno de los mecanismos anteriores protege al sistema contra ataques activos, como la generación de paquetes con contenido falso, que es uno de los objetivos del presente trabajo. Un nodo legítimo pero malicioso podría tratar de inyectar información falsa que no se corresponda con la información real de su entorno. Para hacer frente a este problema, el sistema descrito en [46] utiliza un mecanismo basado en firmas umbral, que evita que los atacantes internos envíen mensajes con contenido falso e imposibilita relacionar la identidad de un mismo usuario o vehículo en diferentes condiciones o puntos de la carretera. Sin embargo, esa propuesta requiere la participación de

una autoridad gubernamental de confianza, que no está disponible en redes totalmente distribuidas y descentralizadas como las que aquí se analizan. Otros trabajos también proponen la agregación de datos para hacer frente a este problema, pero en condiciones diferentes. Por ejemplo, [114] analiza la relación entre la seguridad y la agregación de datos en redes de sensores inalámbricos.

El tema de este capítulo se centra en la necesidad de actuar para evitar que se envíe información falsa en una VANET. Por lo tanto, se trata de gestionar la confianza en la red. Relacionado con este tema, en el reciente informe [148] se analizan los retos y se identifican algunas características deseadas para la gestión eficaz de la confianza en VANETs llegando a la conclusión de la falta de eficacia de todos los modelos existentes. Por otro lado, con respecto a la identificación de ataques maliciosos, un trabajo que trata este tema es [65], donde cada nodo compara los datos recibidos con la información almacenada, asumiendo que cada vehículo tiene un conocimiento global de la red, cosa que en una red totalmente distribuida y descentralizada como la que proponemos en este trabajo no se cumple.

Con respecto a la autenticación de los datos en VANETs, en [147] se introduce un nuevo esquema de autenticación de mensajes que hace responsable a las RSUs tanto de la verificación de los mensajes enviados por los vehículos, como de la notificación de los resultados a los vehículos. A diferencia de esta propuesta, el modelo aquí presentado no requiere de ninguna RSU. Por otro lado, los autores de [47] proponen que no sólo se tengan en cuenta los datos acerca de alertas recién generadas sino que también se considere el historial de los datos agregados de alertas que los vehículos mantienen almacenado, lo cual es un enfoque diferente al utilizado en el presente trabajo. Otra propuesta se puede encontrar en [52], donde se utiliza la agregación de varios mensajes que describen el mismo evento y el uso de mensajes de revocación de los vehículos en caso de proporcionar información falsa. Sin embargo, este mecanismo tiene una debilidad importante, porque es posible que mensajes reales puedan ser revocados como falsos. En [118] la solución propuesta se basa en el uso de un dispositivo a prueba de falsificaciones y consiste en realizar una pregunta a un vehículo agregado acerca de uno de los registros agregados escogido al azar. La principal desventaja de este método es la dependencia de un dispositivo a prueba de falsificaciones, ya que un atacante podría fácilmente saltarse este servicio con el fin de componer datos

agregados falsos. [49] propone otro mecanismo para garantizar la seguridad a través de la agregación en un esquema donde las carreteras se dividen en segmentos de tamaño fijo según la cobertura de señal Wi-Fi. Los autores de [144] también esbozan un esquema de agregación que combina toda la información que se genera en cada segmento de carretera de longitud fija mediante el valor promedio. Sin embargo, ambos criterios de agregación utilizan la idea de dividir la carretera en segmentos fijos, lo que se ha demostrado que no funciona correctamente en grandes áreas cuando el número de vehículos es elevado, como por ejemplo en grandes atascos de tráfico que cubren kilómetros.

Un trabajo bastante relacionado con la propuesta aquí presentada es [90], que propone un algoritmo que elige uno de los múltiples agregados que se generan en una misma área basándose en una aproximación probabilística. La diferencia con nuestra propuesta es clara, porque aquí el enfoque probabilístico se propone en la verificación de los datos agregados, y no en la fase de agregación. Además, en aquel esquema se usa la agregación para combinar en un solo valor, la información generada en zonas de gran tamaño, en lugar de utilizar varios paquetes agregados producidos por diferentes grupos reactivos, que es una de las bases de la propuesta descrita en este capítulo.

Finalmente, respecto a la formación de grupos de vehículos, que es también un tema tratado en este capítulo, hay muchos trabajos con diferentes propuestas. Así, por ejemplo, [74] presenta un protocolo para la transmisión de información bajo el supuesto de que los vehículos forman grupos en la carretera, llamados clúster, y que los nodos en el mismo grupo se intercambian detalles sobre velocidad e información de tráfico. En su propuesta, la información agregada contiene también las posiciones relativas de todos los coches respecto al líder del grupo y la velocidad media, lo cual no es necesario en el esquema aquí descrito. Su propuesta también reduce la cantidad de datos transmitidos sobre un grupo de coches, pero no incluye ningún mecanismo de fusión de datos agregados.

3.2. Preliminares

A continuación se describe un método natural para tratar el problema de la autenticación de datos, que implica que los nodos almacenen y procesen la información recibida,

incluyendo datos sobre el tipo de advertencia, la vía donde fue generado el evento, el sentido de circulación del tráfico, y la identificación del nodo que ha generado el paquete, entre otros. Cuando un nodo recibe un aviso, su dispositivo tiene básicamente dos opciones: o bien alertar al conductor del peligro aunque la información no sea verdadera, o no alertar al conductor y esperar para poder comparar los datos recibidos de diferentes vehículos con el fin de asegurar que la información es verdadera. La primera opción podría afectar a las decisiones del conductor y provocar una pérdida de tiempo y/o dinero si la información recibida no es verdadera. Además, puede crear desconfianza sobre futuros mensajes que se reciban de la red. Por lo tanto, parece que la opción recomendada en VANETs sería la segunda a pesar de que esta demora podría causar un accidente. El retardo no solo es causado por el tiempo requerido para comparar los datos recibidos sobre el mismo evento, sino también por la espera hasta recibir un número suficiente de paquetes con el mismo contenido y de diferentes fuentes. Además esta opción requiere que los vehículos tengan un gran espacio de almacenamiento para guardar los eventos recibidos, así como un mecanismo rápido para comparar los diferentes registros. Por lo tanto, la implementación de esta opción debe tener en cuenta que el tiempo de espera del sistema debe ser lo suficientemente corto como para advertir al conductor con la suficiente antelación de manera que le permita tomar decisiones para evitar el problema, y lo suficientemente largo como para recibir evidencias sobre el mismo evento de un número de vehículos que le permita comprobar que el contenido de la información es verdadero.

En un modelo básico, todos los vehículos que detectaran un evento, firmarían el mensaje de advertencia y lo transmitirían a la red, lo que implicaría una sobrecarga considerable. Los nodos que recibieran este paquete firmado, tendrían que verificar su firma y comparar el contenido del mensaje con otros mensajes relacionados y recibidos previamente de diferentes vehículos, lo que también provocaría un retardo importante (véase la Figura 3.1). Para resolver estos problemas, se podría utilizar un modelo sencillo que combine las firmas generadas por diferentes vehículos para alertar sobre el mismo peligro. Sin embargo, la combinación directa de las firmas en un solo paquete aumentaría el tamaño del paquete, a medida que aumenta el número de vehículos que confirma la información. Esto también implicaría una sobrecarga de los canales y requeriría que el receptor verificara

todas las firmas, lo que significaría un nuevo retardo que igualaría o incluso superaría el tiempo requerido por el modelo básico.

Con el fin de tratar de resolver los problemas mencionados anteriormente, en este capítulo se propone un nuevo modelo basado en dos componentes principales. Por un lado, se propone el uso de lógica difusa para la toma de decisiones. Por otro lado se utilizan grupos reactivos para proporcionar información confiable a través de la combinación limitada de firmas en un paquete agregado, y para solucionar el retraso ocasionado por la verificación de firmas, se propone un esquema probabilístico que define la verificación de sólo algunas firmas elegidas al azar entre todas las firmas incluidas en el paquete. Estos dos mecanismos de seguridad son la base del nuevo modelo que se detalla a continuación.

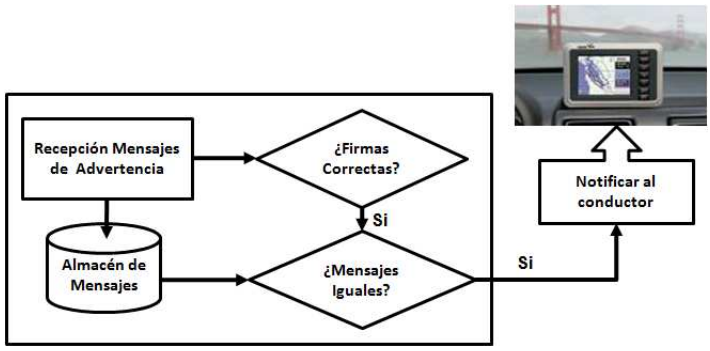


Figura 3.1: Modelo Básico

3.3. Agregación de Datos con Lógica Difusa

Los esquemas de agregación suelen estar basados en tres fases consecutivas:

- La toma de decisiones. Cuando el sistema debe decidir si dos piezas de información son lo suficientemente similares como para agregarlas en un mismo paquete o no.
- La agregación de datos. Una vez que el sistema ha concluido que los paquetes recibidos deben ser agregados, se aplica un método de agregación para combinarlos.
- La entrega de datos. El paquete de agregación generado debe ser enviado a través de la red para alcanzar al máximo número de usuarios posible.

La toma de decisiones se basa normalmente en grupos predefinidos, árboles o estructuras fijas, que no son adecuados para su uso en VANET. Por lo tanto, aquí se propone un nuevo esquema para la toma de decisiones respecto a si dos piezas de información recibidas son lo suficientemente similares como para generar un paquete de agregación. Tales decisiones tienen en cuenta posibles errores o aproximaciones con respecto a los datos de espacio, tiempo, velocidad y dirección que definen el evento. En particular, la propuesta para la toma de decisiones se basa en la aplicación de lógica difusa centrándose especialmente en la descripción de los criterios de decisión.

Con el fin de decidir si los datos recibidos coinciden o no con la información percibida por el receptor, la propuesta para la etapa de toma de decisiones se basa en cuatro dimensiones: tiempo, espacio, dirección y velocidad. En primer lugar, hay que tener en cuenta el intervalo de tiempo en el que un evento es válido. En particular esta propuesta realiza una verificación sobre la fecha de generación de un evento. En caso de tratarse de un evento antiguo se descarta el paquete y no se hace ninguna otra comprobación sobre los demás parámetros para reducir el procesamiento. En segundo lugar, la ubicación de un evento en la carretera o en el mapa es otro parámetro esencial. Los nodos que reciben un evento comprueban las coordenadas geográficas del paquete recibido y las comparan con las coordenadas geográficas actuales de sus vehículos. Si se encuentran dentro del mismo rango, se supone que son capaces de detectarlo. En tercer lugar, hay que distinguir el sentido de la marcha en el que se encuentra el evento debido a la existencia de carreteras con doble sentido de circulación en las que por ejemplo es posible que exista un atasco en una dirección y que el sentido contrario de circulación de la vía se encuentre totalmente libre de vehículos. Por tanto, los nodos deben verificar la dirección en la que el paquete indica que se encuentra el evento y compararla con el sentido de su marcha para decidir si firmar/agregar el paquete recibido. Por último, ya que el próximo capítulo presenta una herramienta que detecta automáticamente las congestiones, se propone un control de velocidad. Si un nodo recibe un paquete de información sobre la existencia de un atasco de tráfico en la carretera en la que está actualmente circulando, el nodo comprueba, durante un período de tiempo, si su velocidad es inferior a la velocidad normal permitida en dicha carretera. Si es así, puede certificar que hay un atasco de tráfico.

A continuación definimos una estrategia difusa a aplicar sobre los parámetros mencionados con objeto de tomar las mejores decisiones teniendo en cuenta todos los valores posibles para identificar el mismo evento. Previamente introducimos algunos conceptos necesarios.

3.3.1. Sistema de Lógica Difusa

En la Figura 3.2 vemos un esquema típico de control de lógica difusa. Como se puede observar, inicialmente las variables de entrada reales se convierten en variables lingüísticas mediante un proceso de fusificación. Usando esas variables, se evalúa un conjunto de reglas de control, que darán como resultado una serie de valores lingüísticos a su salida. Por último, hay que defusificar estas variables obteniéndose los valores reales de salida.

Figura 3.2: Esquema Inicial de Agregación

1. Fusificar: Mediante este proceso se calcula el grado de pertenencia de la entrada a uno o varios conjuntos difusos. Las funciones de pertenencia se definen a partir de los datos y/o el sentido común.

2. Evaluación de las reglas de control (también llamado proceso de inferencia): Con el objeto de determinar cómo se va a comportar el sistema, se establece una serie de reglas de la forma **SI (...) ENTONCES (...)** que indican la acción a realizar según a qué conjunto pertenezca la entrada en función de la combinación de los parámetros

escogidos. Evaluar las reglas significa determinar cuáles de ellas se activarán frente a un subconjunto de valores de pertenencia a los conjuntos difusos del dominio de entrada. En particular, según las reglas definidas, los grados de los parámetros escogidos se combinan usando operadores tales como AND, OR y NOT definidos como mínimo, máximo y complementario, respectivamente. La salida de cada regla solo tiene dos posibles valores: Si o No.

3. Defusificar: La entrada para el proceso de defusificación es un conjunto difuso y la salida es un número real simple.

3.3.2. Espacio y Tiempo

Como ejemplo de la toma de decisiones de lógica difusa basada en reglas, se propone en primer lugar seleccionar y combinar los parámetros de localización espacial y temporal denotados como Espacio-Diferencia (SD) y Tiempo-Diferencia (TD) para paquetes referentes a un evento en carretera. SD mide la distancia entre las coordenadas geográficas actuales del nodo y las de un evento que ha sido recibido por el nodo. TD mide la diferencia entre el momento actual y el momento indicado en el paquete en el que se detectó el evento recibido. La Figura 3.3 es un ejemplo de este proceso, donde ambas variables representan la influencia de ambos parámetros teniendo en cuenta que la coordenada X representa respectivamente SD en metros o TD en minutos, y la coordenada Y es la probabilidad que se corresponde con los adjetivos: “Bajo”, “Medio” y “Alto”. Cada uno de estos adjetivos se describe mediante una función de pertenencia que asocia el valor real de entrada del factor de influencia con un grado de pertenencia correspondiente al adjetivo descrito por la función. En el ejemplo representado, se considera que el error típico de un GPS normal es de unos 23 metros de ambigüedad, por tanto, una entrada SD de menos de 3 metros es fusificada como “Bajo” con un grado de 1. Por ejemplo, la salida de la función para una SD de alrededor de 9 metros se clasifica al mismo tiempo como “Bajo” y como “Medio” con un grado de 0,5. De 27 metros en adelante, SD es considerado “Alto” con una probabilidad de 1. Lo mismo se considera de TD en minutos, considerando que en general un mismo evento puede ser detectado durante al menos 3 minutos por diferentes usuarios y que a partir de 27 minutos

podría tratarse de un evento distinto.

El siguiente paso después de fusificar es la formulación de reglas particulares que expresan la combinación de las influencias. A modo de ejemplo, una posible estructura simple de tales reglas de lógica difusa sería la siguiente:

Algoritmo Reglas Difusas sobre SD y TD

01: **if** (SD **es** Bajo) **OR** ((SD **es** Medio) **AND** (TD **es NOT** Alto))
02: **then**
03: Decisión-Agregar **es SI**;
04: **else**
05: Decisión-Agregar **es NO**;
06: **endif**

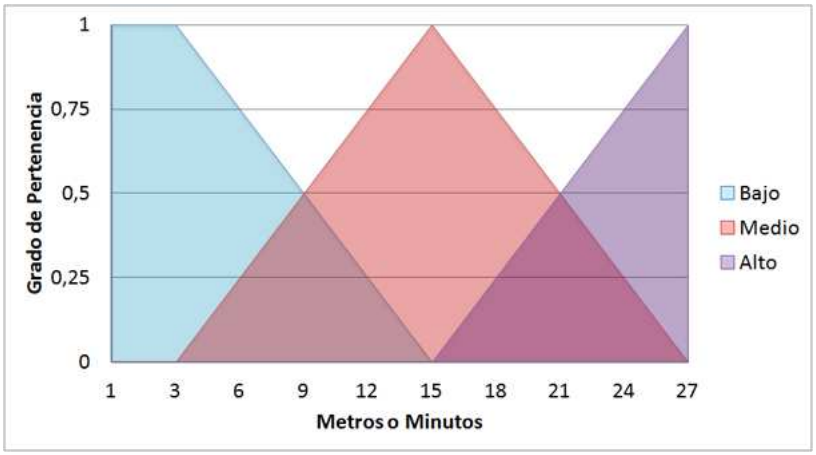


Figura 3.3: Función de Fusificación para Espacio y Tiempo

Puede haber más de una regla de asignación de valores para la decisión de agregación. En este caso, las asignaciones a la decisión de agregación se combinan mediante AND, por lo que la probabilidad correspondiente a la decisión de agregación se corresponderá con el valor mínimo entre las probabilidades de entrada de SD y TD. Después de que todas las reglas hayan sido evaluadas, la decisión será SI o NO dependiendo de cuál de los dos tiene la mayor probabilidad asignada por las reglas. Por ejemplo, si SD es 10 m es fusificada como BAJO con grado 0,32 y como MEDIO con grado 0.68, y si para el mismo par de paquetes TD es 20 min y por tanto TD fusificada como MEDIO con grado 0,58 y

como ALTO con grado 0.42, la decisión de agregación es que SI con grado 0,58 y que NO con grado de 0.42, por lo que la decisión final es SI con mayor probabilidad, de forma que se agregarían los dos paquetes, porque el sistema llegaría a la conclusión de que se refieren al mismo evento.

3.3.3. Velocidad

La velocidad a la que un vehículo se mueve en un carril determina si existe o no congestión en la ruta. Con el fin de determinar la existencia de una congestión, una primera aproximación utilizada en este trabajo consiste en obtener la velocidad máxima permitida en la vía por la que se circula Vel_{Max} y por otro lado la velocidad actual a la que circula el vehículo Vel_{actual} con el fin de comparar ambos valores. Si la diferencia entre la velocidad a la que circula el vehículo y la velocidad máxima permitida es muy grande, por ejemplo es la cuarta parte, el sistema concluye que el vehículo se encuentra en una congestión.

$$\begin{cases} Si \quad Vel_{actual} > \frac{Vel_{Max}}{4} \quad No\ congestion \\ Si \quad Vel_{actual} < \frac{Vel_{Max}}{4} \quad Si\ congestion \end{cases} \tag{3.1}$$

Cuando el sistema recibe un paquete acerca de la existencia de una congestión en la ruta, deberá comparar la velocidad a la que el vehículo circula por la vía y determinar si efectivamente se detecta una congestión en la vía tal y como indica el paquete. Como puede observarse en la ecuación 3.1, el sistema anterior es muy estricto pudiendo llevar al sistema a la conclusión de que la información es falsa por una diferencia de una décima entre la velocidad indicada en el paquete y la velocidad a la que circula el vehículo. Como solución el receptor de un mensaje refiriéndose a un evento de congestión decide agregar su firma o no en función de un esquema difuso definido sobre si el movimiento de su vehículo es alto o bajo. En particular, a medida que la velocidad a la que circula el vehículo crece es más improbable que la decisión sea agregar y firmar el paquete de aviso de congestión, independientemente del resto de parámetros.

En este caso la salida de la función puede ser “Baja” o “Alta” como se puede ver en la Figura 3.4, correspondiente a una vía de $Vel_{Max} = 120Km/h$. En dicha figura la coordenada X representa la velocidad en km/h a la que circula el vehículo que ha recibido el

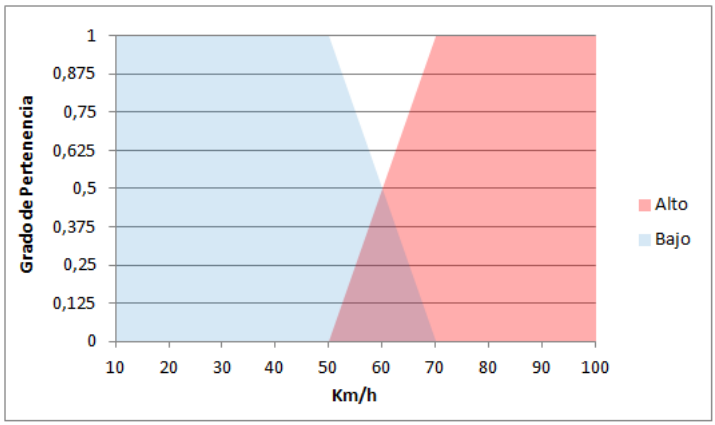


Figura 3.4: Función de Fusificación para Velocidad

mensaje que anuncia una congestión, mientras que la coordenada Y indica la probabilidad de que dicho vehículo decida firmar el mensaje agregado, correspondiendo a los adjetivos "Bajo" o "Alto" según se acepte o no que se detecta una congestión de tráfico. El punto de inflexión entre Alto y Bajo dependerá de la velocidad máxima de la vía, correspondiéndose su valor $\frac{Vel_{Max}}{2}$ y por lo tanto la decisión no dependerá de la comparación entre dos números sino de si el sistema determina que el vehículo se está moviendo a una velocidad alta o baja. Las reglas que definimos para este caso son simples.

Algoritmo Reglas Difusas sobre Velocidad

```
01: if Vel_actual es Baja then
02:     Decisión-Agregar es SI;
03: endif
04: if Vel_actual es Alta then
05:     Decisión-Agregar es NO;
06: endif
```

Supongamos por ejemplo que el valor de Vel_{actual} a la que circula el vehículo es de 55 km/h, para una vía con Vel_{Max}=120km/h, por lo que la velocidad se considera baja, con grado 0,75 y por lo tanto la decisión de agregación es SI con el mismo grado. Del mismo modo, la decisión será NO con grado 0,25 debido a que la velocidad es alta, con este grado. Por tanto la decisión de agregación en función de la velocidad sería SI con mayor

probabilidad porque se entiende que efectivamente el vehículo puede detectar un atasco de tráfico.

3.3.4. Dirección

El cuarto parámetro considerado a la hora de determinar si los datos recibidos acerca de un evento se corresponden con la información percibida es la dirección en la que los vehículos se están moviendo. La información que proporciona el GPS asocia los cuatro puntos cardinales Norte, Sur, Este y Oeste con grados según los valores mostrados en la Figura 3.5 donde 0^o se corresponde con el Norte.

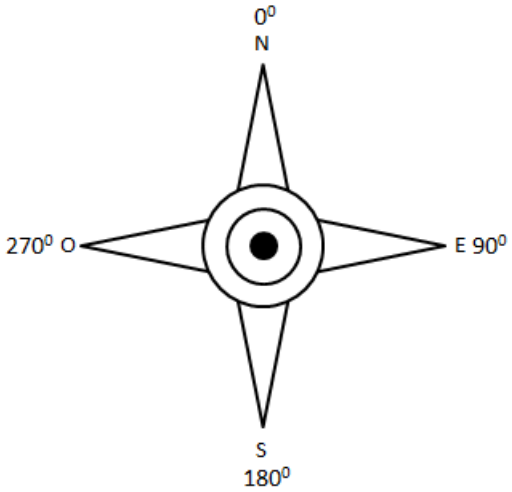


Figura 3.5: Puntos Cardinales

Un vehículo que circula a 43^o se puede considerar que circula en sentido Norte teniendo en cuenta la descripción anterior. Si el sistema genera un evento con esta información y es recibido por un vehículo que se mueve a 46^o que se corresponde con Este, el sistema concluiría que no se están moviendo en el mismo sentido y por lo tanto no se trata del mismo evento. En este caso parece claro que podría tratase de un error y más si se tiene en cuenta que las carreteras suelen tener curvas. Para solucionar este problema se podría pensar en añadir los puntos cardinales Nordeste, Noroeste, Sudeste y Suroeste. Véase la Figura 3.6 con la "Rosa de los Vientos" representando los ocho puntos cardinales.

Sin embargo, tras realizar pruebas hemos comprobado que el problema es exactamente el mismo que se ha indicado anteriormente. Por ejemplo, si Norte es de 337,5º a 22,5º y Nordeste es de 22,5º a 67,5º en los límites entre Norte y Nordeste tenemos el mismo problema. Por lo tanto este mecanismo es demasiado restrictivo como para poderlo utilizar de forma exacta para determinar si dos vehículos circulan en la misma dirección. Sin embargo, mediante el empleo de técnicas difusas se pueden implementar sistemas de control más flexibles.

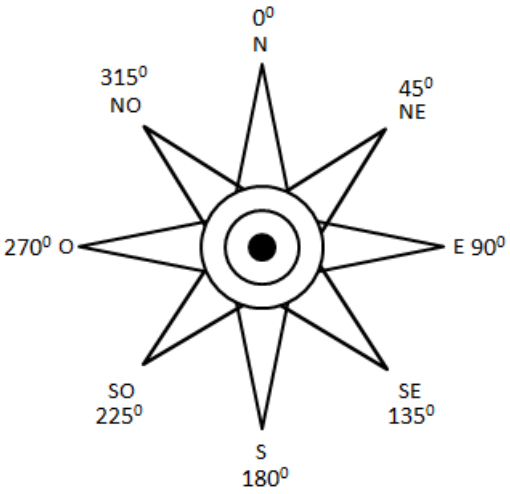


Figura 3.6: Rosa de los Vientos

Para resolver el problema de la dirección se plantea una estrategia similar a la usada con espacio y tiempo basándose en la diferencia entre la información recibida y la información percibida. El parámetro a utilizar para tomar la decisión será Dirección-Diferencia (DD). La coordenada X representa DD en grados y la coordenada Y es la probabilidad que se corresponde con los adjetivos Bajo, Medio y Alto. Cada uno de estos adjetivos se describe mediante una función de pertenencia que asocia el valor de la diferencia de la información recibida y la percibida con el grado de pertenencia correspondiente al adjetivo descrito por la función. A continuación se muestra el cálculo que se realiza antes de pasar la entrada a la función de fusificación. Siendo el ángulo uno (A1) el grado correspondiente a la dirección en la que circula el vehículo que generó el paquete recibido y el ángulo dos (A2) el grado que devuelve el sistema donde se está llevando a cabo el proceso de decisión, la diferencia será la

que nos indique la decisión a tomar. Teniendo en cuenta que los datos proporcionados se corresponden con los 360º de una circunferencia se define la siguiente función diferencia DD:

$$\begin{cases} 360 - |A1 - A2|, & si\ |A1 - A2| > 180 \\ |A1 - A2|, & si\ |A1 - A2| \leq 180 \end{cases} \tag{3.2}$$

Como se observa en la Figura 3.5, para pasar de una dirección a otra hay una diferencia de 90º. Por tanto una entrada de DD de menos de 45º se fusifica como una diferencia baja con un grado de 1. Una diferencia de alrededor de 60º se fusifica al mismo tiempo como media y baja. A partir de los 135º grados en adelante DD se considera alto con una probabilidad de 1. La Figura 3.7 muestra la gráfica resultante de aplicar lógica difusa a este parámetro. En dicha figura representamos en la coordenada X los grados correspondientes al movimiento según GPS, y en la coordenada Y la probabilidad que determinará si el mensaje se firma para ser agregado o no en función de la coincidencia o no entre las direcciones del mensaje y de la información recibida del entorno. En este caso las reglas de control se basan en dichas coincidencias.

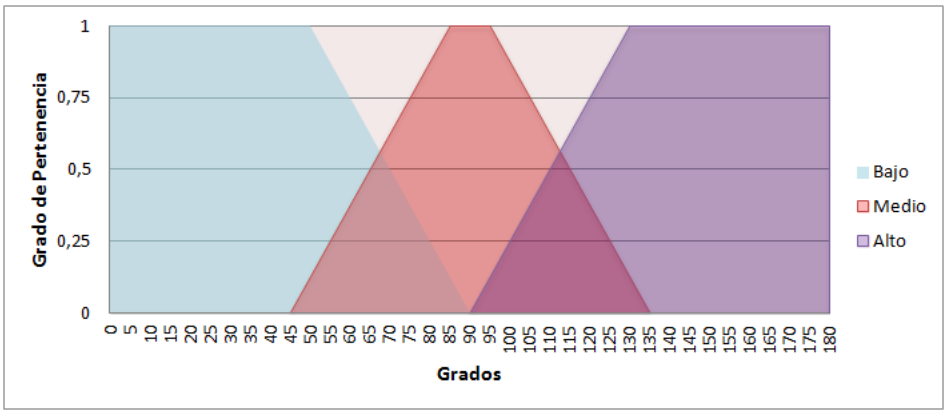


Figura 3.7: Lógica Difusa. Grados

A continuación se muestran las reglas que expresan la influencia de este parámetro.

Algoritmo Reglas Difusas sobre DD

01: **if** (DD **es** Bajo) **then**

02: Decisión-Agregar **es SI**;

```
03: else
04:   if (DD es Alto) then
05:     Decisión-Agregar es NO;
06:   else
07:     if (DD es Medio) then
08:       Decisión-Agregar según otros parámetros;
09:     endif
10:   endif
11:endif
```

3.3.5. Reglas de Control

Tras el análisis de cada una de las cuatro dimensiones sobre un paquete de información recibida y la información percibida, la regla de control sobre la decisión de si agregar/firmar o no vendrá determinada por la coincidencia de las conclusiones sobre la acción a realizar, es decir, que si del análisis de las cuatro dimensiones se concluye que se debe agregar ésa será la decisión final. Por el contrario si en algunas de las dimensiones se concluyó que no debía agregarse, entonces esa es la decisión final.

3.4. Agregación de Datos con Verificación Probabilística

En esta sección se describe en detalle un nuevo esquema de agregación de datos que contiene un mecanismo de seguridad basado en grupos reactivos creados bajo demanda para asegurar a priori que los vehículos generan información confiable, y un sistema de verificación probabilística para detectar intentos de ataque a posteriori de una manera eficiente y con un consumo mínimo de tiempo.

3.4.1. Zonas Geográficas

El concepto de zona geográfica es clave para comprender la propuesta del mecanismo basados en grupos. Debido a las características específicas de las redes vehiculares, como son la alta movilidad y el cambio frecuente de topología, es especialmente difícil proteger los

datos en estas redes. Por lo tanto, los mecanismos de seguridad propuestos para este entorno no deben suponer la existencia de una infraestructura estable y centralizada, sino sólo la existencia de nodos móviles funcionales dentro de la red. En el esquema de agregación de datos propuesto a continuación, se consideran tres comportamientos diferentes de los nodos o vehículos:

- Vehículos que detectan un evento en su ruta y generan automáticamente un mensaje de advertencia.
- Vehículos que reciben un paquete de advertencia y pueden confirmar directamente que se corresponde con un evento verdadero.
- Vehículos que reciben paquetes de advertencia sobre la existencia de un evento y la confirmación correspondiente, pero no tienen contacto directo con el evento recibido.

En la mayoría de los casos, la información generada en un determinado lugar en una VANET no es de interés fuera de cierto radio de distancia. Por ejemplo, si ocurre un accidente en el centro de la ciudad, en la mayoría de los casos no tiene ningún sentido que el mensaje de aviso alcance un pueblo o ciudad vecina. En consecuencia, en este trabajo se definen tres zonas geográficas diferentes dependiendo de donde se considere de interés el mensaje de advertencia recibido. En particular, dependiendo de la zona geográfica donde el nodo recibe el paquete se ejecutará una u otra parte del protocolo de agregación. Como se muestra en la Figura 3.8, se definen tres zonas geográficas diferentes respecto a la posición donde se genera el evento:

- Zona de Peligro: Es el área definida por la zona más cercana al evento, donde, el evento puede ser detectado directamente por los vehículos que se encuentran allí.
- Zona de Incertidumbre: Donde los nodos no pueden detectar el evento directamente, pero necesitan tomar decisiones con rapidez porque en un corto período de tiempo pueden entrar en la zona de peligro.
- Zona de Seguridad: Donde los nodos no pueden confirmar la información directamente, pero tienen tiempo suficiente para recoger evidencias sobre la existencia del evento en forma de paquetes agregados.

El particular el tamaño del radio de estas zonas es fijado por el nodo origen dependiendo de factores como el tipo de carretera y el tipo de evento detectado.

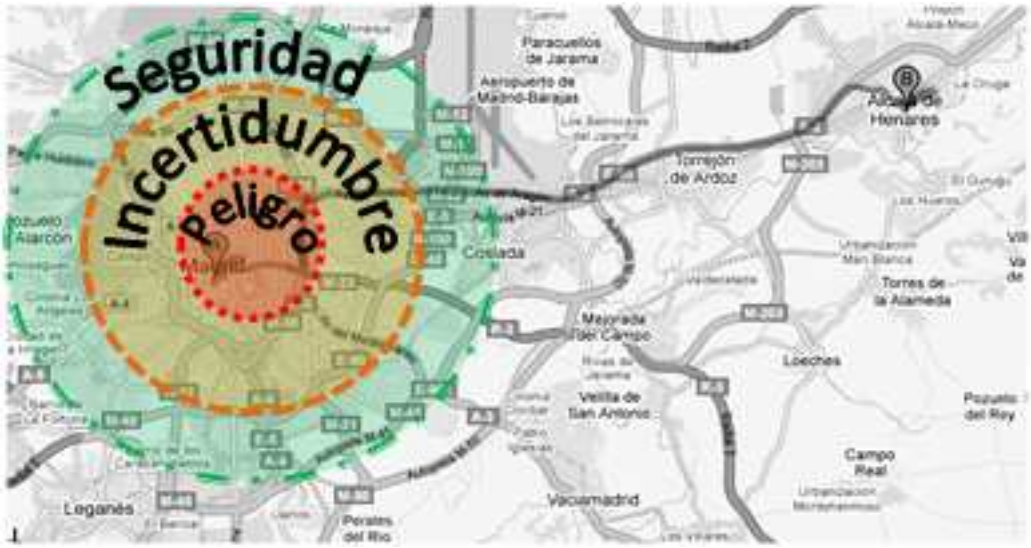


Figura 3.8: Zonas Geográficas

Los mensajes de advertencia agregados y firmados sólo pueden ser producidos por los vehículos que se encuentran dentro de la zona de peligro, mientras que la comprobación de mensajes agregados sólo se realiza por los vehículos que no son capaces de verificar directamente la información que les llega, es decir, por los vehículos que se encuentran fuera de la zona de peligro. En particular, cuando uno de estos vehículos recibe un mensaje de advertencia acerca de un incidente que no está bajo la cobertura de sus antenas, y quiere confirmar la autenticidad del mensaje recibido, tiene que actuar de manera diferente según la zona geográfica donde se encuentre:

- En la zona de incertidumbre, la decisión se debe tomar de forma rápida de manera que permita notificar al conductor con suficiente antelación, y además pueda cerciorarse de que la información recibida es verdadera. En este área, si un vehículo recibe un paquete de agregación firmado, el sistema debe utilizar un mecanismo de verificación lo suficientemente rápido como para comprobar todas las firmas contenidas en el paquete. Sin embargo, como se mencionó anteriormente, por un lado no es eficaz verificar todas las firmas contenidas en el paquete, pero por otro lado es necesario

comprobar la información antes de aceptarla como válida. Con el fin de solucionar este problema, se propone en el esquema verificar sólo algunas firmas.

- En la zona de seguridad, dado que esta región está bastante alejada del lugar donde se generó el evento, el vehículo receptor tiene la posibilidad de recibir diferentes mensajes agregados acerca del mismo evento antes de tomar una decisión. En este caso, al igual que en el caso anterior, el vehículo tiene que verificar las firmas de los paquetes recibidos, pero además los vehículos pueden realizar otras verificaciones de manera que obtengan un mayor nivel de fiabilidad sobre la veracidad de la información recibida. Al estar en esta área, es posible recibir diferentes paquetes de advertencia sobre el mismo evento. Teniendo en cuenta que estos paquetes son generados independientemente, cuanto mayor sea el número de paquetes de alerta recibidos y verificados, mayor será la fiabilidad y exactitud de la información proporcionada. Una verificación adicional que los vehículos en esta zona pueden realizar es la de recreación de las distintas celdas, donde los paquetes recibidos pudieron ser creados a partir de los datos contenidos en el paquete. Este sencillo procedimiento proporciona información de valor añadido pues todas las firmas contenidas en un mismo paquete deben corresponder a la misma celda ofreciendo un modo adicional de detectar nodos maliciosos.

3.4.2. Grupos Reactivos

Con el fin de proporcionar información en tiempo real y confiable sobre la existencia de eventos en la carretera, y teniendo en cuenta que no existe infraestructura técnica para coordinar los vehículos de manera que formen grupos, proponemos el establecimiento de grupos reactivos de forma totalmente autorganizada. Así, los grupos se forman sólo cuando sean necesarios, y se evita que el tamaño del paquete crezca infinitamente, ya que implica un límite en el número de firmas contenidas en él. Aquí se propone un mecanismo donde la formación de grupos no es requerida a priori sino que cuando un vehículo detecta un evento automáticamente trata de formar un grupo con todos los vehículos que se encuentran dentro de su rango, lo que define la zona geográfica de peligro centrada en el evento. Si no existiera dicha formación y todos los vehículos que detectan un evento enviaran el mensaje

de advertencia, en entornos densos la sobrecarga de las comunicaciones en la red sería muy elevada. De esta manera, la organización de los vehículos en grupos reactivos para generar información agregada permite reducir el envío de advertencias repetidas. La posición en la que se localiza el evento se define como el centro de la zona de peligro, y a su vez esta zona se divide en celdas para formar grupos. El líder de cada grupo será el encargado de construir el mensaje de advertencia firmado y agregado.

Como se ha mencionado, los grupos propuestos en este trabajo son reactivos y creados bajo demanda, ya que sólo se forman una vez se detecta el evento en la carretera. En particular, cuando un vehículo detecta un evento estático, genera un paquete con información sobre el evento, tales como sus coordenadas geográficas (X, Y, Z), fecha y hora de generación del paquete, sentido de circulación del tráfico, etc. Este paquete se transmite a todos los nodos que están en el rango de la zona de peligro. Por ejemplo, el radio puede ser de 100 metros, lo que coincide más o menos con el rango de transmisión de una red Wi-Fi. Esta zona de peligro se divide en celdas en las que se crean los grupos reactivos, como se muestra en la Figura 3.9. Por lo tanto, una celda es un área geográfica limitada por el número máximo de vehículos que pueden formar un grupo, mientras que un grupo es un conjunto de vehículos dentro de una celda, que produce un paquete agregado. En un caso límite, el número de vehículos dentro de un grupo se corresponde exactamente con el número máximo de vehículos que caben en una celda. Puesto que las dimensiones tanto de las celdas como de la zona de peligro se incluye en el mensaje, todos los nodos receptores en la zona de peligro definen las mismas celdas con respecto a las coordenadas del evento (X, Y, Z).

Cuando un nodo recibe un paquete de advertencia P, de un nodo con identidad ID ejecuta el algoritmo de formación de grupo reactivo. En primer lugar, comprueba si su vehículo está dentro de la zona de peligro definida por el paquete. A continuación, comprueba si es capaz de detectar el evento, y en ese caso, calcula las dimensiones de la celda a la que pertenece de acuerdo a las coordenadas (X, Y, Z) de modo que, si no ha recibido ninguna solicitud de formación de grupo, inicia una nueva solicitud para todos los nodos dentro de su celda. En caso de recibir varias solicitudes los vehículos siempre deben elegir solicitudes con la marca de tiempo más antigua. Es posible aunque raro, que dos o más

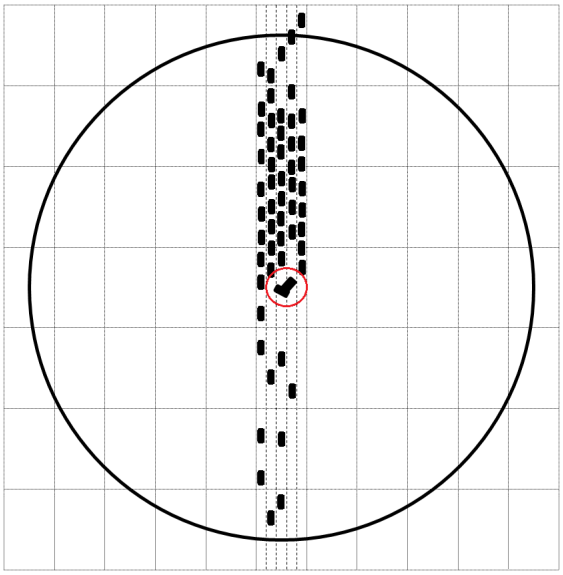


Figura 3.9: Celdas en la Zona de Peligro

vehículos inicien la solicitud de formación de grupo al mismo tiempo, pero este problema tiene fácil solución, ya que en ese caso se elige la solicitud correspondiente al vecino más cercano al centro de la celda. El nodo seleccionado se convierte en el líder del grupo reactivo y es el encargado de generar el paquete agregado correspondiente. Una vez que el grupo se crea, el líder firma el mensaje de advertencia y lo envía a todos los miembros de su grupo.

Algoritmo Formación de Grupo Reactivo

```
01: function Main(Paquete P, Nodo ID)
02:   boolean PeticionFormacionGrupo= verdadero;
03:   //Se comprueba si es un nuevo mensaje de advertencia
04:   If (NuevaAdvertencia(P)) then
05:     // Se obtiene la información del paquete
06:     double X = P.X;
07:     double Y = P.Y;
08:     double Z = P.Z;
09:     string TipoEvento = P.TipoEvento;
10:     string Direccion = P.Direccion;
```

```
11:     string Via = P.Via;
12:     double MarcaTiempo = P.MarcaTiempo;
13:     //Se comprueba si el vehículo está dentro de la zona de peligro
14:     if (EnZonaPeligro(X,Y,Z,TipoEvento,Direccion,Via,MarcaTiempo)) then
15:       //Comprueba si es capaz de detectar el evento
16:       if (DetectaEvento(X,Y,Z,TipoEvento,Direccion,Via)) then
17:         //Calcula la dimensión de su celda
18:         double Celda = DimensionesCelda();
19:         // Si no ha recibido petición de formación de grupo
20:           // de su celda, lanza una
21:         if (!PeticionFormacionGrupo()) then
22:           //Envia una petición de formación de grupo
23:           EnviaPeticion(X,Y,Z,TipoEvento,Direccion,Via,MarcaTiempo);
24:         end if
25:       else
26:         MarcarMalicioso(ID); //Identifica un intento de ataque
27:       end if
28:     else
29:       //No puede detectar el evento
30:     end if
31:   end Main
```

Cuando los vehículos del grupo reciben un mensaje de advertencia firmado por el líder, comprueban si pueden validar la información recibida mediante la verificación de datos como la hora del evento reportado y si están en el rango de detección. Si cumplen con los requisitos anteriores pero no detectan el caso reportado, se podría considerar como un intento de ataque. En este caso, los nodos marcan al nodo líder como malicioso según el esquema propuesto en el capítulo de cooperación y descartan el paquete. Por otro lado, si están de acuerdo con la información recibida, firman el mensaje y lo envían de vuelta al líder. Las firmas recibidas proporcionan evidencias sobre la existencia del evento, por

lo que el líder agrega todas las firmas y genera un mensaje de advertencia agregado. Esto implica que el líder crea un paquete firmado por los diferentes vehículos de su grupo que alertan sobre el mismo evento, de forma que dichas firmas pueden ser utilizadas como prueba de que el contenido de la información es verdadero. Finalmente, el líder transmite este mensaje agregado y todos los receptores primero verifican las firmas de acuerdo con el esquema descrito más adelante, y luego almacenan y retransmiten el mensaje.

Puede ocurrir que un nodo esté solo en su grupo en el momento de la detección de un evento. En este caso el nodo genera un paquete con una sola firma y lo envía como paquete agregado. Como el paquete no tiene suficientes evidencias sobre la veracidad de la información, no se tiene en cuenta en la zona de incertidumbre. Sin embargo, cuando el paquete alcanza la zona de seguridad puede ser usado en combinación con otros mensajes de advertencia recibidos anteriormente alertando sobre el mismo evento.

3.4.3. Tipo de Paquetes

La propuesta de agregación de datos descrita en esta sección se basa en cuatro tipos de paquetes:

- Paquete tipo W (Advertencia): Paquete de advertencia generado por el nodo origen tras detectar un evento, que se retransmite hasta alcanzar a todos los nodos que se encuentran dentro de la zona de peligro. Este paquete contiene el mensaje M formado por las coordenadas (X_0, Y_0, Z_0), tipo de evento, sentido de circulación del tráfico, nombre de la vía y marca de tiempo con la fecha y hora en la que se detectó el evento, junto con el ID_0 del nodo origen y la firma de M usando la clave privada Pr_0 del nodo.

- Paquete tipo F (Petición): Corresponde a la solicitud de formación de grupos después de la recepción de un paquete de tipo W. Este paquete es enviado a todos los vehículos de su celda por un nodo que se propone como líder del grupo reactivo. En este caso, el paquete contiene el mensaje M recibido, junto con el ID_1 del líder, la firma del M, y sus coordenadas y marca de tiempo para permitir que otros nodos puedan determinar si el nodo emisor debe ser el líder del grupo o no. El nodo que envía este

paquete determina si otro nodo pertenece o no a su grupo gracias a las coordenadas geográficas por lo que esperará hasta recibir todas las firmas de los nodos que forman parte de su grupo.

- Paquete tipo S (Firmado): Este paquete contiene la firma de un nodo i que esté de acuerdo con la información recibida en el paquete tipo F, tanto con el evento de advertencia como con la información del grupo. El paquete contiene el mensaje M y los datos correspondientes del remitente, y es enviado por el nodo i al líder de su grupo.

- Paquete A (Agregado): Contiene todas las firmas que ha recibido el líder en forma de paquetes tipo S por parte de todos los miembros de su grupo con el fin de proporcionar una mayor evidencia sobre la existencia de un evento recibido. Este paquete es enviado por el líder y retransmitido a través de la red.

La Figura 3.10 muestra un resumen sobre el contenido de cada tipo de paquete.

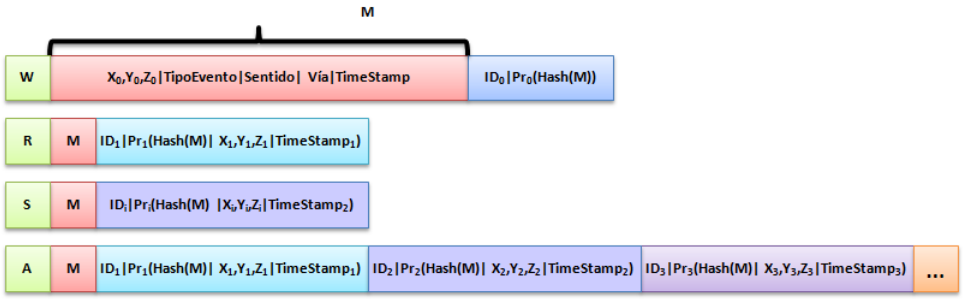


Figura 3.10: Tipo de Paquetes

3.4.4. Tamaño de Celdas

La división de la zona de peligro en celdas se ha propuesto como una forma de evitar la sobrecarga del canal por la generación de paquetes muy grandes que contienen todas las firmas de todos nodos que son capaces de detectar un mismo evento en una zona. Los paquetes agregados producidos en cada celda deben proporcionar evidencia suficiente acerca de la validez de los eventos informados. Por lo tanto, el tamaño de la celda debe

ser elegido cuidadosamente a fin de abordar ambas cuestiones, ya que está directamente relacionado con el tamaño del paquete y con el número de firmas que contiene. De hecho, el tamaño de la celda define el número mínimo y máximo de vehículos que puede contener, lo que determina, respectivamente, la fiabilidad y el máximo tamaño del paquete de tipo A correspondiente.

De acuerdo con [75], el tamaño óptimo de celda para una VANET es de tamaño fijo: 16 metros de ancho y 126 metros de largo. Sin embargo, en un entorno dinámico como es una VANET, estos parámetros deben ser también dinámicos y no fijos para adaptarlos a las diferentes características del entorno. En una VANET, los vehículos se ven afectados por diferentes factores que varían mientras circulan, como puede ser el número de carriles de la carretera, la velocidad y la densidad de vehículos soportada por los diferentes tipos de carretera, etc. Teniendo en cuenta estas características cambiantes, es necesario definir criterios específicos para determinar el tamaño de una celda basados en esa información.

Para los nodos es imposible determinar exactamente cuántos vehículos hay en la carretera por la que circulan en cada momento, y la distancia y la velocidad de cada uno de esos vehículos. Sin embargo todas las vías tiene una velocidad máxima permitida y los vehículos deben respetar una distancia mínima de seguridad con el resto de vehículos, lo que da una idea del número máximo de vehículos que puede haber en una determinada zona mientras los vehículos circulan. En caso de tratarse de una vía sin restricción de velocidad, fijaremos este valor a 120 km/h. En base a estos parámetros se propone a continuación una medida para calcular el tamaño de las celdas.

El objetivo es encontrar un tamaño de celda óptima que minimice el tamaño de los mensajes agregados al tiempo que maximice la fiabilidad de la información. En las simulaciones realizadas en este trabajo y comentadas más adelante, se estableció, a partir de valores obtenidos experimentalmente que la longitud de cada celda fuera dos veces la distancia de seguridad, valor que puede ser fácilmente calculado pues se corresponde con el cuadrado de la velocidad de la carretera, mientras que su ancho es el número de carriles de la carretera, multiplicado por el ancho de cada carril, aproximadamente 4 metros lo que proporciona el área final de la celda:

$$AreaCelda = (2 * safety\_distance) * (4 * number\_of\_lanes)$$

Esta expresión produce aproximadamente el tamaño de celda óptimo definido en [82] para cuatro carriles en una carretera con velocidad máxima permitida de 80 km/h.

De esta manera, la zona de peligro puede ser dividida en celdas, de modo que el centro de la zona de peligro (X_0, Y_0, Z_0) se corresponda con el centro de la celda central. Las otras celdas se calculan a partir de la celda central, como se muestra en la Figura 3.9.

El número máximo de firmas que se pueden generar dentro de un grupo en cualquiera de estas celdas es fácil de calcular. Por ejemplo, en una carretera de tres carriles con límite de velocidad 120 km/h, en cada celda con AreaCelda=3456 m^2 se genera un máximo de 9 firmas por celda en condiciones normales de tráfico. Téngase en cuenta sin embargo que la influencia del valor AreaCelda cuando las condiciones no son las de la fórmula de aproximación, es muy alta. Por ejemplo, en caso de existir una densidad alta de nodos como en un atasco de tráfico, se podrían generar alrededor de 170 firmas en una misma celda, mientras que en casos de densidad baja de tráfico, puede ocurrir que exista un solo vehículo en la celda y que no se puedan generar más firmas. Sin embargo, si las condiciones de tráfico no son estables, no se debe ni reducir el AreaCelda para reducir el número de firmas dentro de la misma porque los vehículos podrían moverse demasiado rápido impidiendo la posibilidad de formación de grupos, ni aumentar el AreaCelda para aumentar el número de firmas posible porque eso podría generar paquetes demasiado grandes. Por otro lado, la fórmula AreaCelda no debe depender de las condiciones del tráfico, porque éstas pueden variar con el tiempo, y su valor tiene que estar claramente definido para todos los nodos utilizando solo coordenadas geográficas.

Una solución para el caso de densidad alta de vehículos, donde se generan muchas firmas dentro de una misma celda, podría basarse en un enfoque probabilístico para la agregación de datos de tal manera que en lugar de generar un paquete de agregación completo, se eligiesen unas cuantas firmas entre las recibidas por los nodos de la celda al azar para que sean sólo éstas las que se incluyan en el paquete probabilísticamente agregado resultante y lograr así que su tamaño se reduzca.

3.4.5. Verificación Probabilística

La verificación probabilística de las firmas contenidas en un paquete tipo A es la segunda parte de nuestra propuesta. En particular, el aspecto probabilístico se utiliza para la elección de firmas a verificar. Una vez que se eligen las firmas, su verificación se realiza a través del método tradicional basado en una infraestructura de clave pública (PKI, Public Key Infrastructure), que implica la obtención de los certificados de clave pública de las firmas, y la ejecución de tres pasos: cálculo del valor hash del mensaje firmado, descifrado de la firma digital con la clave pública del remitente, y comparación entre ambos valores. Ya que en VANET no es posible confiar en los mecanismos que requieren un sistema centralizado, como una autoridad de certificación centralizada, la certificación de clave pública se realiza a través de una solución distribuida, como la descrita en [26].

La verificación probabilística solo se lleva a cabo por los vehículos que se encuentran fuera de la zona de peligro porque estos vehículos son incapaces de verificar directamente la información contenida en los paquetes recibidos, y la única información que tienen al respecto es la recibida a través de los paquetes tipo A. El algoritmo de verificación probabilística propuesto utiliza hilos, que son procesos ligeros que permiten la ejecución concurrente y por lo tanto una ejecución más rápida de todo el protocolo. En el algoritmo que se muestra a continuación, Th[i] denota un hilo para la variable i que toma un valor entero entre 1 y n, donde n denota el número de firmas agregadas en el paquete recibido. Cuando un vehículo recibe un mensaje agregado, el proceso principal lanza tantos hilos como firmas contiene el paquete recibido, pero antes de eso, comprueba si hay firmas suficientes para determinar si el mensaje es fiable por haber sido confirmado por un número significativo de vehículos diferentes. Este número será fijado por el nodo de origen en función de la densidad del tráfico. En nuestras simulaciones e implementaciones reales, si el paquete recibido tiene menos de tres firmas es descartado. En caso contrario cada hilo Th[i] determina si hay que verificar la firma S correspondiente a la posición i con una probabilidad Prob(Th[i]), que se expresa en porcentajes con un número aleatorio entre 0 y 99. Si Th[i] realiza una verificación, y comprueba que la firma es válida, devuelve como valor verdadero informando que se trata de una firma válida. De lo contrario, devuelve como valor falso. El resultado

de las comprobaciones que realizan los diferentes hilos se almacenan en una estructura St. Si al terminar las comprobaciones todos los campos de la estructura St son verdaderos, se interpreta como evidencia de que todas las firmas verificadas son correctas por lo que el mensaje es aceptado como válido. Por otro lado, si St contiene algunos campos con contenido falso, esto es interpretado como que se trata de un mensaje falso. De hecho antes de firmar un paquete, los nodos legítimos deben comprobar si pueden validar la información y de lo contrario, descartar el paquete.

Prob(Th[i]) será discutida en detalle en la siguiente sección, donde se explica el motivo del límite específico que se incluye en el algoritmo de verificación probabilística de firmas.

Algoritmo Verificación Probabilística de Firmas

```
01: function Main(...)
02:   bool St[n];
03:   Thread Th[n];
04:   for (i=0;i<n;i++) do
05:     ThreadStartH[i].CompruebaFirmas(n,St);
06:   end for
07:   if (TodasVerdaderas(St)) then
08:     return MensajeConfiable;
09:   else
10:     return NoMensajeConfiable;
11:   end if
12: end Main

13: bool function CompruebaFirmas(n, St)
14:   int j=0
15:   if(n > 3)
16:     for (i=0;i<n;i++)
17:       Prob(Th[i])=rand(0..99);
18:       if (Prob(Th[i]) > ((1-10/n)*100)) then
```

```
19:          M=RecuperaMensaje(); //Función que recupera el mensaje
20:          S=RecuperaFirma(i); //Función que recupera la firma del paquete
21:          St[j]=(VerificaFirma(S,M));
22:          j++;
23:        end if
24:      end for
25:    else
26:      //No hay firmas suficientes a verificar
27:      return NoSuficientesFirmas;
28:    end if
29: end function

30: bool function CompruebaFirmas(Signature S,text M)
31:    if (IsValid(S)) then
32:      return verdadero;
33:    else
34:      return falso;
35:    endif
36: end function
```

3.4.6. Análisis de las Probabilidades

En este apartado se realiza una evaluación analítica de un parámetro importante de la propuesta. En particular, se estudia a fondo la verificación probabilística. Se incluye una breve discusión sobre posibles ataques y las contramedidas llevadas a cabo. Para garantizar la validez de un mensaje específico, una primera aproximación sería que al menos un hilo realice la verificación de una firma. De esta manera, la probabilidad de que al menos un hilo realice alguna verificación de firmas debe ser lo más cercano a 1 como sea posible. Sin embargo, desde el punto de vista de la agregación, el hecho de que sólo un hilo realice la verificación de una firma no es suficiente. Supongamos que se recibe un mensaje con varias firmas de modo que sólo una de ellas es verdadera y el resto son falsas. Entonces, si el hilo

Th[i] sólo comprueba la firma real, daría como resultado que el mensaje es válido. Por lo tanto, más de un hilo debería realizar la verificación de las firmas del mensaje enviado por un vehículo. A continuación se analizan los valores óptimos para cumplir con esta restricción.

Cada hilo puede ser visto como un experimento independiente de Bernoulli que con probabilidad p dada por el porcentaje Prob(Th[i]) produce la verificación de la firma iésima de un paquete agregado [53]. Sea X la variable dada por el número de éxitos de los n hilos, que sigue una distribución binomial de parámetros n y p. Por lo tanto, la probabilidad del evento B, según el cual existen al menos dos hilos que realizan la verificación de firmas de los paquetes, puede ser expresada en función de n y p de forma siguiente:

$$Prob\{B\} = 1 - (1-p)^n - n \cdot p \cdot (1-p)^{n-1} \tag{3.3}$$

En la ecuación 3.3, $P\{X = 0\} = (1-p)^n$ es la probabilidad de que ninguna de las firmas sea verificada, y $Prob\{X = 1\} = n \cdot p \cdot (1-p)^{n-1}$ es la probabilidad de que exactamente una de las n firmas sea verificada.

El objetivo es hacer $Prob\{B\}$ lo más cercano a 1 como sea posible. La Figura 3.11 muestra la relación entre $Prob\{B\}$, p y n. Se puede ver que $Prob\{B\}$ crece a medida que p o n crecen y rápidamente se aproxima a 1. El objetivo es elegir un valor de p que haga que $Prob\{B\}$ se aproxime lo máximo posible a 1 para un valor fijo de n, y al mismo tiempo que su valor sea lo más pequeño posible porque un valor pequeño de p implica que un vehículo puede potencialmente ahorrar tiempo de procesamiento. En conclusión, el parámetro p debe ser elegido adecuadamente de modo que ambas condiciones se cumplan. Esta relación se muestra a continuación.

3.4.7. Discusión sobre el Tamaño del Paquete

Cuando se elige el número de firmas a introducir en un paquete, se debe tener en cuenta tanto el tamaño máximo de paquete que puede soportar una VANET como el número mínimo de firmas necesario para asegurar que el contenido de la información es cierto. De acuerdo con la primera condición, y considerando la capacidad del canal inalámbrico en una VANET, el tamaño de los paquetes puede ser de 256 bytes a 1500 bytes. Teniendo

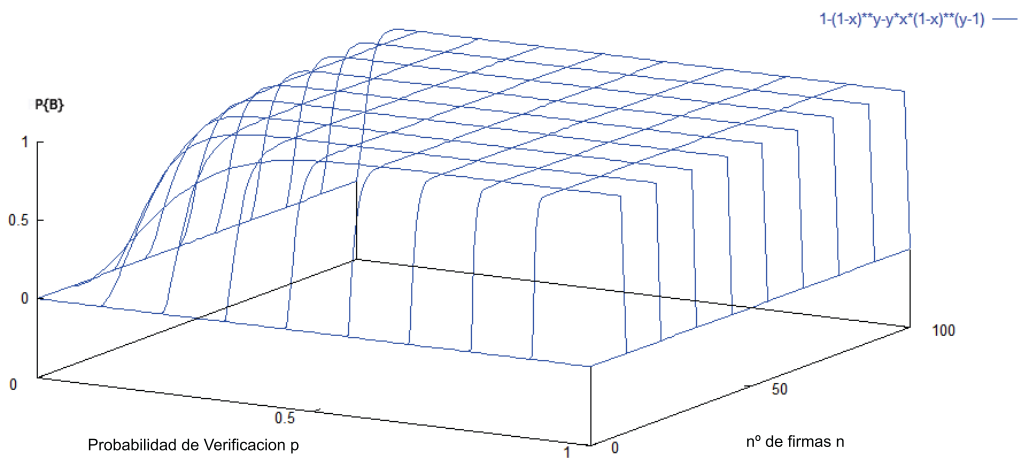


Figura 3.11: Probabilidad de Verificar al Menos dos Firmas

en cuenta que en este tipo de redes se pueden generar un gran número de paquetes, sería aconsejable no utilizar el tamaño máximo de paquete permitido porque en ese caso, un pequeño número de paquetes podría saturar el canal. En este trabajo consideramos el uso de aproximadamente 100 bytes para el contenido del mensaje y el resto para las firmas, por lo que disponemos de una capacidad de 156 bytes para las firmas en el peor de los casos y de 1400 bytes en el mejor. Dado que el resultado de cifrar con clave privada apenas modifica el tamaño de la entrada, sólo se considera el resultado de aplicar la función hash al mensaje. Por ejemplo, la función hash SHA-1 [113] produce una salida de 20 bytes, por lo que con ella se podría generar siete firmas para paquetes de 256 bytes y 70 firmas para paquetes de 1.500 bytes. Estos datos se toman como punto de partida para la discusión sobre los valores óptimos de los parámetros.

Con el fin de elegir un valor apropiado de p para los diferentes valores de n, la variable $k = n \cdot p$ se podría utilizar para reflejar la relación inversamente proporcional de p y n. Téngase en cuenta que k representa el número medio de firmas que un vehículo verifica dado que n es el total de firmas en el paquete y p es la probabilidad de verificación. Si podemos encontrar un valor adecuado de k, entonces se podría determinar el valor de p correspondiente. Basándonos en la ecuación (3.3), se puede obtener la relación entre $Prob\{B\}$ y n para diferentes valores de k, permitiendo determinar el valor de p. Dado que

la probabilidad de p tiene un valor máximo de 1, si se utiliza SHA-1, n sería superior a 6, por lo que se puede utilizar este valor para concluir haciendo uso de la Figura 3.12 que con $k = 6$, $Prob\{B\}$ es cercana a 1, pero no lo suficiente. Sin embargo, con $k = 10$, $Prob\{B\}$ es suficientemente cercana 1 cuando el paquete contiene 10 o más firmas. Por lo tanto, k se puede ajustar a un valor constante, por ejemplo, $k = 10$, y una vez que k es fijo, p se puede calcular a partir de k/n de manera que toma el valor $10/n$, que es el límite utilizado en el algoritmo para la verificación probabilística de firmas. En otras palabras, podemos expresar p en términos de n. Por ejemplo, un vehículo que recibe un mensaje con 20 firmas, verifica cada firma con una probabilidad de 0,5. Por otra parte, si n es inferior a 10, p es ≈ 1, lo que parece correcto porque con tan pocas evidencias sobre la existencia de un evento, todas las firmas deben ser verificadas para evitar ataques.

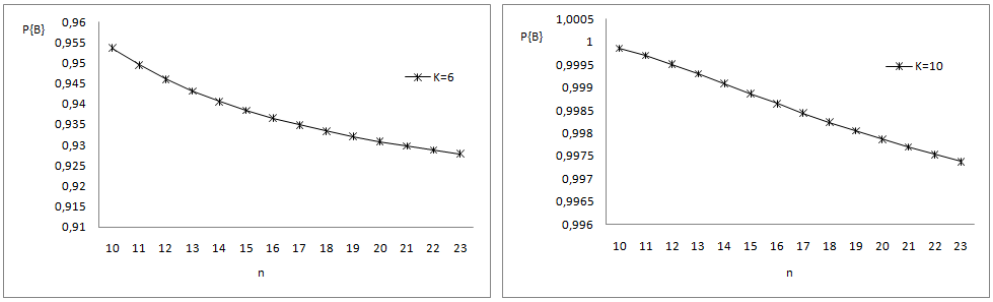


Figura 3.12: Probabilidad de Verificaciones suficientes with k=6 y k=10

Teniendo en cuenta el número mínimo de firmas que puede contener un paquete para maximizar la probabilidad $Prob\{B\}$, y calculando las probabilidades y el número máximo de firmas que caben en un paquete, se puede discutir sobre la función hash óptima a utilizar en este tipo de redes para el esquema propuesto. La Tabla 3.1 muestra los datos correspondientes a diferentes funciones hash. Teniendo en cuenta estos datos, MD5 [122] produce firmas de 16 bytes, SHA-1 de 20 bytes y SHA-256 de 32 bytes. En consecuencia, si se utilizan 100 bytes para el contenido del mensaje, la tabla muestra el número máximo de firmas que se pueden añadir en cada paquete agregado. Por ejemplo, un paquete de 512 bytes puede incluir un total de 20 firmas con SHA-1. Incluso el uso de la función hash SHA-256, con paquetes de 512 bytes es posible y aumenta la seguridad del sistema.

Tabla 3.1: Parámetros de Función Hash para un Mensaje de 100 bytes

Función Hash	**Tamaño del Paquete**	**Tamaño de la Fima**	**N. de Firmas**
MD5	256	156	9
SHA-1			7
SHA-256			4
MD5	512	412	25
SHA-1			20
SHA-256			12
MD5	1024	924	57
SHA-1			46
SHA-256			28
MD5	1500	1400	87
SHA-1			70
SHA-256			43

3.4.8. Análisis de Seguridad

Con el fin de analizar la eficacia y robustez de la propuesta, se ha realizado una evaluación de seguridad de la propuesta. En esta sección se analizan brevemente algunos posibles ataques de adversarios y la forma en que el sistema los afronta.

En particular, se describen ocho ataques: ataque Sybil, generación de información falsa, eliminación de mensajes agregados, modificación de mensajes agregados, ataque de suplantación, ataque a la privacidad, falso aumento de la confianza sobre un evento y generación de registro falso.

- **Ataque Sybil.** Este tipo de ataque se produce cuando un nodo malicioso crea diferentes identidades falsas en el sistema con el fin de obtener una mayor influencia en la red. Nuestro sistema sólo permite una identidad por nodo utilizando como parámetro único de identidad el número de teléfono, por lo que este ataque es imposible.

- **Generación de información falsa.** Un atacante puede falsificar un mensaje que no corresponde con la información real de su entorno. Este caso es desestimado por el modo de funcionamiento de la agregación de datos propuesta ya que los otros vehículos sólo firmarán el mensaje si son capaces de detectar el mismo evento y las condiciones

que se especifican en el mensaje.

- **Eliminación de mensajes agregados.** Los atacantes pueden intentar descartar algunos mensajes agregados, lo que impide la difusión de información. Para resolver este problema se puede utilizar algún esquema de cooperación como los propuestos en el capítulo anterior. De todos modos, el daño que pueda resultar de la eliminación de uno o varios paquetes de datos de agregación no es muy alto, ya que normalmente en el sistema propuesto se genera más de un paquete agregado en relación a cada evento.

- **Modificación de mensajes agregados.** Un atacante podría modificar los mensajes agregados transmitidos a través de la red. Sin embargo, cuando un vehículo no tiene ningún contacto directo con la información contenida en un mensaje recibido, que no está en la zona de peligro, tiene que realizar la verificación de firmas. En primer lugar, un vehículo que recibe el paquete en la zona de incertidumbre o de seguridad debe comprobar algunas firmas de acuerdo a la probabilidad de verificación de modo que se correspondan con el contenido del mensaje. Además, los vehículos en la zona de seguridad deben verificar la existencia de diferentes paquetes agregados provenientes de diferentes grupos reactivos advirtiendo sobre un mismo evento. En cualquier caso, la generación de mensajes agregados falsos es detectada con una alta probabilidad.

- **Ataque de suplantación.** Con el fin de proteger el esquema de posibles ataques donde un nodo ilegítimo pretende hacerse pasar por un nodo legítimo, se utilizan tanto un esquema de clave pública robusto como un protocolo de autenticación fuerte.

- **Ataque a la privacidad.** La privacidad es una de las preocupaciones más importantes en VANETs. Los sistemas de comunicación basados en PKI no protegen la privacidad de los nodos porque la difusión de cualquier mensaje normalmente incluye la firma y certificados del nodo. Aunque estos datos no contienen información explícita sobre la identidad del emisor, el receptor puede seguirle la pista. Esta cuestión se ha tenido en cuenta en la implementación de la propuesta descrita en el siguiente capítulo haciendo uso de pseudónimos variables en los beacons retransmitidos. Con esta idea de pseudónimos cambiantes la vinculación de la identidad de un nodo con su

pseudónimo sólo es posible después de una autenticación bidireccional fuerte. De esta manera, el seguimiento de los movimientos de un vehículo no es posible en el esquema propuesto.

- **Falso aumento de la confianza.** Un nodo legítimo que no es capaz de detectar un evento podría tratar de sumar su firma al paquete agregado a fin de aumentar la confianza sobre el evento. Este ataque no sirve de nada en el esquema propuesto porque la verificación de una sola firma no es suficiente para aceptar la validez del mensaje.

- **Generación de Registro Falso.** Un líder puede tratar de añadir una firma falsa a un paquete agregado sobre un hecho real con el fin de que otros nodos descarten el paquete como falso. Este tipo de ataque sería fácilmente detectado por los nodos que se encuentran en la zona de peligro ya que pueden detectar correctamente el evento.

3.4.9. Evaluación del Rendimiento

En esta sección se incluye un análisis de la propuesta para comprobar la eficacia de los procesos de agregación y de verificación, mediante simulaciones en NS-2. Con el fin de analizar la rapidez y efectividad del módulo de agregación de datos, se ha realizado una simulación basada en los datos obtenidos de la implementación en dispositivos reales que se presenta en el capítulo 4. Esta sección incluye algunos detalles y resultados obtenidos para un promedio de 100 simulaciones con diferentes tamaños de red sobre un área de 1000 metros cuadrados, es decir, teniendo en cuenta diferentes situaciones con respecto a la densidad del tráfico. Debido a las limitaciones de cómputo de los ordenadores utilizados, las simulaciones están formadas por redes de entre 10 y 40 nodos. Los parámetros más relevantes utilizados para las simulaciones han sido: número total de carriles para cada sentido = 3, tiempo de simulación = 1000s, momento en el que comienzan los movimientos= 0s, momento en que comienza la retransmisión = 40s, período de retransmisión = 10s, rango de transmisión = 100m, distancia recorrida antes que se produzca el evento = 800m. Tanto la velocidad como el sentido de circulación de los nodos fueron aleatorios.

El objetivo es evaluar, por un lado el número de paquetes generados utilizando

grupos reactivos y por otro lado, los efectos de la propuesta en la complejidad computacional, y por lo tanto el tiempo que necesita el mecanismo de agregación propuesto para advertir a todos los nodos de la red sobre la existencia de un evento. Finalmente, se analiza el tiempo empleado en la verificación de firmas contenidas en un paquete de acuerdo con la verificación probabilística propuesta y considerando diferentes tamaños de paquetes. Con el fin de obtener datos se comparan las simulaciones de la propuesta con un esquema básico sin ningún mecanismo de agregación implementado.

Las primeras simulaciones corresponden a la detección de una congestión de tráfico en la carretera y la correspondiente retransmisión de paquetes de agregación utilizando y sin utilizar la idea de grupos reactivos, es decir, con y sin agregación. En la Figura 3.13 se puede ver que el número de paquetes generados en la simulación sin utilizar agregación es mucho mayor que cuando se utiliza el esquema de agregación propuesto, aunque en este caso el número incluye los paquetes generados por el algoritmo de formación de grupos. La disminución en el número de paquetes generados permite hacer un mejor uso del canal. Por lo tanto, la Figura 3.13 muestra cómo el uso de agregación con grupos reactivos puede reducir el número de paquetes generados y mejorar el uso del canal de forma significativa.

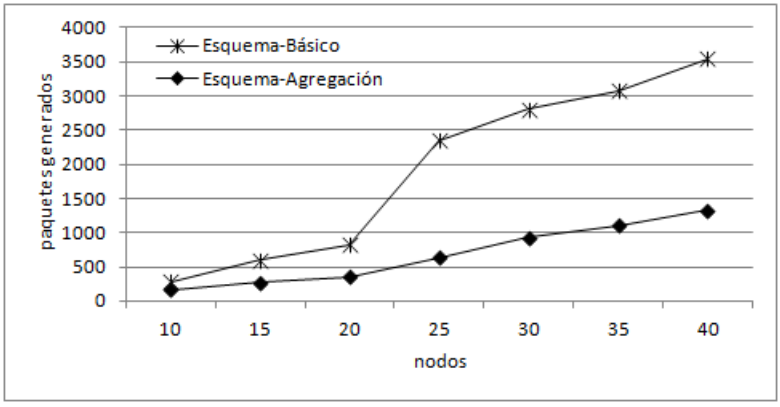


Figura 3.13: Número de paquetes Generados

El tiempo de conexión entre vehículos en VANETs es generalmente corto por lo que cualquier mecanismo propuesto requiere que la comunicación entre vehículos sea lo suficientemente rápida como para evitar la pérdida de datos durante las comunicaciones. En la Figura 3.14 podemos ver el impacto de la densidad de nodos en el coste de tiempo de

comunicación tanto con el esquema básico como en el esquema de agregación propuestos. La conclusión de esta simulación es que el uso del mecanismo de agregación propuesto no implica un aumento significativo en el coste de gestión del paquete recibido. Se puede ver que cuando el tamaño de la red está entre 10 y 20 nodos, el mecanismo funciona aún mejor. Este efecto parece sorprendente, pero se puede explicar por el hecho de que la comunicación es más organizada cuando se utiliza la agregación y la formación de grupos reactivos además permite alertar a un mayor número de nodos en menos tiempo. A medida que el número de nodos en la red aumenta, el tiempo necesario para procesar y generar los paquetes agregados para advertir a todos los nodos de la red también aumenta.

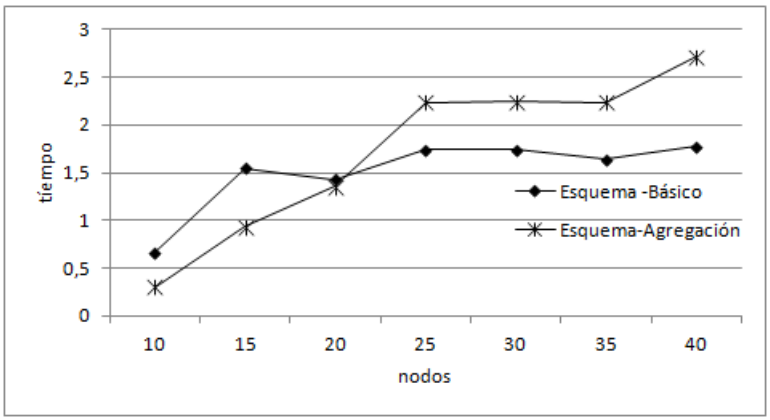


Figura 3.14: Coste de Tiempo

Otro aspecto delicado en la presente propuesta es la verificación de firmas y el retardo que conlleva. Como se ha comentado en apartados anteriores, no es práctico para los nodos verificar todas las firmas que contiene el paquete, por lo que en este trabajo se ha propuesto un algoritmo probabilístico en el que se verifican sólo algunas de ellas. Para analizar el retardo y averiguar si la propuesta mejora el tiempo de verificación se han realizado varias simulaciones donde se han tenido en cuenta los diferentes tamaños de paquetes y el número máximo de firmas que caben en cada uno de ellos con las diferentes funciones hash mencionadas en la sección 3.4.6.

La Tabla 3.2 muestra los resultados de las simulaciones. Los resultados muestran el tiempo promedio en minutos de 100 simulaciones para diferentes tamaños de paquetes y

número de firmas. Se puede ver que cuando el número de firmas contenidas en el paquete es menor que 10, el tiempo es aproximadamente igual en ambos casos, aunque nuestro método tiene un costo de tiempo ligeramente superior debido al cálculo de probabilidades. En la última columna se muestra el promedio de la cantidad de firmas que verifica el sistema propuesto. Debido al enfoque probabilístico propuesto, a medida que el número de firmas aumenta, el tiempo consumido por el sistema presentado disminuye respecto al esquema básico, lo que hace que el sistema planteado sea más eficiente. De acuerdo con estos resultados, el sistema propuesto mantiene el tiempo de verificación en unos 0.138 min $\approx$ 8,3s, independientemente del número de firmas contenidas en el paquete. Por lo tanto, se puede concluir que el uso de un enfoque probabilístico mejora en gran medida el tiempo empleado en la verificación de firmas, y por lo tanto, demuestra la escalabilidad de la propuesta.

Tabla 3.2: Simulación para Diferentes Funciones Hash

Función Hash	**Tiempo Esquema Básico**	**Tiempo Esquema Agregación**	**N. de Firmas Verificadas**
MD5	0,14121162	0,148480218	9
SHA-1	0,093453982	0,096037364	7
SHA-256	0,058070086	0,064743644	4
MD5	0,379342788	0,133570748	10
SHA-1	0,289680197	0,111626543	8
SHA-256	0,182566924	0,171111552	10
MD5	0,81221526	0,144320619	9
SHA-1	0,728451986	0,146179821	9
SHA-256	0,452038378	0,132944675	9
MD5	1,246871085	0,135551543	11
SHA-1	1,007615639	0,143795704	10
SHA-256	0,679152238	0,135551543	11

Capítulo 4

VANETs en Teléfonos

El objetivo de las VANETs es hacer posible la detección de eventos en la carretera y el intercambio de mensajes sobre ellos entre los vehículos. Existen diversas iniciativas tanto desde la industria como desde el entorno académico destinadas a analizar este tipo de redes. Sin embargo, todas las propuestas existentes tienen en común la hipotética existencia previa de una infraestructura en la carretera y de dispositivos instalados en los vehículos, lo que implicaría un gran desembolso monetario tanto para el Estado como para los usuarios. Ambos temas se suponía que iban a estar resueltos en 2011, pero la crisis económica ha impedido este despliegue de las VANETs porque las instalaciones requeridas son demasiado costosas, tanto para el Estado como para los usuarios.

Con el fin de intentar implementar las VANETs sin costes adicionales, para los usuarios o para el gobierno, se propone en este capítulo un nuevo modelo de VANET totalmente auto-organizado que no supone la existencia de RSU y en el que el rol de las OBUs es reproducido parcialmente por una novedosa aplicación de software llamada VAiPho (acrónimo del inglés, VANET in Phones) [26], que hemos diseñado para que se ejecuten en los teléfonos móviles dentro de los vehículos. Teniendo en cuenta que muchas referencias bibliográficas sugieren que las OBUs utilizadas en los vehículos se conecten a través de Wi-Fi utilizando el protocolo 802.11, VAiPho implementa las VANETs en los teléfonos inteligentes, acogiéndose a sus características de dispositivos que proporcionan amplia conectividad Wi-Fi, GPS, y que cuentan con capacidad de cómputo y almacenamiento suficiente. De esta

manera se logra hacer un despliegue real de una VANET en la actualidad sin necesidad de inversiones. En particular, VAiPho está pensado para ser un sistema gratuito de comunicaciones seguras para redes vehiculares espontáneas y autogestionadas que utilizan los teléfonos inteligentes como OBUS en los vehículos y que no necesitan ninguna infraestructura en los vehículos ni las carreteras. El modo de funcionamiento propuesto es totalmente distribuido y descentralizado, y tiene en cuenta la protección de la privacidad e integridad ante posibles ataques.

En particular VAiPho contiene varios de los algoritmos y soluciones descritos anteriormente en esta tesis, además de utilidades prácticas como la detección de atascos y plazas de aparcamiento libres, entre otras. La implementación se ha hecho en dispositivos Windows Mobile 6.0 haciendo uso del lenguaje de programación *C#*. Este capítulo contiene algunos de los principales detalles de implementación. Las ideas principales expuestas en este capítulo has sido objeto de la publicación [25], así como de la patente [26] cuya licencia de explotación ha sido adquirida por la empresa DETECTOR [48].

4.1. Planteamiento del Problema

La finalidad principal de este capítulo es crear un modelo sencillo y escalable para VANETs donde los usuarios son capaces de colaborar a través de sus teléfonos móviles y adquirir información actualizada de interés sobre el estado del tráfico en las zonas próxima a él y de este modo poder elegir la ruta óptima para alcanzar su destino, mediante la información proporcionada en tiempo real. El número de congestiones aumenta cada año debido al rápido crecimiento del número de vehículos en la carretera, elevándose en 2011 a casi 1000 millones de vehículos a nivel mundial. En 2007 las pérdidas ocasionadas por las congestiones, tal como evaluaron los autores en [133], ascendían a 78 mil millones de dólares, donde 4.2 billones correspondían a número de horas perdidas y 2,9 billones a litros de gasolina consumidos sólo en Estados Unidos.

Además las congestiones de tráfico son una de las principales causas de contaminación del aire y en muchas ocasiones, del estrés que sufre la humanidad. Se ha probado que las personas atascadas en una congestión de tráfico tienen tres veces más probabilidades de

sufrir un paro cardíaco que aquellas que no están atrapadas en ninguna congestión, aunque no esté claro si los ataques al corazón se deben al estrés producido por el tráfico o están más relacionados con los niveles de contaminación a los que están sometidos los conductores en estas circunstancias. El intercambio de información entre vehículos podría mejorar esta situación, llegando a disminuir el número de congestiones y por tanto el enorme consumo de tiempo y dinero de los usuarios así como el consumo de gasolina con las consecuencias económicas y medioambientales correspondientes.

Existen numerosas aplicaciones GPS que utilizan un servicio centralizado para proporcionar información sobre el estado del tráfico. Estas aplicaciones obtienen la información de los servicios proporcionados por las autoridades de tráfico locales, departamentos de policía o/y sistemas de vigilancia del flujo del tráfico. Sin embargo, la información proporcionada por estos servicios en muchas ocasiones no proporcionan información en tiempo real. Además generan falta de privacidad sobre el usuario, el cual podría rehusar utilizarlos por miedo a que dicha información se aprovecha para otros objetivos.

Mucha de la bibliografía existente propone servicios similares a los que implementa VAiPho, pero ninguna de las propuestas utiliza el teléfono móvil como herramienta. En su lugar se basan en la existencia de RSUs y OBUs para alcanzar el objetivo. Otras aplicaciones orientadas a teléfonos móviles para la obtención de información sobre el estado del tráfico son: Google Traffic [67], TomTom [132], Sygic [131] o Waze [140], las cuales proporcionan información previo pago. Las principales diferencias con VAiPho son que todas ellas necesitan conexión 3G o GPRS para desempeñar su funcionamiento. Además, el cliente pierde completamente su privacidad dado que debe proporcionar información personal a las empresas que dan soporte al dicho servicio. Como resultado, esta información podría ser utilizada para proporcionársela a la Dirección General de Tráfico, como ya hizo TomTom, lo que nos lleva a cuestionarnos el alcance que podría tener esta información, llegando incluso a utilizarse como medio para poner una sanción.

VAiPho es el fruto de la investigación presentada en esta tesis. Consiste en un sistema de comunicaciones seguras, totalmente distribuido y descentralizado. La red generada surge de manera espontánea y se auto-organiza haciendo uso solo de teléfonos móviles inteligentes, los cuales incorporan GPS y comunicación Wi-Fi inalámbrica. No necesita de

ningún tipo de infraestructura instalada en vehículos ni en carreteras, y tiene en cuenta la protección de privacidad e integridad frente a diferentes ataques. Su principal propósito es incrementar la seguridad y el confort de los pasajeros mediante el intercambio de información sobre el estado de la carretera entre vehículos. Además permite disminuir las emisiones de CO_2, mejorar la eficiencia en la conducción aminorando el consumo de tiempo y combustible en las congestiones de tráfico, y aumentar el confort del usuario al reducir el tiempo necesario para alcanzar su destino y el número de desaceleraciones bruscas. La estructura de VAiPho permite aprovechar otros servicios de interés para el usuario como la detección de plazas de estacionamientos libres, así como encontrar el lugar donde está el coche aparcado.

En este capítulo presentamos el primer prototipo de VAiPho que permite el despliegue de una VANET haciendo uso de un pequeño número de dispositivos móviles, que cooperan en la retransmisión de la información. Aunque esperamos que la estructura de grupos reactivos permita una buena escalabilidad del sistema, a medida que se amplíe el número de usuarios que hagan uso de VAiPho se debe realizar un seguimiento para asegurar que las comunicaciones se llevan a cabo de manera satisfactoria independientemente del número de usuarios.

4.2. Estado del Arte

El desarrollo de un mecanismo para crear una red vehicular auto-organizada con el objetivo de incrementar la seguridad vial, requiere proporcionar sistemas que permitan garantizar la precisión y fiabilidad de la información retransmitida. Por lo tanto, la seguridad, a la hora de diseñar un sistema de comunicaciones para VANETs, es uno de los temas más importantes a considerar [120]. Como se ha visto en capítulos anteriores, las redes vehiculares son susceptibles a diversos ataques, por lo que en la bibliografía se puede encontrar diferentes propuestas de esquemas que permiten proteger VANETs auto-organizadas [1], [40], [116], [143]. Los autores de estos trabajos tratan de buscar soluciones a todas o parte de las debilidades de seguridad que presentan este tipo de redes, siendo estas propuestas diferentes a las presentadas como solución en este capítulo. El trabajo [50] tiene un objetivo similar a nuestro trabajo, pero no considera la protección en las comunicaciones

siendo este uno de los principales y más débiles aspectos de seguridad. Otro requerimiento que se ha considerado en VAiPho es el anonimato de los usuarios. Para ello se propone el uso de pseudónimos variables que permiten evitar los ataques de seguimiento monitorizado. En [9] se presenta un esquema de pseudónimos, que permite proteger la información relacionada con la velocidad y coordenadas de los vehículos en el envío de beacons. A diferencia de este trabajo, nuestra propuesta no envía este tipo de información en los beacons.

En este trabajo se ha considerado un despliegue gradual de las VANETs donde no se considera la existencia de RSUs, y donde el número de usuarios inicialmente será bajo e irá aumentando a medida que aumente el número de usuarios. De hecho el aumento de usuarios que entren a formar parte de la red dependerá del grado de aceptación que tenga la aplicación, que vendrá influenciado por la utilidad facilidad de manejo de la aplicación. Por este motivo uno de los principales elementos de la aplicación será la interfaz de usuario.

En relación con la idea presentada que permite encontrar aparcamientos son pocas las propuestas existentes y ninguna de ellas ha sido implementada en la realidad. El artículo [94] propone instalar en la puerta del copiloto un dispositivo que permite encontrar plazas de aparcamientos libres pero al igual que otras muchas aplicaciones existentes requiere de tecnología 3G y GPR para su funcionamiento. Esto se debe a que la información está centralizada en un servidor, el cual está a cargo de recibir la información de los vehículos y proporcionárselos a otros vehículos que lo consulten para localizar aparcamientos libres. Otra propuesta para encontrar aparcamiento se presenta en [115] donde se propone una solución basada en una red segura la cual no requiere la intervención de ninguna infraestructuras. El mayor problema que presenta esta solución es su facilidad para ser atacada dado que los usuarios podrían modificar la información indicando la existencia de una plaza de aparcamiento ficticia o en caso contrario marcándola como ocupada cuando está libre para evitar que otros usuarios puedan acceder a ella. En [41] se presenta una solución para encontrar una plaza dentro de un área de aparcamiento. Esta propuesta nos proporciona información sobre el número de plazas libres, la plaza de aparcamiento más próxima a nuestra posición, además de proporcionar su localización exacta. En cuanto a soluciones relacionadas con localizar el vehículo una vez aparcado existen numerosas propuestas para dispositivos móviles debido a su facilidad de implementación, [6], [84],[54], [61], [89], [128].

4.3. Requisitos del Sistema

La implementación de la aplicación será multiplataforma pero en esta tesis en particular se ha implementado una versión VAiPho de prueba para la plataforma Windows Mobile. En los trabajos futuros se implementará para iOS y Android dado que son estas las que en breve ocuparán todo el mercado. Independientemente de la plataforma a desarrollar, los móviles en los que se implemente VAiPho deben presentar unos requerimientos mínimos que se listan a continuación:

- Bluetooth®: Se utiliza para conectar teléfono móvil con el vehículo, y será el que arrancará de manera automática la aplicación VAiPho sin requerir ninguna acción por parte del usuario.
- Wi-Fi® IEEE 802.11 b/g: Permite el intercambio de información de manera gratuita entre los dispositivos que implementan VAiPho.
- 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 realicen operaciones complejas usando bastante información.

VAiPho hace uso de las características mínimas y comunes que presenta hoy en día cualquier teléfono inteligente. El número de usuarios que actualmente manejan este tipo de dispositivos es elevado y se encuentra en crecimiento debido a que cada día su precio es más asequible. Esto supone una gran ventaja para el despliegue de VAiPho por que mientras

mayor sea el número de usuarios que lo utilicen mejor será su funcionamiento. Otra ventaja es que los usuarios de este tipo de dispositivos están acostumbrados al manejo de estos dispositivos los GPS que incorporan, por lo que la interfaz que presenta VAiPho les resultará fácil de utilizar, evitando el rechazo a esta clase de aplicaciones por el desconocimiento y el esfuerzo que requiere empezar a manejarlas.

Las comunicaciones inalámbricas en VAiPho se realizan mediante el estándar de comunicaciones inalámbricas *IEEE 802.11b/g*. Este tipo de comunicaciones emite en el rango 2.4GHz lo que podría ocasionar interferencias con otros dispositivos Bluetooth, Zigbee u otras redes WLAN que ocupan el mismo canal de emisión. Además existen otros problemas, como el retardo en la propagación de la señal, el poco alcance que presentan las antenas al ser pequeñas, el consumo de batería el hecho de que el estándar *IEEE 802.11b/g* no está diseñado para trabajar a altas velocidades como las que presentan los vehículos, por lo que existe la posibilidad de que las comunicaciones fallen. Sin embargo, tras realizar numerosas pruebas para comprobar las comunicaciones utilizando el estándar *IEEE 802.11b/g* con vehículos se ha demostrado que el uso de este tipo de conexiones es satisfactorio. Es cierto que mediante este estándar, los vehículos que circulan a gran velocidad en direcciones opuestas apenas tienen tiempo para intercambiar información sin embargo, los que circulan en el mismo sentido o en ciudad donde la velocidad es más baja sí son capaces de establecer comunicaciones.

Para solucionar todos estos problemas se diseñó el estándar de comunicaciones inalámbricas *IEEE 802.11p* [79], que es más apropiado para este tipo de comunicaciones. No obstante, actualmente no existe ningún dispositivo con capacidad de retransmisión en la frecuencia de envío del estándar *IEEE 802.11p*. Por el contrario, la mayoría de los teléfonos inteligentes que se fabrican hoy en día sí tienen capacidad de comunicación mediante el estándar *IEEE 802.11b/g*. Por tanto, queda más que justificada la implementación de VAiPho haciendo uso de esta tecnología. En el caso de que el estándar *IEEE 802.11p* llegase a ser una realidad en los dispositivos, VAiPho podría modificarse de manera sencilla para ajustarse al nuevo estándar de comunicaciones debido a su estructura modular, como se verá en las siguientes secciones.

4.4. Estructura de VAiPho

A continuación se presentan los diferentes módulos implementados en VAiPho, incluyendo el esquema de agregación propuesto en el capítulo anterior. La Figura 4.1 muestra esquemáticamente los principales módulos utilizados:

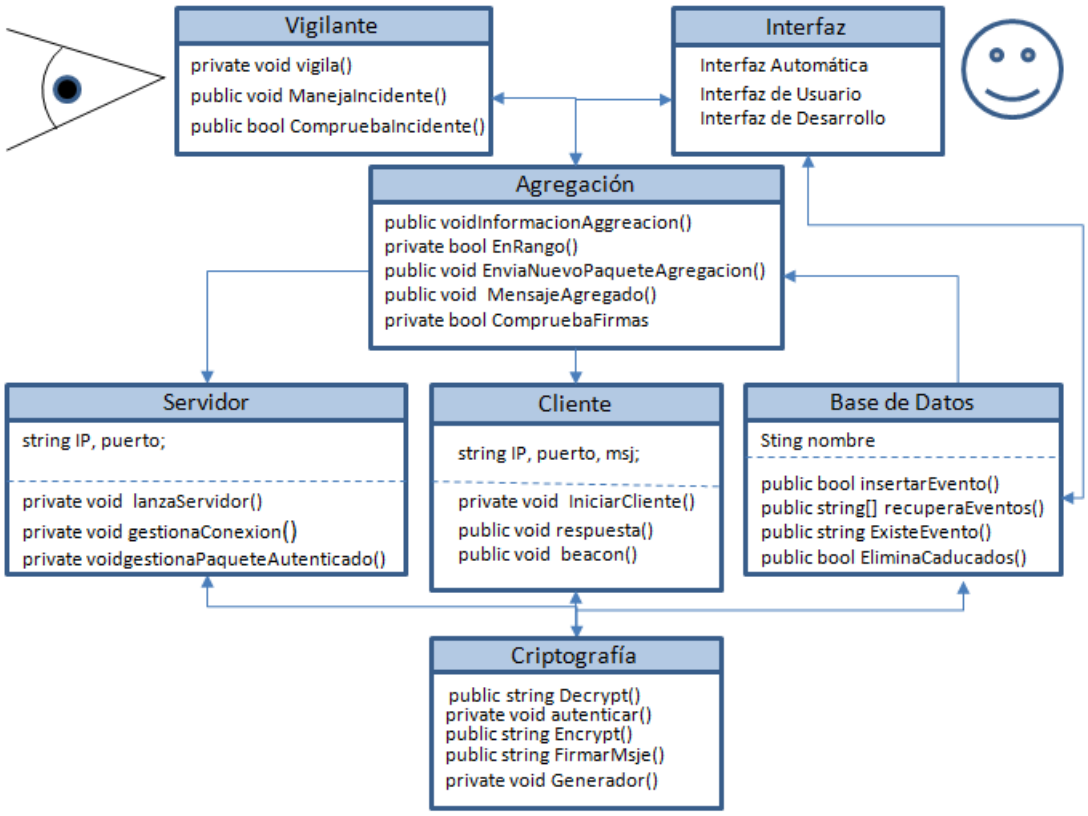


Figura 4.1: Módulos

- **Módulo Cliente**: Envía mensajes, ya sea en modo unicast a un nodo o en modo broadcast a toda la red con el fin de formar un grupo reactivo, intercambiar información sobre un evento, etc.

- **Módulo Servidor**; Es responsable de recibir y clasificar los paquetes en tres grupos: paquetes tipo R para ser firmados, si el nodo receptor está de acuerdo con el caso reportado, paquetes tipo S con las firmas para generar un paquete agregado y paquete

agregado tipo A. Dependiendo del tipo de paquete recibido, el módulo realiza la acción correspondiente.

- **Módulo Base de Datos**: Contiene todas las funciones encargadas de interactuar con la base de datos, como la inserción de eventos, así como su creación, consulta y eliminación. La base de datos almacena las advertencias de eventos producidos por el módulo Vigilante o incluidos en los paquetes de información recibida por el dispositivo.

- **Módulo Vigilante**: Es responsable de la detección de eventos en la carretera, como por ejemplo los atascos de tráfico. El módulo está continuamente consultando el GPS con el fin de obtener el tipo de vía por la que se circula, la velocidad mínima y máxima permitida y la velocidad actual del vehículo. Si esta velocidad es anormalmente baja de acuerdo con la velocidad recomendada para la vía, el dispositivo genera una posible alerta sobre la existencia de un atasco de tráfico. Después de un período de tiempo, si el vehículo continúa a una velocidad anormalmente baja, se genera un aviso de atasco de tráfico que pone en marcha el módulo de agregación y se almacena en la base de datos.

- **Módulo de Agregación**: Implementa la funcionalidad de agregación que se explicó en el capítulo anterior sobre una advertencia de evento producido por el módulo Vigilante. La primera tarea es comprobar si el evento indicado por el módulo Vigilante existe o no en la base de datos de manera que si no está en la base de datos, eso se interpreta como que se trata de un caso nuevo por lo que pone en marcha un procedimiento de agregación y se inicia un proceso de formación de grupos reactivos.

- **Módulo de Criptografía**: Es responsable de la firma, cifrado y descifrado de la información.

- **Módulo Interfaz**: Módulo encargado de las interfaces que se le presentan al usuario para que pueda interactuar con la aplicación. En concreto se han desarrollado 3 interfaces, dos para el usuario final y otra para las fases de desarrollo y prueba.

Para la implementación de cada uno de estos módulos se utiliza la programación concurrente de modo que la aplicación puede realizar varias tareas a la vez. En VAiPho

cada hilo tiene una tarea específica y determinada, lo que permite aumentar la eficiencia del uso del procesador. Si bien los hilos son generados a partir de la creación de un proceso, podemos decir que un proceso es un hilo de ejecución, conocido como Monohilo. Sin embargo las ventajas de los hilos se dan cuando hablamos de Multihilos, que es cuando un proceso tiene múltiples hilos de ejecución los cuales realizan actividades distintas, que pueden o no ser cooperativas entre sí. Los beneficios de los hilos se derivan de las implicaciones en el rendimiento:

1. Se tarda mucho menos tiempo en crear un hilo nuevo en un proceso existente que en crear un proceso, lo cual nos beneficia porque como se ha venido hablando durante la tesis, el tiempo juega un papel crucial en esta aplicación.

2. Se tarda mucho menos en terminar un hilo que un proceso. Teniendo en cuenta que se trabaja con dispositivos móviles donde los recursos son limitados, los hilos permiten liberar recurso de manera más sencilla.

3. Se tarda mucho menos tiempo en cambiar entre dos hilos de un mismo proceso. Además de puede parar o dormir un hilo cuando no es necesario, y volverlo a activar una vez se requiera su funcionamiento.

4. Los hilos aumentan la eficiencia de la comunicación entre programas en ejecución. En la mayoría de los sistemas, en la comunicación entre procesos debe intervenir el núcleo para ofrecer protección a los recursos y llevar a cabo la comunicación. En cambio, los hilos pueden comunicarse entre sí sin la invocación al núcleo. Por lo tanto, si hay una aplicación que debe implementarse como un conjunto de unidades de ejecución relacionadas, es más eficiente hacerlo con una colección de hilos que con una colección de procesos separados.

4.5. Módulos Cliente y Servidor

Las comunicaciones entre los dispositivos es de tipo cliente/servidor de manera que cada dispositivo actúa como un cliente y como un servidor. El módulo cliente se encarga de enviar paquetes a la red mientras que el módulo servidor está continuamente a la espera de

recibir paquetes para posteriormente procesarlos. El protocolo utilizado para el intercambio de paquetes es el UDP (acrónimo del inglés, User Datagram Protocol) debido a su rapidez ya que no realiza control de flujo, ni hay confirmación de entrega o recepción. Aunque este protocolo de transporte es muchísimo menos fiable que TCP (acrónimo del inglés, Transmission Control Protocol), se utiliza en esta aplicación porque es más importante la recepción rápida de los datos que la verificación de los mismos, debido a la alta movilidad que presentan los vehículos en este tipo de redes. En el caso de requerir verificación, se incorporan a nivel de aplicación procesos que llevan a cabo estas comprobaciones. UDP utiliza puertos para permitir la comunicación entre aplicaciones. El campo de puerto tiene una longitud de 16 bits, por lo que el rango de valores válidos va de 0 a 65.535. En VAiPho el puerto utilizado para las comunicaciones cliente/servidor es el 9050.

4.5.1. Creación de la Red VAiPho

Antes de poder llevar a cabo comunicaciones cliente/servidor es necesario establecer una red que permita la comunicación entre dispositivos que implementen VAiPho. Para permitir la comunicación entre dispositivos, es necesaria la existencia de una red a la cual se puedan conectar los dispositivos para hacer el intercambio de información. Al ser una red completamente distribuida y descentralizada que surge de manera espontánea, no existe ninguna infraestructura que la mantenga. Por lo tanto cada dispositivo tendrá la capacidad de crear una red en caso de que esta no exista, y en caso contrario simplemente se conectará a ella.

La VANET que se crea es única y tendrá un nombre común que conocerán todos los dispositivos de manera que les permita reconocerla y conectarse a ella. La red VAiPho no está protegida y por lo tanto todos los usuarios podrán conectarse a ella. Sin embargo, solo aquellos que hayan obtenido el par de claves pública y privada y el certificado correspondiente, podrán autenticarse e intercambiar información dentro de la red.

En este trabajo se utiliza una red Wi-Fi con nombre “vaipho” coincidiendo con el nombre de la aplicación. Para su implementación se utiliza OpenNETCF [43] que proporcionar extensiones útiles del núcleo de la librería .NET Compact Framework, permitiendo a los desarrolladores de dispositivos inteligentes concentrarse en la construcción de

las principales funcionalidades de la aplicación. La primera tarea es acceder a la interfaz inalámbrica del dispositivo. Una vez obtenida, se añade a la lista de redes predefinidas una red con el nombre que se le pasa por parámetro, en este caso "vaipho". Si todo funciona correctamente, una vez definida la red nos conectamos a ella mediante la llamada "wzc.ConnectToPreferredNetwork(wifi)". Si ya existe la red y había otros dispositivos en ella nos conectamos, en caso contrario nuestro dispositivo crea la red ad-hoc y se conecta a ella quedando a la espera de otros usuarios. En la Figura 4.2 puede verse el código.

```
public bool conectawifi(string wifi)
{
    INetworkInterface[] nIntarfaces = NetworkInterface.GetAllNetworkInterfaces();

    foreach (NetworkInterface ni in nIntarfaces)
    {
        if (ni is WirelessZeroConfigNetworkInterface)
        {
            WirelessZeroConfigNetworkInterface wzc = (WirelessZeroConfigNetworkInterface)ni;

            bool b;
            b = wzc.AddPreferredNetwork(wifi, false, "", 1, AuthenticationMode.Open, WEPStatus.WEPDisabled, null);
            if (b)
            {
                wzc.ConnectToPreferredNetwork(wifi);
                wzc.Bind();
                //MessageBox.Show("VAiPho Success.");
                return true;
            }
        }
        else
        {
            // MessageBox.Show(ni.Name + " Not WZC");
        }
    }
    return false;
}
```

Figura 4.2: Creación de la Red VAiPho

4.5.2. Envío de Paquetes

El módulo Cliente es el encargado de enviar paquetes a la red por lo que solo se ejecuta cuando existe un paquete a enviar. Los envíos pueden ser en modo broadcast a todos los nodos que forman parte de la red o bien, si se trata de un intercambio entre dispositivos se puede hacer a una dirección concreta en modo unicast. Tanto para el envío como para la recepción de paquetes se utilizan sockets. Los sockets son puntos o mecanismos de comunicación entre procesos que permiten que un proceso emita o reciba información

con otro proceso incluso estando estos procesos en distintas máquinas. Esta característica de interconectividad entre máquinas hace que el concepto de socket sea de gran utilidad en VAiPho. En concreto y teniendo en cuenta que se utiliza el protocolo UDP, el tipo de socket a utilizar es de datagrama. Este tipo de socket no requiere una conexión previa, sino que el servidor se sitúa en un puerto y espera la llegada de paquetes, mientras que el cliente se coloca en otro y envía paquetes al puerto del servidor. La clase cliente tiene tres campos que son: la dirección a la que se quiere enviar el paquete, el puerto a utilizar y el mensaje que se desea enviar. Cuando se vaya a enviar un paquete se invoca a la función "respuesta", que tiene como objetivo inicializar el campo IP que indica a qué dirección se quiere enviar el paquete, y a continuación se genera el mensaje que se va a mandar, el cual contiene el código que se corresponde con el tipo de mensaje, el pseudónimo del nodo que envía el mensaje y finalmente el contenido del mismo. El puerto no se inicializa en esta función porque al ser un parámetro fijo se inicializa cuando se crea una instancia de cliente. Una vez se tiene la dirección IP y el mensaje a enviar, se genera un nuevo hilo que será el encargado de enviar el paquete a la red mediante la función "IniciarCliente". De este modo queda liberado el hilo actual y puede seguir realizando otras acciones al mismo tiempo que otro hilo se encarga del envío del paquete. Una vez se invoca la función "IniciarCliente", transforma el mensaje al formato adecuado, en este caso bytes, y prepara el socket con la dirección IP y puerto al que se enviará el paquete. Finalmente, una vez preparado el socket se envía el paquete. En la Figura 4.3 puede verse el trozo de código con las dos funciones descritas anteriormente.

Al utilizarse el protocolo UDP para el intercambio de paquetes en la red no podemos estar seguros de la recepción de los mismos. En algunas ocasiones, como es el caso de la autenticación, es necesario asegurarnos de la recepción de los paquetes. Para llevar a cabo este proceso hemos implementado un control de envío/respuesta de manera que sabremos si el paquete ha alcanzado al nodo destino con el que nos estamos autenticando si nos responde con el correspondiente paquete de respuesta asociado al envío. Si transcurrido cierto período de tiempo no se obtiene la respuesta esperada, se envía el paquete nuevamente. Este proceso se repetirá hasta tres veces en caso de no recibir respuesta. Si tras los tres intentos no se obtiene la respuesta esperada, se desestima la conexión.

```
private void IniciarCliente()
{
    IPEndPoint serverEndPoint = new IPEndPoint(IPAddress.Parse(IP), int.Parse(puerto));
    UTF8Encoding encoding = new UTF8Encoding();//Pasamos la cadena de entrada a bytes
    byte[] plainTextBytes = encoding.GetBytes(msj);// + "/n");
    try{
        Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
        sock.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.Broadcast, 1);
        sock.SendTo(plainTextBytes, serverEndPoint);
        sock.Close();
    }catch
    {//error de socket
        Formulario.Invoke(Formulario.myDelegate, new Object[] { "Socket error" });
    }
}
public void respuesta(string codigo, string Pseu, string mensaje, string IPaddr)
{
   IP = IPaddr;//txtIpCliente.Text;
   msj = codigo + "," + Pseu + "," + mensaje;
   Thread hiloCliente = new Thread(new ThreadStart(IniciarCliente));
   hiloCliente.Start();
 }
```

Figura 4.3: Envío de Paquetes

4.5.3. Beacons

Cuando un dispositivo está conectado a la red, es necesario enviar beacons de manera que otros dispositivos sepan de la existencia de un nodo y puedan iniciar el proceso de autenticación. Para eso se utiliza el módulo Cliente, el cual se encargará de lanzar periódicamente mensajes que solo serán capaces de interpretar otros nodos que tengan implementado VAiPho. Los beacons se envían en modo broadcast a toda la red de manera que todos los dispositivos que forman parte de la red y se encuentren bajo la cobertura de la señal recibirán el paquete. En la implementación del envío de beacons se han considerado ciertas medidas de seguridad con el fin de evitar seguimientos monitorizados dentro de la red. Por una parte, los nodos tendrán asociado un pseudónimo de manera que en la red no se pueda determinar de qué nodo se trata. Es bastante fácil hacer un seguimiento de un nodo si este pseudónimo siempre permanece constante. Para evitar este problema simplemente se realiza un cambio de pseudónimo cada cierto tiempo. Sin embargo, el hecho de utilizar y cambiar de pseudónimo no impide hacer seguimientos por medio de los beacons, si la periodicidad con la que un nodo envía un beacon es constante ya que resultaría fácil asociar el cambio de

pseudónimo con un pseudónimo anterior. Para solucionar este problema, se implementa un envío de beacons y un cambio de pseudónimos de manera aleatoria. En particular, el envío de beacons se realiza de manera aleatoria en un periodo entre 5 y 15 segundos y el nodo cambia de pseudónimo en un período aleatorio entre 20 y 30 segundos. El envío de beacons es constante mientras la aplicación está corriendo, por lo que se implementa con un bucle que finaliza al cerrar la aplicación. En la Figura 4.4 puede verse la parte más interesante correspondiente a la implementación de la función "beacon".

```
public void beacon()
{
  ...
  while (!Formulario.cerrar)
  {// EL BEACON SE ENVIA PERIODICAMENTE(en un rango de tiempo)
    IP = IPAddress.Broadcast.ToString();| //BROADCAST
    puerto = "9050";
    msj = "01," + Server.myPseudonimo + "," + DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss") +", Ek1(ID1:KUid1)";
    hiloCliente = new Thread(new ThreadStart(IniciarCliente));
    hiloCliente.Start();
    Thread.Sleep(1000 * (5 + randomNumber.Next(10)));//beacon enviado en tiempo aleatorio entre 5 y 15
    vecesPseu++;
    if (vecesPseu == cambioPseu)//cuando se alcance esta cantidad se cambia el pseudonimo
    {
        newPseu = "pseu" + randomNumber.Next(99999);
        Server.viejoPseu = Server.myPseudonimo;
        msj = "01," + Server.myPseudonimo + "," + DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss") +",00,"+ newPseu);
        hiloCliente = new Thread(new ThreadStart(IniciarCliente));
        hiloCliente.Start();
        DButiles.cambiamyPseu(Server.myPseudonimo, newPseu);
        Server.myPseudonimo = DButiles.recuperamyPseu();
        Formulario.Invoke(Formulario.myDelegate5, new Object[] { newPseu });
        vecesPseu = 0;
        cambioPseu = 50 + randomNumber.Next(100);
        Thread.Sleep(1000 * (20 + randomNumber.Next(10)));
    }
  }
  ...
}
```

Figura 4.4: Beacons

4.5.4. Recepción de Paquetes

El módulo Servidor es el encargado de recibir paquetes de la red, y realizar una u otra acción en función de la información recibida. El modo de actuar dependerá del tipo de paquetes recibido. Cuando se inicia VAiPho, se lanza un hilo que será el encargado de preparar el módulo Servidor de manera que se quede a la escucha de posibles paquetes que lleguen de la red. Esto, al igual que en el módulo Cliente, se implementa haciendo

uso de los sockets. La función que lleva a cabo este proceso se llama "lanzaServidor", la cual inicializa el socket y se queda a la espera de que lleguen los paquetes. Una vez llegue un paquete, el hilo actual lanzará un nuevo hilo que será el responsable de la siguiente entrega de paquetes mientras que el actual se encarga de procesar toda la información y es responsable de ejecutar las nuevas operaciones en función de la información recibida. Con esto se permite a VAiPho seguir recibiendo información de la red mientras realiza otras acciones. El código correspondiente se muestra en la Figura 4.5.

```
private void lanzaServidor()
{
    ...
    if (!Formulario.cerrar)
    {
        hostName = Dns.GetHostName();
        thisHost = Dns.GetHostEntry(hostName);
        thisIpAddr = thisHost.AddressList[0].ToString();//recogo mi IP
        try
        {
            Formulario.Invoke(Formulario.myDelegate4, new Object[] { thisIpAddr });

        catch { }
        Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
        IPEndPoint iep = new IPEndPoint(IPAddress.Any, 9050);
        sock.Bind(iep);
        EndPoint ep = (EndPoint)iep;
        Console.WriteLine("Ready to receive...");
        byte[] data = new byte[1024];
        int recv = sock.ReceiveFrom(data, ref ep);//se queda esperando por el paquete entrante
        string stringData = Encoding.ASCII.GetString(data, 0, recv);

        IPEndPoint AddressIP = ep as IPEndPoint;//recuperamos ip del paquete entrante
        string clientIPAddress = "";
        if (AddressIP != null)
            clientIPAddress = AddressIP.Address.ToString();
        sock.Close();

        Thread HiloServidor = new Thread(new ThreadStart(lanzaServidor));//lanzamos hilo
        HiloServidor.IsBackground = true;
        HiloServidor.Start();
        string[] datosconexion=stringData.Split(',');
        gestionaConexion(stringData, clientIPAddress);//tratamos el paquete entrante
        ...
    }
}
```

Figura 4.5: Recepción de Paquetes

En la implementación de VAiPho el módulo Servidor puede recibir una gran variedad de paquetes diferentes. Los distintos tipos de paquetes se diferencian por medio de los campos que contiene el mensaje recibido y a partir de los cuales el servidor determinará la acción a realizar mediante la llamada a la función *gestionaConexion*.

4.6. Módulo Base de Datos

En cualquier aplicación normalmente son muchos los datos que son necesarios almacenar en una base de datos con el fin de poder acceder a ellos cada vez que se requieran. VAiPho también cuenta con una base de datos que es creada cuando se instala la aplicación. A medida que se ejecuta la aplicación se insertan, borran y modifican los contenidos de dicha base de datos. En este módulo se implementan todas las funciones encargadas de crear las tablas a utilizar así como las funciones encargadas de acceder a los datos para consultarlos o modificarlos. Son múltiples y variados los datos que se almacenan en esta aplicación. Entre ellos destacamos el almacén de claves, el almacén de certificados, información acerca del usuario, así como los eventos detectados e intercambiados. En relación a todo lo que se ha explicado en esta tesis nos centraremos en esta sección únicamente en las tablas relacionadas con los eventos detectados *Eventos* y *PosibleEventos*, en las que sus campos son prácticamente iguales, pero su coexistencia se debe a requerimientos de la implementación del módulo de Agregación que veremos más adelante. En la Tabla 4.1 se muestran los campos de la tabla *Eventos* que se han definido en esta tesis.

Tabla 4.1: Eventos

Campo	**Tipo**
ID	INT
TipoEvento	NTEXT
NombreVia	NTEXT
SentidoMarcha	NTEXT
CoordX	NTEXT
CoordY	NTEXT
CoordZ	NTEXT
TimeStamp	DATETIME
Expire	DATETIME
Signatures	NTEXT

Como puede verse en la Tabla 4.1 los campos almacenados se corresponden con los campos que se han definido en capítulos anteriores para la detección de eventos, además de almacenar información sobre las firmas como evidencias de la existencia de un evento recibido. Otros campos interesantes en esta tabla son el Timestamp y el Expire que permiten

optimizar el espacio de almacenamiento borrando aquellos eventos caducados de la base de datos. Las funciones encargadas de modificar esta tabla son "insertarEvento" que, como bien indica su nombre, se encarga de almacenar eventos recibidos o detectados en la base de datos. La función "recuperaEventos" permite recuperar todos los eventos vigentes almacenados en la base de datos. Esta función se utiliza para hacer el intercambio de todos los eventos con otro dispositivo cuando se establece una conexión. La función "ExisteEvento" permite determinar si un evento que se ha recibido ya existía en la base de datos. Si es así, es un evento ya detectado por lo que no se trata, ahorrando procesamiento, y tampoco se vuelve a almacenar, optimizando el espacio de memoria. Esta función junto con "EliminaCaducados" permite mantener actualizada la base de datos, de manera que cuando los eventos han alcanzado la fecha de expiración son eliminados. Esta última función, que se muestra en la Figura 4.6, es ejecutada por un proceso que se lleva a cabo cada cierto tiempo y, como se presentaba en el capítulo de cooperación, se corresponde con uno de los parámetros de cooperación para la optimización en la utilización del espacio de almacenamiento que usa VAiPho en la memoria del dispositivo.

Como puede verse en el código, lo primero que hace es conectarse a la base de datos que en la implementación se llama *VAIPHO_DB*. Una vez se crea la conexión, se hace la consulta *SELECT* para recuperar todos los eventos existentes en la base de datos. Cuando se han recuperado todos los registros almacenados, se recorren uno a uno comparando el campo expire con la fecha y hora actual, y si el evento ya ha caducado se procede a su eliminación. Una vez finalizado el proceso de actualización de la base de datos se cierra la conexión y termina la ejecución.

```
/*Elimina de la tabla eventos todos los eventos caducados*/
public bool EliminaCaducados()//string tipoEvento, DateTime FechaActual)
{
    try
    {
        SqlCeConnection conn = new SqlCeConnection("Data Source = VAIPHO_DB.sdf");
        conn.Open();

        //1.Create an instance of the command class, by using the SqlCeCommand object:
        SqlCeCommand cmd = conn.CreateCommand();

        cmd.CommandText = "SELECT * FROM Eventos";
        SqlCeDataReader myReader = cmd.ExecuteReader();
        while (myReader.Read())
        {
            DateTime expire = myReader.GetDateTime(8);//miramos campo expire
            //Se comprueba que el evento sea actual

            if (DateTime.Now.CompareTo(expire) > 0)

            {
              ...
              //borramos de la bd
              cmd.CommandText = "DELETE FROM Eventos WHERE id = " + myReader.GetInt32(0).ToString();
              cmd.ExecuteNonQuery();
            }

        }
        if (conn.State == ConnectionState.Open)
            conn.Close();
        return true;

    }
    catch (Exception e)
    {
      Formulario.Log("DB=>Error en existe evento: " + e.Message);
      return false;

    }
}
```

Figura 4.6: Función del Módulo Base de Datos “EliminaCaducados”

4.7. Módulo Vigilante

De todos los módulos este es uno de los más complejos y delicados a la hora de su implementación dado que de él depende el buen funcionamiento de la aplicación. Este módulo es el encargado de detectar qué está ocurriendo en la carretera. Su papel es el de ser los ojos de VAiPho. Actualmente la aplicación es capaz de generar y detectar 3 tipos de eventos diferentes:

1. Aparcamientos Libres. Este evento no requiere del módulo Vigilante, ya que simplemente cuando un nodo se enciende, se sincroniza con el vehículo y establece la posición

GPS en la que se encuentra. Una vez obtenida la posición genera un evento indicando que se ha dejado un aparcamiento libre. Si al encender el vehículo, el dispositivo no tiene cobertura GPS, por ejemplo si se encuentra en un domicilio particular, el dispositivo no genera ningún evento, lo que es correcto porque ningún otro vehículo podría ocupar ese lugar.

2. Publicidad. Este tipo de evento proporciona publicidad geolocalizada sobre algún establecimiento. Este evento tampoco es generado por el módulo Vigilante, sino por el propio establecimiento al acercarnos a él. Actualmente está implementado de manera que cuando nos acerquemos al establecimiento, se envíe el paquete y en el mapa se dibuje el logotipo del establecimiento que ha enviado la publicidad.

3. Detección de Atascos. Este evento sí es generado por el módulo Vigilante, el cual por medio de la velocidad del vehículo detecta la posible existencia de un atasco. Transcurrido cierto tiempo confirmará la existencia del evento y con la ayuda de otros vehículos haciendo uso de la agregación, generará un evento. Este funcionamiento se detalla a continuación.

El módulo Vigilante se implementa por medio de un hilo que arranca al iniciar la aplicación. Este hilo estará ejecutándose continuamente mientras la aplicación esté corriendo. El objetivo es determinar la velocidad a la que circula el vehículo y compararla con la velocidad máxima permitida en la vía. Como se presentó en el módulo de agregación, para determinar la existencia de un atasco por medio de la velocidad se propone la utilización de lógica difusa como mejora a la otra propuesta presentada en esta tesis, que es la que actualmente está implementada.

La función "vigila" es la encargada de detectar atascos en la carretera en tiempo real. Esta función hace uso de la API (acrónimo del inglés, Application Programming Interface) de Sigyc [131], que proporciona funciones de gran utilidad además de permitir mostrar en el mapa los puntos donde se localizan los eventos. La función "vigila", consulta continuamente a qué velocidad circula el vehículo y cuál es la velocidad máxima de la vía. Tal como está actualmente implementado, si en un momento dado la velocidad de circulación es inferior a la cuarta parte de la velocidad máxima de la vía se genera una alarma

indicando que posiblemente se encuentre en una situación de atasco. Como puede observarse si el sistema detecta una velocidad anormalmente inferior no genera el atasco inmediatamente, ya que durante la conducción es posible que el vehículo se detenga por una señal de stop, por un semáforo, etc., lo cual se debe diferenciar de una situación de atasco. Una vez el sistema genere la alarma de posible atasco, comprobará durante un cierto período de tiempo, aproximadamente 1 minuto, si el dispositivo continúa en la misma circunstancia. Si transcurrido este período el sistema continúa en situación de atasco, generará una señal de atasco e iniciará un proceso de generación de evento. El primer paso consistirá en recoger toda la información correspondiente al atasco como son: longitud, latitud y altitud, nombre de la vía y sentido de circulación en el que se detectó el atasco. La parte más interesante de esta función puede verse en el código de la Figura 4.7.

```
private void vigila()
{
  ...
    while (!Formulario.cerrar){
        //Se obtiene información de posición gps y velocidad de circulación
        if (CApplicationAPI.GetActualGpsPosition(out err, out gps, false, 1000) == 1){
            LONGPOSITION position = new LONGPOSITION(gps.Longitude, gps.Latitude);
            CApplicationAPI.MetersToWorldDegree(ref position);
            try{
                //Se obtiene la velocidad límite de la vía
                if (CApplicationAPI.GetCurrentSpeedLimit(out err, out speedLimit, 1000) == 1){
                    //Si estamos cin una vía y la velocidad de circulación es considereblemente inferior
                    //a la velocidad normal de la via marcamos un posible atasco
                    if ((gps.Speed >= 0) && (gps.Speed <= (speedLimit / 4))){
                        if (atascado == 0){text4 = "Possible Traffic Jam";}
                        atascado++;
                        Thread.Sleep(1000*15);
                        if (atascado == TiempoAtascado){
                            //Se recoge el nombre de la vía donde hay atasco y el sentido de la marcha
                            if (CApplicationAPI.GetLocationInfo(out err, position, out direccionTJam, 0) == 1)
                            {
                                longitude = position.lX;// / 100000.0;
                                latitude = position.lY;// / 100000.0;
                                altitud = gps.Altitude;
                                sentido = "";
                                //Se obtiene el sentido de la marcha
                                sentido = ObtenerSentidoMarcha(gps.Course);
                             }
                        //Se llama a la función encargada de manejar los incidentes y el hilo actual sigue vigilando.
                        ManejaIncidente(util.cortaDireccion(direccionTJam),sentido,longitude, latitude, altitud);
                        atascado = 0;
  ...
}
```

Figura 4.7: Función del Módulo Vigila “vigila”

Una vez se ha detectado el atasco y se ha recogido toda la información relacionada con él se llama a la función “ManejaIncidente” encargada de la generación y envío del evento. Esta función primero comprobará si ya el evento se encuentra registrado como detectado en

la base de datos, en cuyo caso no hará nada porque o bien está en proceso, o bien ya se ha enviado la información. En caso de no existir el evento como confirmado se comprueba en la base de datos si existe como posible evento. Si existe como posible evento es porque ya se ha detectado con anterioridad y se ha generado el paquete pero nadie lo ha firmado, por lo que se vuelve a enviar pues hasta que otros nodos no corroboren la información, el evento no se toma como válido. Si tampoco se encuentra almacenado como posible evento significa que se trata de un nuevo evento por lo que se almacena en la base de datos y se prepara para enviarlo. En este caso el nodo actual se considera líder del grupo reactivo y se queda a la espera de recepción de firmas, siguiendo el esquema de agregación de datos propuesto. En el código de la Figura 4.8 se puede ver el proceso hasta que el nodo determina que es líder y se queda a la espera de la recepción de firmas y envía el paquete.

```
public void ManejaIncidente(string NombreVia, string sentidoMarcha, double X, double Y, double Z)
{
 ...
 //Se comprueba si ya existe el incidente en las coordenadas X,Y aprox.
 evento = AccesoBD.ExisteEvento("T1", NombreVia, sentidoMarcha, X, Y);
 //Formulario.Invoke(Formulario.myDelegate, new Object[] { "¿¿Existe Evento??" });
 if (!evento.Equals("error"))
 {
     string[] words = evento.Split('|');
     //Si no existe el evento se introduce en BD
     if (evento.Equals(""))
     {
         // Formulario.Invoke(Formulario.myDelegate, new Object[] { "NO" });
         evento2 = AccesoBD.ExistePosibleEvento("T1", NombreVia, sentidoMarcha, X, Y);
         if (evento2.Equals(""))
         {
             if (AccesoBD.insertarPosibleEvento("T1", NombreVia, sentidoMarcha, X, Y, Z, FechaCreacion);
             {
                 Formulario.Invoke(Formulario.myDelegate, new Object[] { "Inserta en PosibleEvneto" });

                 //Si se trata de un nuevo evento se envía un mensaje buscando agregación.
                 AggregarFirmas.setFirmas();
                 AggregarFirmas.setNumerofirmas();
                 AggregarFirmas.setEsperoFirmas();
                 //AggregarFirmas.getEsperoFirmas();
                 Formulario.Invoke(Formulario.myDelegate, new Object[] { "Me quedo a la espera de firmas"
             }


       //Se envia el paquete para que otros vehículos lo firmen
       agregacion Agpaq = new agregacion(Formulario, NombreVia, sentidoMarcha, X, Y, Z, FechaCreacion);
       Agpaq.EnviaNuevoPaqueteAgregacion();
   ...
}
```

Figura 4.8: Función del Módulo Vigila, “ManejaIncidente”

Una vez se tiene toda la información del evento y el nodo se queda a la espera de firmas, se llama a la función "EnviaNuevoPaqueteAgregacion" encargada de enviar el paquete tipo R descrito en el capítulo 3, haciendo uso del módulo Cliente. En este punto comienza el módulo de Agregación tanto en el nodo receptor como en el nodo actual al recibir las firmas.

Otra funcionalidad que se presenta en este módulo es el envío de información al dejar un aparcamiento libre. Cuando se arranca un vehículo se puede suponer que es para liberar un aparcamiento. El evento se define en VAiPho como posible aparcamiento porque no se puede asegurar que el vehículo que libere un espacio esté realmente bien estacionado. El dispositivo al encenderse invocará a la función "posibleparking" la cual conecta con el GPS obteniendo información sobre su posición, tales como coordenadas (X,Y,Z) y nombre de la vía. A esta información se le añade la fecha y hora, y se envía el correspondiente paquete de información sobre el posible aparcamiento. Esta información tiene un tiempo de expiración muy corto, dado que no podemos asegurar cuánto tiempo permanecerá el aparcamiento libre. En la Figura 4.9 se muestra el código más importante referente a esta función.

```
//señala un posible aparcamiento
public void posibleparking()
{
  ...
   try
   {
       if (CApplicationAPI.GetActualGpsPosition(out err, out gps, false, 1000) == 1)
       {
           LONGPOSITION position = new LONGPOSITION(gps.Longitude, gps.Latitude);
           sentido = ObtenerSentidoMarcha(gps.Course);
           CApplicationAPI.GetLocationInfo(out err, position, out via, 0);
           via = util.cortaDireccion(via);
           int CoorX = gps.Longitude;
           int CoorY = gps.Latitude;
           int CoorZ = gps.Altitude;
           DateTime fechaDato = DateTime.Now;
           string datoAparcamiento = via + ";" + sentido + ";" + CoorX + ";" + CoorY + ";"
                   + CoorZ + ";" + fechaDato.ToString("dd/MM/yyyy HH:mm:ss");
           string firma = criputiles.RSAfirma(datoAparcamiento);
           auxiliar = criputiles.Encrypt(datoAparcamiento+";"+firma, DButiles.recuperamySK());
           mycliente.respuesta("P2", Server.myPseudonimo,auxiliar , thisIpAddr);
           DButiles.insertarEvento("P2", via, sentido, CoorX, CoorY, CoorZ, fechaDato,
                                   fechaDato.AddSeconds(Server.validezAparcamiento), firma);
           break;
       }
   }
   catch(Exception e)
   {
       e.ToString();
   }
  ...
 }
```

Figura 4.9: Función del Módulo Vigila “posibleparking”

4.8. Módulo de Agregación

Este módulo tiene como objetivo asegurar que la información que se genera en la red es de confianza. El módulo de Agregación, tal como vimos en el módulo Vigilante, comienza con el envío de un paquete tipo R para ser firmado por otros nodos. Cuando el módulo Servidor recibe un paquete, ejecuta la función “gestionaPaqueteAutenticado” que determinará si se trata de un paquete tipo R. En ese caso desglosará toda la información recibida en el paquete como: mensaje, nombre de la vía, sentido de la marcha y las coordenadas (X,Y,Z) para luego invocar a la función “InformacionAggregacion”. Esta función es la encargada de devolver el resultado de firmar un mensaje si se está de acuerdo con la información recibida. Esta información lo primero que comprueba es si se trata de un evento que ya existe en la base de datos, en cuyo caso no lleva a cabo ninguna acción porque ya se ha tratado con anterioridad. En caso contrario comprueba si nos encontramos dentro del rango del evento y por lo tanto somos capaces de detectarlo, para lo cual se utiliza la función “EnRango”, la cual, con el nombre de la vía, el sentido de la marcha y las coorde-

nadas GPS, comprueba si el punto donde está el incidente puede ser detectado por nuestro dispositivo. En caso negativo, no se realiza ninguna acción porque no somos capaces de detectar el evento. Si por el contrario está dentro del rango, se comprueba si es capaz de detectar el problema llamando a la función "CompruebaIncidente" del módulo Vigilante. Esta función haciendo uso de la API del GPS volverá a seguir un proceso parecido al de la función "vigila" para determinar si el dispositivo detecta un atasco o no. Si la función "CompruebaIncidente" no es capaz de detectar la congestión encontrándose en el rango de la señal, circulando por la misma vía y en el mismo sentido, deducirá que se trata de un intento de ataque. Actualmente el módulo de cooperación no se encuentra implementado, pero en el futuro será en este punto donde se introduzca a los nodos en la lista individual de reputación. En el siguiente trozo de código de la Figura 4.10 vemos el proceso descrito.

```
/*Función que devuelve el resultado de firmar un mensaje si estamos de acuerdo con la información*/
public void InformacionAggreacion(string msje, string IP)
{
   ...
   //Se comprueba si ya existe el incidente en las coordenadas X,Y aprox.
   evento = AccesoBD.ExisteEvento("T1", NombreVia, SentidoMarcha, CoodX, CoordY);
  // Formulario.Invoke(Formulario.myDelegate, new Object[] { "¿Tengo el evento en BD? " });
   if (evento.Equals(""))            {
       //SE VA A COMPROBAR SI SE PUEDE SER TESTIGO DE ESTE EVENTO
     //1º Obtenemos el la vía y el sentido de circulación de nuestr vehículo
     viaSentido = incidente.ViaSentidoMarcha();
     (variable local) string viaSentido
     string sentido = "";
     int X, Y;
     //Se comprueba si tenemos cobertura gps
     if (!viaSentido.Equals(""))
     {
         string[] viaSentidoAux = viaSentido.Split('|');
         via = utilidades.cortaDireccion(viaSentidoAux[0]);
         sentido = viaSentidoAux[1];
         X = int.Parse(viaSentidoAux[2]);
         Y = int.Parse(viaSentidoAux[3]);

         //2º Si la información está dentro de mi Rango, y circulo por esa vía yo puedo detectar el problema
         if (EnRango(X, Y, via, sentido))
         {
           //3º Si estoy dentro del rango compruebo si detecto el problema
           if (incidente.CompruebaIncidente())
           {
               ...
```

Figura 4.10: Función del Módulo Agregación "InformacionAgregacion"(parte 1)

Si por el contrario, el nodo es capaz de detectar el problema, el siguiente paso es determinar si el evento se encuentra registrado como posible evento. Si se encuentra como posible evento puede ser porque ya se había detectado anteriormente, en cuyo caso es posible que el nodo que haya recibido el paquete haya generado también un paquete de este tipo y

esté a la espera de firmas, por lo que se trata de dos nodos potencialmente líderes. Entonces el nodo actual hará una comparación entre el momento en el que se generó el evento recibido y el momento en el que él envió al paquete y, si el suyo es más antiguo continuará a la espera de firmas y descartará el paquete recibido. En caso contrario, borrará toda la información que lo asocie con el comportamiento de un líder y enviará un paquete tipo S firmado, indicando que está de acuerdo con la información recibida. En el código de la Figura 4.11 puede verse la implementación del proceso descrito.

```
...
//Si ya lo tengo el posibleenveto en BD puede que esté esperando firmas yo también
else
{
 if (AggregarFirmas.getEsperoFirmas())
 {
  string[] info = posibleevento.Split('|');
  if (DateTime.Parse(info[2]).CompareTo(HoraGeneracionPaquete) < 0)
  {
    el mio es más antiguo me quedo con el mio
  }
  //Sino, tengo que modificar el mio y quedarme con este
  else
  {
   //Ya no se queda a la espera de ninguna firma
   AggregarFirmas.resetEsperoFirmas();
   //Firmo el paquete
   hash = FirmarMsje(msje, Server.myPseudonimo);
   //Busco my pseudónimo
   string myPseudonimo = AccesoBD.recuperamyPseu();
   //Mando el mensaje  con  la Firma
   //Nota hay que agregar el hash
   Cliente Client = new Cliente(Formulario, "9050");
   Client.respuesta("T1", myPseudonimo, criputiles.Encrypt("S_Traffic Jam_" + NombreVia + "_" +
    SentidoMarcha + " " + Server.myPseudonimo + "_" + hash + "_" + CoodX + "_" + CoordY + "_" +
    CoordZ, AccesoBD.recuperamySK()), IP);
...
```

Figura 4.11: Función del Módulo Agregación "InformacionAgregacion" (parte 2)

Si por el contrario el evento recibido, que hemos comprobado que somos capaces de detectar, no se encuentra registrado como posible evento, entonces es almacenado como posible evento, firmado y enviado siguiendo un proceso similar al que se mostraba anteriormente. Cuando el paquete tipo S es recibido por el líder, entrará en el módulo Servidor y otra vez se invocará a la función "gestionaPaqueteAutenticado", la cual en este caso realizará el proceso indicado para este tipo de paquetes. En primer lugar, desglosa la infor-

mación recibida y comprueba que se trata de un evento almacenado en la tabla de posibles eventos. Si determina que tiene almacenada información relacionada con este evento, comprueba si está a la espera de firmas y por lo tanto si es el nodo indicado para recibir este evento. En caso afirmativo agregará la firma recibida a otras firmas que ya hubiese recibido con anterioridad, si existieran. El paso siguiente será comprobar si tiene firmas suficientes para poder generar el evento y enviarlo a la red. En el caso analizado en el capítulo de agregación de datos, el líder esperaba a tener todas las firmas de los nodos de su grupo. En este caso sin embargo, como solo se ha contado con cuatro dispositivos donde cada uno tenía su rol bien definido, el número de firmas requerido ha sido de dos, como puede verse en el código. Una vez recibidas todas las firmas necesarias, el nodo elimina el registro de posible evento y lo inserta en la tabla evento porque ha sido confirmado por un número suficiente de vehículos. A continuación genera un paquete tipo A que, además de la información sobre el evento, contiene todas las firmas y lo envía haciendo uso del módulo Cliente. En el código de la Figura 4.12 se muestra su implementación.

```
...
posiblevento = AccesoBD.ExistePosibleEvento("T1", nombreVia, sentidoMarcha, X, Y);
Formulario.Invoke(Formulario.myDelegate, new Object[] { "Does the possible Event exist?" });
if (!(posiblevento.Equals("") && posiblevento.Equals("error")))
{
  string firmas = AggregarFirmas.getfirmas();
  AggregarFirmas.agregafirmas(firmas + firma + "|" + IDVehículo);
  AggregarFirmas.incrementarfirmas();
  firmas = AggregarFirmas.getfirmas();//<--NUEVO para meter la firma con el IDVehículo

  //Ya tengo todas las firmas que necesito, mando el paquete agregado
  if (AggregarFirmas.getnumerofirmas() >= 2){
      string IP = IPAddress.Broadcast.ToString();
      if ( AccesoBD.insertarEvento("T1", nombreVia, sentidoMarcha, X, Y, Z, myDateTime,
        myDateTime.AddSeconds((validezAtasco), firma))
        {
          utiles.anadePoi("Atasco", "Atasco", (int)(X), (int)(Y), 100);
        }
        Cliente Client = new Cliente(Formulario, "9050");
        string prueba = "A_Traffic Jam." + AggregarFirmas.getfirmas() + "_" + nombreVia + "_" + sentidoMarcha + "_"
      + X.ToString() + "_" + Y.ToString() + "_" + Z.ToString() + "_" + myDateTime.ToString("dd/MM/yyyy HH:mm:ss");
        Client.respuesta("T1", myPseudonimo, Criptutiles.Encrypt(prueba, AccesoBD.recuperamySK()), IP);
        AggregarFirmas.resetEsperoFirmas();
    }
}
...
```

Figura 4.12: Función del Módulo Agregación “InformacionAgregacion”(parte 3)

La última función interesante que presenta este módulo es "MensajeAgregado" que se encarga de comprobar que las firmas recibidas en el paquete son correctas para insertar el mensaje en la base de datos, mandar la información al módulo Interfaz y proporcionar la información al conductor. En primer lugar, la función se encarga de comprobar si el evento recibido ya se encuentra en la tabla Eventos de la base de datos. Si es así no se realiza ninguna acción porque el evento ya fue tratado y el conductor ya ha recibido la información sobre el incidente por lo que se ahorra procesamiento. En caso contrario se llama a la función que implementa el código descrito en la sección 3.4.5 que comprueba las firmas contenidas en el paquete. Si el resultado de la comprobación es correcto, se obtiene el evento del paquete, se inserta en la base de datos y por medio de la función de interfaz "anadePoi" se alerta al conductor sobre una congestión en la ruta. En caso contrario se detectaría un intento de ataque y se incluiría al nodo generador del paquete en la LIR. Todo el proceso descrito se muestra en el código de la Figura 4.13.

```
/*Función que comprueba las firmas y si son correctas agrega el mensaje en BD*/
public void MensajeAgregado(string msje)
{
   ...
  evento = AccesoBD.ExisteEvento("T1", NombreVia, SentidoMarcha, CoodX, CoordY);
  Formulario.Invoke(Formulario.myDelegate, new Object[] { "Do I have the event in DB? " });
  if (evento.Equals(""))
  {
    //Compruebo las firmas contenidas en el mensaje
    if (CompruebaFirmas(msje))
    {
        Formulario.Invoke(Formulario.myDelegate, new Object[] { "Firmas OK" });
        string[] info = msje.Split('.');
        string mensaje = info[0];
        if (AccesoBD.insertarEvento("T1", NombreVia, SentidoMarcha, CoodX, CoordY, CoordZ,
          HoraGeneracionPaquete, HoraGeneracionPaquete.AddSeconds(Server.validezAtasco), info[1]))
        {
         utiles.anadePoi("Atasco", "Atasco", (int)(CoodX), (int)(CoordY), 100);
        }
    }
    //Informo de que podría tratarse de un ataque
    else
    {
        Formulario.Invoke(Formulario.myDelegate, new Object[] { "Intento de ataque en la red" });
    }
  ...
}
```

Figura 4.13: Función del Módulo Agregación "MensajeAgregado"

4.9. Módulo de Criptografía

Además de la protección de la identidad de los usuarios por medio de los beacons, como se presentaba en la sección dedicada al módulo Cliente, VAiPho incluye otras herramientas de seguridad implementadas en el módulo de Criptografía. En concreto este módulo presenta una función encargada de cifrar los datos con una clave pública "EncryptData" y otra función "DecryptData", encargada de descifrar los datos con una clave privada. Si se usa "DecryptData" con la clave privada del emisor la información no puede ser revocada y solo puede ser leída por parte de otros nodos haciendo uso de la clave pública del emisor. La función "DecryptData" por su parte puede usarse para descifrar los datos que hayan sido cifrados con la clave pública del nodo receptor, lo que asegura que solo ese nodo es capaz de descifrar esta información. Estas dos funciones se utilizan en el proceso de autenticación de nodos mediante la función "autenticar", que al no ser objeto de esta tesis, no será definida aquí. Otra función importante que ya presentábamos en el módulo de Agregación es "FirmarMsje", que se encarga de hacer primero un hash del mensaje y a continuación aplicar la función "DecryptData" con la clave privada del emisor.

El módulo de criptografía incluye también la implementación del generador presentado en el capítulo de cooperación. La función que lo implementa se llama "Generador" y permite cifrar las comunicaciones de salida de nuestra aplicación, por lo que será llamada desde el módulo Cliente. La función "Generador" permite fijar el tamaño de los elementos del generador, así como el polinomio primitivo a utilizar. Los parámetros por defecto son $L = 20$ y $polinomio = x^{20} + x^3 + 1$. A continuación la función calcula el número primo menor y más cercano a L/2 y define la función tal como se presentó en la sección 2.8.2. Luego comienza a funcionar el registro inicializándolo con la semilla. Cada semilla se usa para generar tantos bits como haga falta para cifrar. A continuación podemos ver en el código de la Figura 4.14 los pasos más interesantes de la implementación de la función "Generador".

```
static void Generador(string[] args)
{
    ...
    L = 20;
    Polinomio = new int[] { 20, 3 };
    //Se busca el primo inmediatamente inferior a L/2
    primo = BuscaPrimoInferio((int)L / 2);
    //Generamos el registro y todas las claves o semillas posibles con L bits
    Registro = new int[L];
    clave = Math.Pow(2, L) - 1;
    for (i = 1; i <= (int)clave; i++){
        //Se inicializa el vertor a 0
        for (int t = 0; t < Registro.Length; t++){
            Registro[t] = 0;
        }
        //Pasamos las claves a números binarios
        Registro = PasarBinario(i, (int)L);
        // Para cada clave ejecutamos el generador ejecutagenerador veces
        ejecutagenerador = Math.Pow(2, L);
        for (j = 0; j <= (int)ejecutagenerador; j++){
            //Se calcula el bit de Realimentación
            bitRealim = Registro[Registro.Length - Polinomio[0]];
            for (k = 1; k < Polinomio.Length; k++){
                bitRealim = bitRealim ^ Registro[Registro.Length - Polinomio[k]];
            }
            //Se recoge el bit de salida Sj
            Sj = Registro[0];
            //Se desplaza el registros un bit a la izquierda
            Registro = DesplazaRegistro(Registro);
            //Se introduce el bit de realimentación
            Registro[Registro.Length - 1] = bitRealim;
            //Se calcula la función f y la introducimos en suma
            Zj = Registro[primo];
            for (l = 2; l <= primo; l++){
                n = L / l;
                m = 0;
                ind = 0;
                while (m < n){
                    producto = 1;
                    for (d = 0; d < l; d++){
                        producto = Registro[ind] * producto;
                        ind++;
                    }
                    Zj = Zj ^ producto;
                    for (d = m; d < m + l; d++){
                        producto = Registro[d] * producto;
                    }//end for
                    Zj = Zj ^ producto;
                     m++;
                }//end while
            }//end for
            if (Sj != 0){
                EscribeSalida(Zj);
                }
    ...
}
```

Figura 4.14: Función del Módulo de Criptografía “Generador”

4.10. Módulo Interfaz

Como todas las aplicaciones, VAiPho presenta varias interfaces. En concreto distinguimos 3 interfaces diferentes. Las más importantes son la Interfaz Automática y la Interfaz de Usuario. Se ha desarrollado una tercera interfaz llamada Interfaz de Desarrollo, para poder hacer pruebas y comprobar errores durante el desarrollo de la aplicación pero dicha interfaz no será accesible para los usuarios. Esta interfaz puede verse en la sección 4.11.

4.10.1. Interfaz Automática

Se puede considerar a esta interfaz como la principal. Es la interfaz encargada de notificar al usuario diferentes eventos que suceden en la carretera. Su aspecto gráfico, como vemos en la Figura 4.15, es el mapa de un GPS donde se muestran las condiciones de tráfico, la publicidad y, haciendo uso del botón que se encuentra en la parte superior, las posibles plazas de aparcamiento disponibles en la zona. Para evitar posibles distracciones del conductor, al detectar un evento además de mostrar la información recibida por pantalla se emite por medio de mensajes voz.

Figura 4.15: Interfaz Automática

4.10.2. Interfaz de Usuario

Esta interfaz es la que permite la configuración de todos los parámetros variables de VAiPho, como vemos en la Figura 4.16. Uno de esos parámetros variables es la posibilidad de que VAiPho se apague cuando la batería del móvil llegue a un determinado umbral.

En términos de consumo de batería la idea fundamental es que la principal utilidad de un teléfono móvil es hacer llamadas. Si el usuario detecta que la aplicación VAiPho consume tanta batería que no le permite hacer un uso adecuado de su móvil optará por detenerla. Como se ha expuesto en el capítulo de Cooperación esto tiene un efecto negativo en el funcionamiento de la red. En la configuración, el botón "conf" de la aplicación se permite fijar umbrales de batería por debajo de los cuales la aplicación se detendrá automáticamente. Con esta medida se asegura que el usuario no apague VAiPho por miedo a quedarse sin batería sino que coopere en la retransmisión de eventos porque sabe que VAiPho no va a agotar la batería del dispositivo. Una ventaja que presenta VAiPho con respecto al consumo de batería en comparación con otras aplicaciones existentes en el ámbito de servicio de información sobre el tráfico en tiempo real es que no utiliza conexiones 3G, que es el tipo de conexión que mayor consumo de batería genera. Con el fin de determinar el gasto aproximado de batería que supone VAiPho, hemos realizado un estudio con dispositivos reales. Los elementos principales a considerar en el consumo de batería de los dispositivos móviles son las funcionalidades GPS, Wi-Fi, y Bluetooh. Tras realizar un estudio sobre los dispositivos con los que trabajamos hemos obtenido los resultados siguientes: El GPS tiene un consumo de aproximadamente de un 25 % de batería cada 8 hora, mientras que el consumo de la Wi-Fi, Bluetooth y envío de paquetes de datos suman un total de 10 % de consumo cada hora, lo que supone un total de 13,25 % de consumo por hora aproximadamente.

Otra de las aplicaciones asociadas a esta interfaz, es la posibilidad de conocer dónde se ha dejado el vehículo estacionado. Una vez aparcamos nuestro coche se almacenan en la base de datos las coordenadas donde lo hemos dejado. Si se pulsa el botón "find" de la aplicación a través de la interfaz de Usuario se muestra en el mapa la localización donde se ha dejado estacionado el vehículo. Además calcula la ruta desde nuestra posición actual hasta nuestro vehículo tal como se muestra en Figura 4.17.

Figura 4.16: Configuración

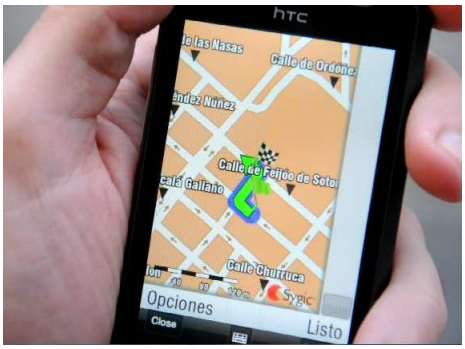


Figura 4.17: Ruta hasta el Vehículo

4.11. Implementación en Dispositivos Reales

En esta sección se incluye un análisis de la aplicación VAiPho realizada gracias a su implementación en dispositivos reales. Para comprobar la eficacia de los procesos de agregación y de verificación, la mejor opción sería ponerlos a prueba en una implementación a gran escala. Sin embargo, hacer esto con un gran número de dispositivos reales no es fácil. Por lo tanto, la alternativa elegida ha sido utilizar la aplicación en unos pocos dispositivos para obtener datos reales de ellos.

La implementación en dispositivos reales se realizó en cuatro dispositivos móviles. Tres fueron utilizados para detectar un atasco de tráfico y crear la correspondiente firma

de paquetes agregados. El cuarto dispositivo se ubicó en un vehículo que no podía detectar directamente el atasco, pero que sí recibió el paquete agregado y por tanto pudo verificar tres firmas. La Figura 4.18 muestra en detalle la situación simulada.

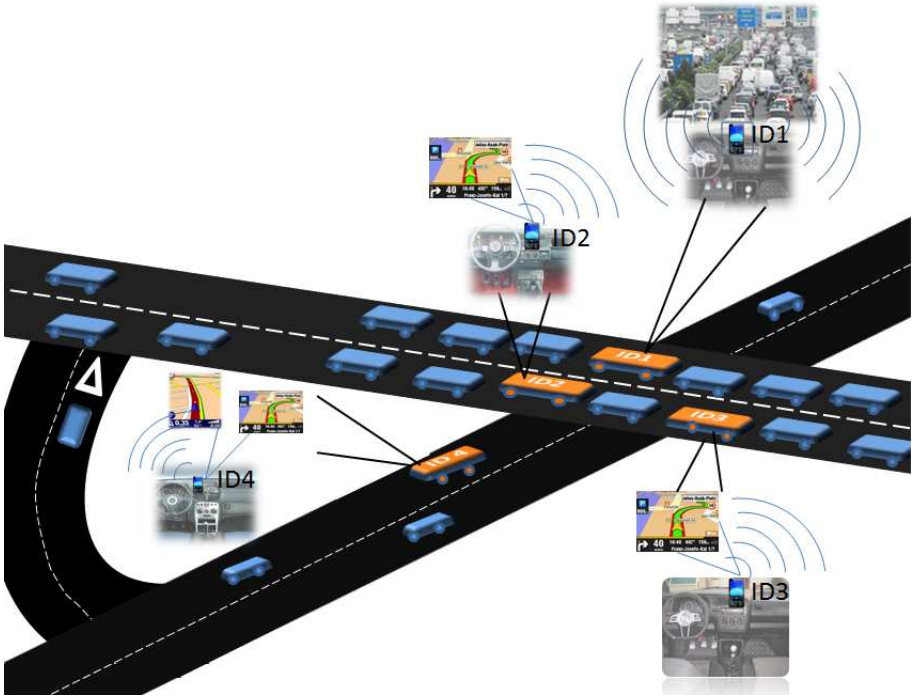


Figura 4.18: Situación Simulada

En la Figura 4.18 los tres dispositivos móviles capaces de detectar los atascos de tráfico se denotan ID1, ID2 e ID3. ID1 es el nodo que una vez detectado el evento, inicia un procedimiento de agregación de datos mediante el envío de un paquete de tipo R a todos los nodos de la red a través de un broadcast. Los nodos ID2 e ID3 e incluso ID4 pueden recibir este paquete ya que se encuentran en el radio de emisión de ID1. Sin embargo, sólo ID2 e ID3 son capaces de detectar el atasco de tráfico y pertenecen al grupo reactivo correspondiente, por lo que firman el mensaje de advertencia y envían el paquete tipo S correspondiente al nodo ID1. Una vez que el nodo ID1 recibe la información, genera un paquete con las firmas de ID2 e ID3 y realiza un broadcast del paquete tipo A agregado. Cuando el nodo ID4 recibe la información, verifica las firmas contenidas en el paquete y si todo está correcto, advierte al conductor acerca de la congestión en la vía y le sugiere una ruta alternativa.

Cuando el sistema arranca, tanto el módulo Servidor como el módulo Vigilante comienzan. Como se muestra en la imagen de la izquierda de la Figura 4.19. El módulo Vigilante recoge la información del GPS, y cuando la velocidad del vehículo es inferior al margen establecido, se indica que hay un posible atasco. Después de unos segundos se comprueban una vez más las condiciones a fin de descartar la posibilidad de un semáforo en rojo o una señal de stop y si la situación sigue siendo igual, se inserta el caso reportado en la base de datos como se muestra en la imagen de la derecha de la Figura 4.19. Todos los nodos son alertados, pero el proceso sólo es iniciado por el primero que detecta el evento. En particular, el nodo ID1 detecta por primera vez un posible atasco de tráfico y después de cierto tiempo comprueba una vez más su velocidad para determinar finalmente que es un atasco de tráfico. A continuación, el nodo almacena el evento en la base de datos y envía un mensaje con la información para formar un grupo reactivo. Se crea un paquete agregado con la firma de todos los nodos que forman el grupo reactivo y que respondieron a su solicitud.

Figura 4.19: Detección de una Congestión

Los nodos ID2 e ID3 reciben el paquete y comprueban la información proporcionada por el módulo Vigilante. Si están de acuerdo, devuelven un mensaje con la información firmada, como puede verse en la Figura 4.20. Como se explicó anteriormente, la firma del mensaje se calcula con la clave privada de cada nodo y una función hash. Una vez firmado, el mensaje se reenvía al nodo ID1 indicando la conformidad con la información recibida.

En la implementación realizada, el nodo ID1 recibe dos firmas de los nodos ID2 e ID3. Entonces, se genera el paquete Agregado, porque en este caso sabemos que no hay más

Figura 4.20: Confirmación de una Congestión

dispositivos en el grupo reactivo. Finalmente, se envía el paquete firmado a la red para que pueda ser retransmitido y que llegue a otros vehículos que no pueden detectar directamente el evento.

El paquete agregado es recibido por el nodo ID4, que verifica las firmas antes de tomar la información como válida. Después de esta verificación, se proporciona la información al GPS para determinar si el vehículo debe continuar en la misma ruta o tomar una ruta alternativa. La Figura 4.21 muestra las coordenadas del evento reportado y el aviso recibido por el dispositivo.

Como resultado de la implementación y prueba de esta aplicación, llegamos a la conclusión de que el esquema de agregación propuesto funciona bien en VAiPho. Varios vídeos que explican el funcionamiento de VAiPho así como detalles de su implementación se pueden encontrar en el sitio web [26].

Figura 4.21: Verificación de Firmas

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

El intercambio seguro y eficiente de información en VANETs, objeto de esta Tesis, se enfrenta a más retos que en cualquier otro tipo de redes ad-hoc, ya sean MANETs o redes de sensores, debido a las características únicas que presentan las VANETs. Primero, la ausencia de infraestructuras y de autoridades centralizadas para controlar y manejar la red aumenta su vulnerabilidad ya que es necesario confiar esas tareas a los usuarios. Por otro lado, la retransmisión de los paquetes requiere de la cooperación de los nodos por lo que las comunicaciones dependen de que su comportamiento individual no disminuya la eficiencia y fiabilidad de la información transmitida. Además, los recursos limitados con los que cuentan los dispositivos móviles utilizados en la implementación aquí propuesta, como memoria, batería, capacidad computacional o ancho de banda, suponen una dificultad adicional a la hora de proponer soluciones a la problemática de la retransmisión. Finalmente, el contenido de la información generado por los nodos de manera individual dentro de la red debe considerarse inherentemente poco fiable dado que en general es fácil generar paquetes con contenido falso, lo que conduciría a desconfianza sobre futura información recibida, y en el peor de los casos, a accidentes.

Con respecto a la cooperación, se han presentado en esta Tesis diversas propuestas basadas en diferentes estructuras para motivar a los nodos en la retransmisión de paquetes.

En la primera parte proponíamos la motivación a retransmitir paquetes de valor añadido en forma de recompensa teniendo en cuenta que la recompensa tenía que ser soportada o bien por una operadora, o por un anunciante en función de sus recursos. El uso de recompensas tenía un importante problema de gasto incontrolado y podía llevar a otros ataques con el fin de obtener más beneficio del que le corresponde a cada nodo. Por lo tanto, diseñamos una nueva propuesta basada en reputación donde los nodos maliciosos quedan aislados de la red, primero de manera individual por parte del nodo que detecta el mal comportamiento, y finalmente de manera global frente a la red mediante la puesta en común de dicha información. Además, otros aspectos que pueden conducir a la falta de cooperación, como el espacio de almacenamiento o el consumo de batería, se han tenido en cuenta a la hora de diseñar el protocolo propuesto para el fomento de la retransmisión de paquetes sobre eventos en carretera. En estas redes la participación de los nodos puede disminuir si son capaces de recibir información de la red sin participar en la misma. Además se ha demostrado que cuanto mayor sea el número de nodos que entren en la red mejor será su funcionamiento dado que tendrá mayor alcance la cobertura de la misma. Con ambos objetivos se ha propuesto el uso de comunicaciones cifradas de manera que cualquier nodo que quiera obtener información de la red, debe obtener una clave que le permita descifrar las comunicaciones. Para implementar este cifrado se propone un nuevo generador pseudoaleatorio que pretende fomentar la captación de usuarios para la red.

Cuando la red crece en tamaño, el rendimiento de las comunicaciones y en general la administración de la red caen drásticamente. Por lo tanto es necesario definir alguna estructura que permita llevar a cabo este proceso de manera ordenada. En esta Tesis se han propuesto dos estructuras para la retransmisión: según una estructura de árbol y según una estructura de grupo, siendo esta última la mejor opción para nuestras propuestas. De hecho, nos centramos en la formación de grupos reactivos, que se forman solo cuando son necesarios. La estructura de grupos no solo permite mejorar las comunicaciones disminuyendo el número de paquetes en la red sino que genera información de confianza dentro de la red. En particular, en la estructura de grupo propuesta existe un nodo que se conoce como líder del grupo, que es el encargado de gestionar las comunicaciones dentro de su grupo así como de generar los paquetes de agregación. Si bien es posible que los nodos intenten atacar la

red generando información falsa, la estructura de grupo impedirá alcanzar este objetivo al tener todos los nodos pertenecientes a un mismo grupo la misma visión promiscua sobre su entorno. Por tanto la formación de grupo reactivo junto con la agregación de datos protege contra este tipo de ataques.

El problema de la generación de información falsa no solo es afrontado con la estructura de grupo sino con la toma de decisiones sobre si agregar o no y con el mecanismo de agregación concreto propuesto. En cuanto a la toma de decisiones se propone el uso de lógica difusa sobre 4 parámetros: espacio, tiempo, velocidad y dirección. Por otra parte, teniendo en cuenta la alta movilidad de los nodos en la red, las propuestas de agregación deben ser rápidas tanto en generación como en comprobación de la información. La generación de información queda resuelta con la estructura de grupo, que permite hacerla de manera eficaz. La comprobación se realiza una vez el nodo recibe la información ya que según su posición respecto al evento recibido urgirá más o menos proporcionársela al conductor. Por tanto se propone un nuevo protocolo de verificación probabilístico donde no todas las firmas sino un número suficiente serán comprobadas para determinar la veracidad de la información recibida. El proceso completo está protegido contra ataques tanto de generación de información falsa, como de duplicación de información.

En la comunidad científica se ha estado buscando una solución que permita la gestión eficiente del tráfico a través de la creación de redes vehiculares que faciliten la comunicación entre vehículos, para proporcionar de esta forma información en tiempo real sobre los diferentes eventos que surgen en la carretera. Sin embargo, implantar dicha solución mediante RSUs en las carreteras y OBUs en los vehículos, tal como presenta la bibliografía actual, implicaría un gasto tanto para las administraciones públicas encargadas de instalar y mantener la infraestructura de comunicaciones en carreteras, como para los usuarios quienes tendrían que adaptar o cambiar sus vehículos. Gracias a la aplicación propuesta en esta Tesis, denominada VAiPho, se pueden evitar esas inversiones económicas, poniendo al alcance de cualquier conductor el uso de una red vehicular. Además nos ha ofrecido a nosotros como investigadores la oportunidad de poner a prueba algunos de los protocolos propuestos de manera rápida y económica en un entorno real donde efectos que no era posible reflejar en un simulador, como el desvanecimiento de la señal o fading, la pérdida de comunicación, las

interferencias, etc., surgen de manera natural. Además hemos demostrado que un teléfono inteligente es capaz de ejecutar los algoritmos propuestos siendo el gasto computacional asumible. Por lo tanto podemos asegurar que el uso de esquemas criptográficos no suponen un cuello de botella para la autonomía y disponibilidad de los dispositivos.

5.2. Resultados de la Tesis

Las principales contribuciones de esta Tesis se resumen en los siguientes puntos:

- Se ha abordado el tema de la cooperación para VANETS en general, y para una nueva propuesta de VANET totalmente distribuida y descentralizada en particular. Se han propuesto varias contramedidas para evitar el comportamiento no cooperativo ofreciendo nuevas soluciones prácticas para VANETs en las cuales no hay necesidad de autoridad centralizada. Así, se han desarrollado herramientas que permiten formar una red autogestionada usando la tecnología existente, de modo que los nodos puedan recibir y enviar información sobre tráfico a través de sus dispositivos asegurando la cooperación de los nodos participantes. Esto ha permitido abordar diferentes problemas de seguridad en retransmisión y rendimiento en este tipo de redes a través de soluciones gratuitas basadas en la cooperación de los usuarios que han implementado los esquemas propuestos en sus dispositivos. En particular, se ha propuesto el uso de dos listas de reputación y mensajes de acuse de recibo así como diferentes mecanismos en base a parámetros como el tiempo y la distancia que permiten a los nodos detectar automáticamente los malos comportamientos con el fin de aislar a los nodos maliciosos. Se han realizado numerosas simulaciones prácticas de la propuesta demostrando su solidez y utilidad en escenarios VANET, especialmente en condiciones de tráfico denso como congestiones de tráfico demostrando que se logra reducir el número de nodos egoístas en VANETs. Los resultados han sido publicados en los siguientes trabajos [25], [37], [38], [70], [71], [97], [98], [99] , [100], [102], [104].

- Se ha presentado el diseño de un nuevo generador de secuencia cifrante basado en un filtrado no lineal de un registro de desplazamiento con realimentación lineal. El objetivo de la propuesta ha sido usarlo en VANETs para prevenir la existencia de

nodos ajenos a la red que intenten beneficiarse de ella accediendo a la información transmitida. Nuestro generador ha sido analizado mediante diversos tests estadísticos, obteniendo resultados que confirman las buenas propiedades pseudoaleatorias de la secuencia de salida. Adicionalmente este generador ha sido ajustado para funcionar en tecnología RFID según el estándar EPC Gen 2 con tecnología de 16 bits. Esta temática ha sido motivo de las siguientes publicaciones [27], [28], [106].

- Se ha propuesto una nueva solución para satisfacer la necesidad de abordar el problema de seguridad en VANETs consistente en determinar si la información de tráfico disponible para el conductor es fiable o no. En particular, se ha descrito un nuevo esquema para generar paquetes agregados que no pueden ser sustituidos por ningún adversario, ya que contienen las firmas de los vehículos que están de acuerdo con el evento notificado. Tanto para evitar que los paquetes de alerta crezcan indefinidamente, porque la escalabilidad es un aspecto primordial en VANETs, como para añadir confianza a los mensajes de advertencia, se generan las firmas de acuerdo a la formación de grupos reactivos. Por otro lado, cuando un paquete agregado llega a un vehículo, con el fin de evitar la demora producida por la verificación de firmas en entornos densos, se ha propuesto un esquema probabilístico según el cual sólo unas cuantas firmas son elegidas para ser revisadas. Se ha evaluado el rendimiento de los grupos reactivos para la generación de eventos y la detección de mala conducta, y los resultados han confirmado que este es un enfoque prometedor para incrementar la eficiencia del canal y la confianza en la información transmitida. Por otra parte, el sistema propuesto ha sido revisado en un entorno real y los resultados obtenidos han permitido tanto resolver varios problemas reales que no aparecen en los entornos de simulación, como obtener los datos utilizados para hacer simulaciones NS-2 a gran escala, que también han producido resultados prometedores en cuanto al tiempo requerido por el sistema. Las publicaciones resultantes de este trabajo son [36], [99], [107], [108].

- Con objeto de realizar la implementación real de los protocolos propuestos en esta Tesis ha surgido una nueva aplicación software para la asistencia a la conducción, llamada VAiPho (VANET in Phones). VAiPho permite detectar atascos y otros even-

tos de tráfico de forma automática y en tiempo real mediante teléfonos móviles. La retransmisión de estos eventos en VAiPho se lleva a cabo mediante comunicaciones seguras entre los dispositivos que forman la red vehicular. Su objetivo principal es proporcionar al conductor información sobre los eventos detectados automáticamente en la carretera que puedan ser de su interés. En concreto la detección de atascos, la detección de posibles plazas de aparcamiento público libres, la localización del vehículo aparcado y la oferta de publicidad geolocalizada son funcionalidades ya incorporadas a VAiPho. La implementación de VAiPho ha permitido comprobar la validez de gran parte de las propuestas presentadas en esta Tesis en un entorno real y con dispositivos reales ya que hemos desplegado lo que podría ser la primera VANET real utilizando solo teléfonos móviles dentro de los vehículos. En particular, VAiPho ha sido implementado en una versión Beta a modo de prueba en dispositivos con sistemas operativos Windows Mobile, Android y Symbian, lográndose hasta el momento compartir información sobre diferentes eventos entre móviles con la misma plataforma, y llegando a la interesante conclusión de que funciona correctamente en este tipo de redes. El impacto mediático de VAiPho, tras ser patentado y presentada su demostración real en vehículos frente a empresas, entidades y medios de comunicación, ha sido abrumadoramente positivo, apareciendo en numerosos noticiarios, diarios y blogs digitales, así como en múltiples entrevistas en programas de radio y televisión, y medios de comunicación de la prensa escrita [134]. Además la idea ha sido merecedora del primer premio en el concurso de emprendedores “Conocer es Valer” [23] de la Universidad de La Laguna en su edición de 2011, ha sido presentada en [25], [33], [34]y ha sido objeto de un proyecto fin de carrera dirigido que obtuvo la máxima calificación [44].

5.3. Trabajos Futuros

- **Cooperación**

  La cooperación en la retransmisión de paquetes en la red depende entre otras cosas del comportamiento de los nodos, de forma que se hace necesario compensarlos no solo mediante el acceso a la información transmitida, sino también a través de recompensas

por su buen proceder, y castigos en caso de comportamientos maliciosos. Dado que es imposible determinar a priori cuál será el comportamiento de los usuarios en la red, sería necesario estudiar su comportamiento social como red para poder mejorar las propuestas. La mejor forma para hacerlo es implementando los sistemas de recompensas y castigos en VAiPho. Sin embargo esto será una tarea complicada ya que al principio del despliegue el número de usuarios será bajo, por lo que las medidas de seguridad, como la revocación de certificados, no podrán ser muy restrictivas. Por lo tanto el siguiente trabajo en el que nos centraremos en cuanto a la cooperación será la implementación de las listas LIR y LGR en la aplicación VAiPho. En cuanto al generador, solo ha sido testeado frente a algunos tests estadísticos, dado que no se trataba del principal objetivo de esta Tesis, luego han quedado abiertas numerosas cuestiones al respecto, como someter el generador a otras baterías de tests interesantes, y en caso de que los resultados no fueran satisfactorios, modificar su estructura.

- **Agregación**

  Los esquemas propuestos para detección automática de eventos proporcionan plena confianza sobre su veracidad ya que permiten la detección de mensajes falsos, erróneos o malintencionados. Dichos esquemas han producido unos resultados bastante buenos. Sin embargo, sería interesante también validar los sistemas no sólo con respecto a los tiempos necesarios para llevar a cabo la agregación y la verificación de datos, sino también en cuanto a su capacidad para detectar diferentes tipos de ataques. Otro objetivo futuro relacionado es comprobar la exactitud de los esquemas propuesto en la práctica a través de la implementación a gran escala en dispositivo reales y en entornos reales, que permitan evaluar la influencia de la velocidad y edificios en las comunicaciones inalámbricas, y el funcionamiento en escenarios densos. Además nos planteamos aplicar la propuesta de fusificación a la toma de decisiones.

- **VAiPho**

  Tanto en cooperación como en agregación, parte de los trabajos futuros consiste en comprobar el funcionamiento de las propuestas en entornos reales a gran escala. Para ello será necesaria la comercialización y difusión de VAiPho de manera que muchos

usuarios puedan acceder a ella. Como trabajo futuro por lo tanto destaca la implementación completa, eficiente y multiplataforma de VaiPho, como una herramienta totalmente funcional que nos aporte una especie de feedback desde los usuarios para mejorar cualquier problema que pudiera presentarse, suponiendo esta una gran oportunidad de testeo real de los algoritmos, pues al estar en manos de una gran variedad de personas podrían proporcionarse diferentes puntos de vista a las soluciones propuestas hasta el momento.

Por último, el inmenso abanico de posibles nuevas aplicaciones que abre la aplicación VAiPho es realmente espectacular. Actualmente VAiPho permite detectar congestiones y atascos, posibles plazas de aparcamiento libres, situación exacta donde se ha dejado aparcado el vehículo, además de ofrecer una plataforma de publicidad geolocalizada. Sin embargo, en el futuro podrán añadirse de forma fácil e intuitiva, muchas más funcionalidades a VAiPho, gracias al diseño modular que se ha seguido en su implementación. Además la herramienta podrá ser utilizada en cualquier otra aplicación que no sea de VANETs, dado que la base del sistema es permitir las comunicaciones seguras entre teléfonos móviles. Por tanto, los límites que pueda tener VAiPho en el futuro son, por ahora, inimaginables.

NEW COOPERATIVE SECURITY TOOLS FOR VEHICULAR AD-HOC NETWORKS

EXTENDED ABSTRACT

Appendix A

Contributions

The summary of the main contributions of this Thesis in the field of VANETs can be structured as follows:

1. **Countermeasures to prevent selfish and/or malicious behaviour.**

   We propose a series of cooperation mechanisms and isolation of selfish and/or malicious nodes inside the network to try to prevent and/or avoid possible selfish behaviours and malicious attacks, by using the revocation of those nodes, to ensure that VANETs work properly, providing users with reliable information in real time. To achieve this goal, nodes must cooperate actively relaying messages to reach as many nodes as possible. The operation of the proposals have been evaluated with different simulations, showing not only more efficiency but also increased security of communications. In this area, we published four papers in LNCS volumes [12], [38], [70], [98] presented communications in four indexed conferences [97], [99], [100], [102] and in four non-indexed conferences [37], [71], [96], [104] of which the last paper received the "Best Paper Award" of the conference.

2. **Pseudorandom generator to recruit users.**

   For this topic we propose a new secure communication system for VANETs that promotes the recruitment of new nodes and prevents access to information by nodes that do not belong to the network. In particular we propose a stream cipher that uses

a new pseudorandom number generator whose design is based on a nonlinear filtering of a linear feedback shift register. The proposed generator is included in a paper that has been accepted for its publication in an indexed journal [28]. On the other hand, the cipher has been included in a volume of the LNCS [106], and presented in an indexed conference [27].

3. **Data aggregation for authentication.**

   One of the main difficulties in vehicular networks, where traffic information is generated by the most diverse and varied sources and is relayed to many destinations, is the need for a data authentication mechanism to detect any malicious behaviour of users, who launch attacks such as modification or repetition of information. This Thesis proposes a new aggregation protocol for VANETs, which uses probabilistic verification to detect these attacks in an efficient way with minimal overload and delay. The proposal also contains an additional security mechanism that uses the idea of reactive groups created on demand, to ensure that the generated information is reliable. According to a comprehensive analysis of the proposal including numerous simulations, we demonstrate that the proposal is robust. The most remarkable results concerning this topic are included in a paper that is under third round review in a journal with a high impact factor [103]. Preliminary versions had been published in a volume of LNCS [36], and presented in an indexed conference [108] and a non-indexed conference [107].

4. **Implementation in real devices.**

   During the development of this research we identified as a problem the implementation of the proposals in a real environment with the aim of analyzing their performance because the standard defined for VANETs and the environment described in the bibliography for the definition of VANETs have not yet been developed. On the one hand, the OBUs are devices installed in vehicles including different parts such as sensors, and therefore represent a cost impossible to cover for the realization of a Thesis in an academic enviroment. Furthermore, it was assumed that RSUs would be implemented in 2011, but currently they are not present on the road to allow the formation of

VANETs, according to the original definition. We thought about the possibility of deploying these networks without any cost in order to continue with this research and reach a real implementation of the proposals. A lot of literature propose vehicles communicate via Wi-Fi, using the 802.11 protocol. Current smartphones provide this type of connection, so in this Thesis we decided to use programmed phones to develop a real VANET. In this way we managed to implement some of the proposed protocols on mobile devices, using them as OBUs inside vehicles in order to get real data and analyze their performance. This idea of setting up secure VANETs using mobile phones has been called VAiPho (VANETs in Phones). It is remarkable that VAiPho implies a fully distributed and decentralized network with no central authority or RSU as opposed to the proposals found in the literature. There are many implementation problems in mobile phones that we have had to achieve this development. This tool collects implemented solutions related to cooperation, aggregation and encryption proposals. Therefore, we can say that VAiPho is the main contribution of this Thesis, as this invention has been patented by the University of La Laguna in 2010 in its national phase, and is currently in the process of internationalization [26]. In 2011 the license of such a patent has been acquired for its exploitation by the domestic company based in Madrid, DETECTOR S.A. Furthermore, VAiPho has been awarded the first prize in the contest of entrepreneurs "Conocer es Valer" [23] of the University of La Laguna in its 2011 edition. VAiPho has been presented in three conferences [25], [33], [34], and an end-term project [44].

A.1. Indexed Journals and LNCS

- [12] 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.

- [16] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Merging subnetworks in VANETs by using the IEEE 802.11xx protocol. Submitted to Eurasip Journal of Wireless Communications and Networking, 2011.

- [17] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organized clustering architecture for Vehicular Ad-hoc Networks. Submitted to Journal on Cluster Computing, 2011.

- [19] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Self-organized Life Cycle Management of MANETs, Accepted by Security and Communication Network, 2012.

- [20] Caballero-Gil, C., Caballero-Gil, P., Molina-Gil, J. Zero-knowledge authentication in self-organized VANETs. Submitted to IETE Journal of Research, 2011.

- [22] Caballero-Gil, C., Caballero-Gil, P., Peinado-Domínguez, A., Molina-Gil, J. Lightweight authentication for RFID used in VANETs. Computer Aided Systems Theory EUROCAST 2011, Lecture Notes in Computer Science, Vol. 6927, 2011.

- [24] Caballero-Gil, C., Molina-Gil, J., Caballero-Gil, P. Design and implementation of VAiPho, tool for deploying VANETs with phones. Submitted to Computers & Electrical Engineering, 2011.

- [28] 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.

- [32] 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.

- [36] 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.

- [38] 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.

- [69] 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.

- [70] 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.

- [72] 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.

- [98] 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.

- [101] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Countermeasures to Prevent Misbehaviour in VANETs. In second round review at Journal of Universal Computer Science, 2011.

- [103] Molina-Gil, J., Caballero-Gil, P., Caballero-Gil, C. Probabilistic Aggregation for Data Authentication in VANETs. In third round review at Transportation Research Part C, 2011.

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

- [106] Molina-Gil, J., Caballero-Gil, P., Fúster-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.2. Indexed Conferences

- [13] 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.
- [29] 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.
- [30] 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.
- [97] 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.
- [99] 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.
- [100] 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.
- [102] 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.
- [108] 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. Other Conferences

- [11] 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.

- [14] 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.

- [15] 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.

- [18] 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.

- [21] 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.

- [25] 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.

- [27] Caballero-Gil ,P., Caballero-Gil, C., Molina-Gil, J. An EPC Gen2 compliant authentication scheme based on a new pseudorandom number generator. FTRA International Workshop on Strategic Security Management for Industrial Technology, 2011.

- [31] 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, 2009.

- [33] 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. SALT, 2011.

- [34] 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. TRANSNOVA, 2011.

- [37] 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, 2009.

- [71] 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.

- [91] Martín-Fernández, F., Caballero-Gil, P., Caballero-Gil, C., Molina-Gil, J. Implementación de comunicaciones seguras en plataformas móviles para asistencia a la conducción. Submitted to X Congreso de Ingeniería del Transporte. 2012.

- [96] 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.

- [104] 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.

- [107] Molina-Gil, J., Caballero-Gil, P., Hernández-Goya, C., Caballero-Gil, C.: Agregación de datos para autenticar información en VANETs XI Reunión Española sobre Criptología y Seguridad de la Información RECSI. 2010.

A.4. Other Contributions

- [23] 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.

- [26] 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., en Junio de 2011.

- [44] Yanes-García D. End-term Project Directed by Caballero Gil P., Molina Gil J. Implementación de comunicaciones seguras en la plataforma Android para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), June 2011.

Appendix B

Cooperation

Vehicular Ad-hoc NETworks (VANETs) are generally defined as Mobile Ad-hoc NETworks (MANETs) formed by vehicles. Their main goal is to provide information to the drivers so that their deployment permit enhancing safety, efficiency and comfort in every day road travel. In these networks, warning messages affect drivers' decisions, so any incorrect message could lead to increase the time required to reach the destination, fuel consumption, environmental pollution and, in the worst case scenario, traffic accidents.

VANETs, as smart technologies in Intelligent Transportation Systems (ITS), have become a hot topic in network research. In the near future, such networks will reduce the number of deaths from traffic accidents, by providing real time information about traffic and road status. In addition, VANETs might be used for other practical applications such as finding free parking spaces, for example.

Research on ad-hoc networks raised the need for enhancing cooperation between nodes as a prerequisite for their proper operation. VANETs, as an evolution of these networks, also have need for cooperation. However, due to characteristics such as high mobility, real-time constraints, scalability, gradual deployment and privacy, VANETs present additional challenges. A key aspect for the proper functioning of the network is to provide reliable and real time information about traffic and road conditions. To achieve this goal, nodes must cooperate actively sending received events warning to the nodes they find during their life on the network. It is possible that malicious users try to get benefit from the infor-

mation provided by the network and at the same time minimize their battery consumption and storage space, configurating their devices to receive information from the network but without cooperating in relaying it. Therefore, the existence of some mechanisms to prevent these nodes affect the network performance is necessary.

There are many situations where communication among vehicles and cooperation in relaying packets can help to prevent accidents and to avoid collapses. Nevertheless, the behaviour of selfish nodes could break the network into pieces causing a passive Denial of Service (DoS). It is rational to assume that each node has the target of maximizing its own benefit by taking advantage of the network services while minimizing its own contribution to the network. Therefore, the need to motivate nodes to relay information for the benefit of other nodes is justified.

This work proposes the use of cooperative tools that can be implemented with current technology, such as laptops, smartphones, etc., provided with Global Positioning System (GPS) equipment and wireless networking communication. The goal of this work is to create a vehicular ad-hoc network using these technologies inside cars so that they can be also used as an emulation of the devices that will be implemented in future cars to form VANETs. Hence, real data obtained from these networks will be useful for the analysis of the operation in future VANETs.

Nowadays, there are several GPS software applications that provide drivers with information about traffic conditions compiled by local traffic authorities, police departments or other centralized systems. However, in most cases the information provided to the driver is not in real time because it does not reflect events that have just produced and/or involves lack of user privacy due to the centralized operation. In addition, most software tools that provide this service, such as Google Traffic application, require 3G connection, which represents an additional cost for users. Therefore, the motivation to study the secure and efficient deployment of new self-organized and cooperative proposal of VANET deployment, instead of the existing GPS software applications is clear.

This chapter introduces a mechanism to provide real and reliable information to those vehicles actively involved in the correct operation of the network. The scheme includes a decentralized revocation system of selfish and malicious nodes, using node cooperation

and isolation of attackers, based on the use of reputation lists and rewarding mechanisms.

B.1. Related Work

In order to bring VANETs to their full potential, appropriate schemes to stimulate cooperation in transmitting and forwarding packets need to be developed according to the specific properties and potential applications of VANETs. Indeed, cooperation enforcement has been a hot research topic in MANETs. Buttyan and Hubaux proposed in [7] and [8] the use of virtual credit in incentive schemes to stimulate packet forwarding in MANETs. Also, cooperation in MANETS was studied in [125] where the assumption that each node has the goal to maximize its own benefit by enjoying network services and at the same time minimizing its contribution was presented. There it was proposed that nodes were encouraged to relay information for the benefit of other nodes through a charging and rewarding scheme. However, solutions in MANETs are not directly applicable to VANETs mainly due to their high node mobility, limited connectivity and large scale. Thus, Li et al. discussed some unique characteristics of incentive schemes for VANETs in [85] and proposed a receipt counting reward scheme that focuses on the incentive for spraying. However, the receipt counting scheme proposed there has an overspending problem. Based on the specific characteristics of VANETs, a more comprehensive weighted rewarding method was proposed in [70]. Anyway, most existing solutions based exclusively on rewarding mechanisms suffer from lack of fairness assurance and reliance on costly tamper-proof hardware or on-time trusted third parties. These problems are not present in the proposal of this work.

Malicious attackers can cause the VANET to be broken into pieces so that the network cannot provide services such as route establishment and packet forwarding to legitimate users. In this sense, the behaviour of selfish nodes can cause a passive DoS. [77] discusses some of the main security threats and attacks that can be exploited in VANETs. Similarly, [109] uses routing for communications, and introduces cooperation as a service, based on a cluster structure. Both papers are mainly focused on the design of routing schemes, unlike the present proposal.

Several authors have proposed combined approaches to the topic of cooperation in

VANETs. The proposal called VARS described in [51] uses direct and indirect trust as well as appended opinions to enable confident decisions on event packets. The main problem of such a proposal is that it involves accumulating reputation evaluation over much time. [56] gives an overview of existing approaches that try to provide routing security to conventional MANETs and analyzes whether these approaches can be applied to secure VANETs. To promote node cooperation and to protect VANET packets during propagation, [138] proposes a dynamic trust-token based cooperation enhancement mechanism. Another interesting reputation system was described in [137], where trust relationships and packet-acceptance decisions are based on instant observation and relaying behaviour of nodes. However, both watchdog schemes have some drawbacks, implying that tampered data packets may be propagated.

[145] proposes a flocking scheme for a group of vehicles, which focuses on their decentralized coordination so that they can cooperate in complex environments. A good example of VANET application that requires cooperation is described in [83], which proposes a framework for commercial ad dissemination in VANETs where vehicles receive an incentive for forwarding and carrying advertisements. Unlike the above works based on rewarding mechanisms, several recent reputation schemes have been proposed based on node behaviour with respect to its collaborative operation as monitored by other nodes. [88] used an event-based system to prevent nodes from spreading false traffic messages by determining whether incoming traffic messages are significant and trustworthy to the driver. [126] described a mechanism for detecting possible malicious nodes through the use of three different modules whose sum up determines node reputation. However, all the aforementioned tools, including the reputation system proposed in [121], require Certification Authorities (CAs) that are responsible for delivering public/private keys and certificates. In particular, [129] proposed that such a role is played by a regional transportation authority, which can be a state, province, etc, while other authors proposed a Department of Motor Vehicles [86]. Therefore, none of those solutions can be considered applicable to the fully distributed and decentralized networks discussed in this work, where several countermeasures combining rewarding and reputation ideas are proposed.

B.2. Cooperation in Retranmission of Value-Added Packets

Our protocol uses two different schemes depending on the type of packets that are being relayed on. In both cases it uses the idea of groups where nodes driving at a similar speed in the same direction, join together in the same group. This idea of groups permits to reduce the number of packets in the network because the leader is in charge of making decisions about the packets that reach its group. This prevents from having multiple retransmissions of the same packet. Now we analyze in detail how our model behaves for the two types of studied packets:

B.2.1. Internet Packets

This type of packets allows users to connect to Internet in order to browse like they were in front of their computers or mobile phones. We assume that this service has been contracted in advance with an Internet operator. Consequently, we assume that there is some user who has contracted a service, an operator that offers this service and a set of groups of vehicles that carry out the connection between the source and the destination. Nodes responsible for packets retransmission might decide not to transmit Internet packets. In this situation the operator must use some cooperation mechanism in order to provide the service to their customers. In this paper, we propose that the payment is a valuable resource for nodes, such as for example petrol. In order to obtain this valuable resource that is petrol, the nodes will be motivated to cooperate. Each intermediate node should be encouraged to retransmit those packets that allow the connection between the clients and the operator. The cost of this retransmission should be covered by the operator to which the user pays for this service.

With Internet packets, when an initiator A wants to communicate with Internet, it has to set up an end-to-end session with a Road Side Unit. In order to set up a session, A generates a request message and broadcasts it. Each intermediate node in a group receiving the request, authenticate itself including a pseudonymous authentication [39], in the packet, and sends it to the leader of it group. The leader checks the traffic information, puts its pseudonymous authentication in the packet and looks for a forwarding node in the group.

When the request arrives to the RSU, it returns a session setup reply and sends a session setup confirmation message towards A. The session becomes active on the one hand for the RSU when it sends the confirmation messages and on the other hand for the vehicles when they receive a valid confirmation message (see Figure B.1). The goal of using pseudonymous authentication is to sign messages so that the operator knows the identity of cooperative nodes and rewards them without the other relay nodes can discover it.

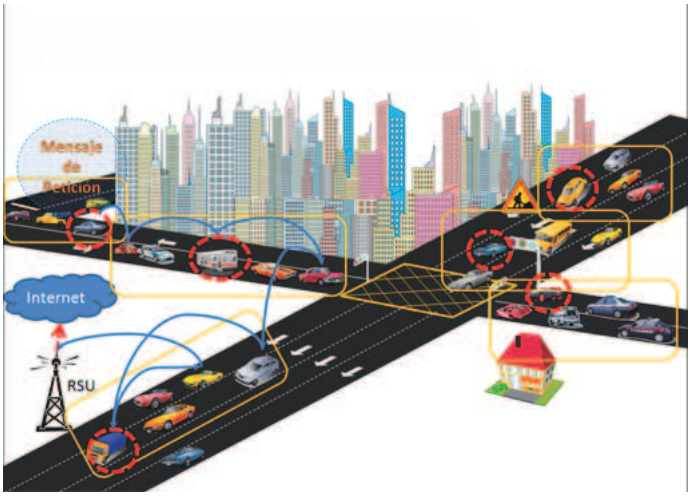


Figura B.1: Internet Packets

In order to perform this connection there are two different types of packets, those responsible for establishing the session between client and operator, and the individual data packets:

1. *Establishing session Packets:* When a node A wants to connect to Internet, it has to find a route to a RSU that provides this service. When the node has no direct connection with a RSU, it will have to find a route through one or more groups of vehicles. Inside each group, the packet may arrive through any node belonging to the group, who will relay it to the group leader. The group leader has knowledge of the group structure, so it can determine the node inside the group that must send this packet to reach a base station or another group closer to the next hop towards the destination. In addition, the leader will establish a route from its group to the RSU by using some node from its group.

2. *Data Packets:* Once the connection is established, data packets are retransmitted without having to go through the leader node. In this way, the leader will not have to receive all these data packets and can focus on its actions as leader. Consequently, each node in charge of relaying data packets within the group will make an estimation of how many packets can be relayed before losing coverage with the RSU or with another intermediary group in the communication.

The operator must distinguish between session packets and data packets because the number of data packets is much larger. Therefore, it may not be profitable for the operator to pay an amount that depends on the number of necessary packets. To solve this problem we propose that the nodes retransmitting data packets must be paid a reward q significantly less than one ($0 < q << 1$) for every transmitted packet. As we mentioned above, a node can estimate the number of data packets that can relay before losing the connection with a base station or another group of nodes. Call n_d to this number of packets and q to the reward for broadcasting each one of these packets. Therefore in a basic scheme such a node would obtain as total reward Q_d:

$$Q_d = q \cdot n_d \tag{B.1}$$

The operator will know who has broadcast packets because the packets are signed.

On the other hand, we have the packets used to establish a connection. Although the number of these packets is less than the number of data packets, the reward should be the same because it is impossible to connect the source and the destination node without establishing a session. Therefore, the reward for establishing session packets for leader nodes should be a value Q_s greater than the reward Q_d for retransmitting data packets.

When a session is established, the leader is in charge of finding the best route among the nodes of its group and the base station. Once this connection is established, the leader node is not longer part of this communication. However, if the leader decides not to establish the connection session, the communication would be impossible. So, the total reward received by the leader for broadcasting and calculating the best route for Internet packets will be Q_s. This would not be fair for the leader if the reward of a node selected as

the relay in an Internet communication is greater than the leader reward, because then it would prefer not to be leader. As we explained before, the reward for being leader is greater than the reward of being a relay node even when the amount of packets broadcast by a leader for establishing an Internet connection is much smaller than any relay node of their group. Hence the leader will be motivated to cooperate and it will want to be leader.

B.2.2. Advertising Packets

In dissemination of commercial advertisements the provider sends out commercial ads, and the nearby receiving vehicles start to disseminate them to other vehicles while they are moving by using leader groups B.2. These ads are forwarded for a certain period of time and distance from source provider. Inspired by a micro-payment scheme [78], each payment for a forwarding service can be thought as a lottery ticket. Upon receiving it, both the payee and the winner node can determine whether it is a winning ticket or no. The payee will not only pay to the node with the winning ticket but also to the node that received the forwarded packet.

For this type of packets the ad provider sends out commercial ads and sends them to all vehicles that are in its scope. In this case the number of generated packets is relatively higher than Internet packets because the aim of such packets is to provide advertising to as many vehicles as possible. If all vehicles were devoted to covering this type of packets without any control, the network would be overloaded. Although the number of generated packets is larger than with Internet packets, thanks to the idea of groups, the leader manages to relay them in an orderly manner B.2 .The leader will receive the packet and will be responsible for broadcasting it within its group. The result is that no nodes will receive the same packet many times from its neighbours because if a leader receives a packet that had previously received, it will not relay it again to its group. Moreover, the leader will seek a route within its group to a neighbour group through one of the members of its group nodes. Hence it achieves in reducing both the number of retransmissions between groups and the number of retransmissions inside the group.

Our model proposes a kind of lottery in which each node will have a probability of being winner. Each ad provider generates a packet that contains a unique identifier *PackID*

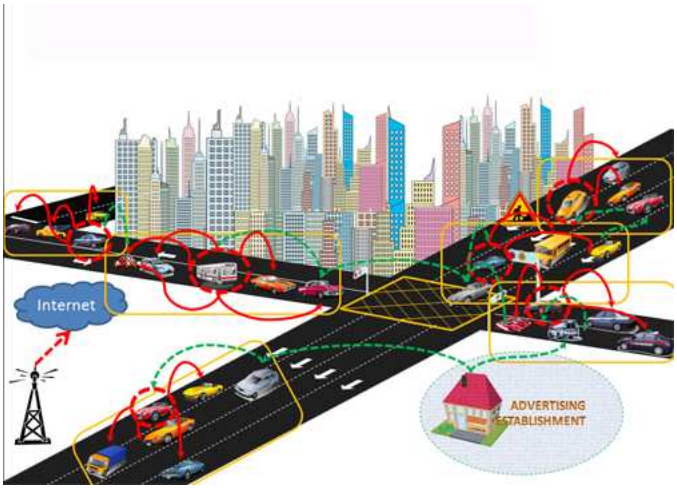


Figura B.2: Adertising Packets

the ad information $AdInformation$ and a hash code H computed randomly with a certain size:

$$[PackID\,|AdInformation|\,H]$$

When a node N receives the packet, it checks the information. If N decides to participate in the forwarding, it sends the message to other nodes i and waits for rec_{N_i}. Then the node N computes for each child node rec_{N_i}a hash on $PackID$, $NodeID_N$ y rec_{N_i}, and checks the result against H.

$$h(PackID\,|NodeID_N|\,rec_{N_i}) = H \tag{B.2}$$

If the equality B.2 fulfills in one of these verifications, then the node N is a winner. We denote by $Prob_h$ the probability that a hash on PackID concatenated with NodeID and the receipts rec_{N_i} that child nodes send to a relaying node collides with a value

$$Prob_h = Prob[h(PackID\,|NodeID_N|\,rec_{N_i}) = H] \tag{B.3}$$

It is assumed that a node can receive only one reward, the probability of a relaying node winning a prize $Prob_P$ in forwarding packets to N_c nodes, where it received the packet from a number of nodes N_f, could be defined as:

$$Prob_P = (N_c + N_f) \cdot Prob_h \tag{B.4}$$

As we explained above and showed in B.4 a node can get a reward if it computes the hash of the packet with the receipts from some of its children and gets a winning code. Furthermore a node can also get a reward if it sends the winner receipt to it father. Therefore a node can transmit packets or receipts to get an award. Hence, the greater the number of retransmissions is, the greater probability of winning. In this way nodes are motivated to cooperate. Moreover, this mechanism will motivate child nodes to send the receipts to node *N*. However, if we analyze the probability , we find that when a node gets a winner receipt, it will not broadcast more packets since the previous function restricts a only one reward for winner. This behaviour would not be desirable since the objective is to motivate relaying all packets. To solve this problem we propose to usea hash function with non-negligible probability of collision, which leads to the possibility of existing more than one winner receipt for each packet:

$$\exists i \neq j : \; h(PackID\,|NodeID_N|\,rec_{N_i}) = h(PackID\,|NodeID_N|\,rec_{N_j}) = H \tag{B.5}$$

In this case, a node can win the lottery for each packet it relays. So, nodes can win more than once with the same packet. On the other hand, nodes can also win the lottery by every receipt they return to their parent node. Similar to the previous case, nodes could win the lottery by one or more receipts. Hence, the problem that once a node wins the award, it ceases to retransmit the same packet is solved. Then if a node wins a prize, this does not mean that it cannot win another prize. According to the group structure introduced, a leader will receive all ad packets in its group. Hence, it will have a bigger probability to receive receipt that produce a hash collision with H so that it could be a winner node. The probability of a leader to win a prize in a group consisting of G nodes, which receive the packet from one node of the group, could be defined as:

$$G \cdot Prob_h \tag{B.6}$$

Therefore, it provides an incentive for leader to propagate ad packets, because the higher the number of retransmissions, the greater the probability of winning a reward. As leader, the packets are broadcast to all of member of its group, the model promotes that nodes prefer to be leaders, and consequently this mechanism motivates the nodes to become leaders and cooperate.

B.3. Cooperation in Retransmission of Road Events

The main aim of this work is to prevent a bad behaviour that could endanger the network connectivity and its proper functionality.

On the one hand, we present a mechanism capable of automatically detecting malicious nodes who try to transmit false information about the existence of an event. For this purpose, we present a tool that permits to determine this kind of attack through the cooperation of neighbouring nodes and prevents such attackers from geting benefit from the information relayed within the network.

On the other hand, this work proposes a mechanism that encourages nodes to participate in packet retransmission and isolates nodes that do not cooperate in packet relaying.

The first mechanism for false information detection is based on reputation schemes whereas the second mechanism for non-cooperation detection uses the rewarding paradigm.

As shown above, both the detection of events and packet retransmission present the possibility of attacks. In both cases, the solution is to detect and isolate malicious nodes from the network. For this purpose, we use cryptographic security tools that allow us to guarantee authenticity of information, privacy of users and non-repudiation on generated information. Thus, public-key digital signature schemes are used to link author and content of each sent message and, like in [9], pseudonyms are used to avoid disclosing real identities of nodes.

The following summarizes the operation of our proposal. For example, if a node detects a traffic congestion, it must notify its neighbours about it so that such communication provides them an augmented reality of what is happening on the road. In this way,

other users outside the congestion zone can make decisions in time to avoid possible accidents and traffic jams, for example by finding an alternative route. Neighbouring nodes that can check such event information are responsible for determining the authenticity of the messages, reporting detected forgeries if they exist. Since then fake nodes will not be authenticated by other nodes and will be unable to get any profit from the network. The whole process is automatic and transparent to the network user so that there is a module responsible for detecting false or altered information and informing the network about it. To do it with security, all forwarding messages must be signed by the origin in order to enable nodes to determine which is the node that presents a bad behaviour.

When developing the proposed mechanism, an important problem that was taken into account to make it possible that the system works properly was the requirement that users cooperate by relaying packets to their neighbouring nodes. Therefore, the possibility that legitimate nodes act passively only receiving information from the network is prevented with the proposal. Such attackers would try to benefit from getting information from the network but without participating in the relay to its neighbour nodes. This would damage the network passively, by degrading its performance and threatening the connectivity. Consequently, a specific module to determine whether nodes cooperate in the network is included in the proposal. There exists another possible attack consisting in relaying packets to overload the network. In this case, legitimate nodes would cooperate in the attack by contributing to disseminate information that is useless or repeated. Tools to avoid this specific type of attacks are also detailed below.

B.3.1. Cryptographic Preliminaries

Important research issues of VANETs are the cryptographic needs of these networks, such as authentication, data integrity, privacy and confidentiality. In order to meet all these requirements, the uses of various known mechanisms such as public-key digital signatures and pseudonyms have been included in the proposed scheme.

During the network construction, each user must get a public/private key pair in a decentralized way. In order to achieve this goal, each new node will perform a key exchange with one or more reliable nodes in the network. Additionally, a pseudonym will be given

to each new node so that it will be associated with its cooperative or selfish behaviour but without revealing its identity. This alias will be created by an automatic generator from its public key, what prevents both the existence of two identical pseudonyms and the possibility of generating a false pseudonym to masquerade as another vehicle.

Furthermore, each network node has a key store that contains other nodes' public keys signed by reliable users of the network. When two nodes meet and want to communicate with each other, their public keys are exchanged. Each public key will be looked up at the key store, and if there is no coincidence, both nodes exchange their stores. Thus, any node will try to find a common path in the resulting web of trust [73]. Otherwise nodes are not authenticated and so they can not trust one another. It is possible that the probability of coincidence at the beginning of the network is small, so lower security levels have to be defined then. When the network reaches a sufficient size, taking into account the small world experiment [95], these levels may be raised. The experiments associated with the so-called "six degrees of separation"[111] are based on the idea that if a person is one step away from each person it knows and two steps away from each person who is known by one of the people it knows, then everyone is at most six steps away from any other person on Earth. This idea is used in our work in an important aspect because according to the principle of "six degrees of separation", the probability to find a common chain between two key stores is high, so the probability find a match in the key stores of two nodes who do not know each other. Besides, mobility, one of the important characteristics of VANETs, permits to efficiently reduce uncertainty and to speed up trust convergence.

B.3.2. Detecting Misbehaviour

The basic idea of this work is that VANETs will allow detecting traffic jams and other events on the road through the automatic exchange among nodes of reports and warning messages about them. This will be done thanks to the information provided by GPS because with GPS software it is possible to know the speed at which nodes are moving and the maximum speed allowed in each lane of the road. Given this information, if a vehicle is travelling at a speed below the minimum, it is probably due to that there is traffic congestion on that road. In this case, a packet will be automatically generated to warn users about the traffic

problem. Wireless network technologies allow devices to move freely. However, this mobility affects permanent access to the network. In the present proposal it is not necessary this type of connections, nodes exchange information when they meet so mobility is not a drawback. This design is based on the so-called store-and-forward routing model. In a typical packet forwarding process in VANETs, vehicles meet one another at different times, and packets are opportunistically forwarded. If an intermediate vehicle stores a packet for a longer time and actively sprays the packet to other vehicles, the packet will be more likely to reach a greater number of vehicles.

In this work we consider that a bad behaviour of a vehicle within the self-managed vehicular network can consist on:

- Inserting in the network false packets with spoofed content on the state of the road or inserting many times the same packet to try to launch a DoS attack.
- Not cooperating in relaying packets of its neighbour nodes so that it benefits from the network without cooperating in its operation.

The detection of an attack attempt should be automatic and transparent to the user. Hence, in order to achieve it, the proposed system uses environmental parameters and compares them with parameters of the received packet. Thus, all data packets in our proposal contain at least the following information:

- **GPS coordinates.** The GPS coordinates will help in two ways. On the one hand, combined with the movement direction, they will provide information about the place where an event data packet was originally generated and where the problem is located. On the other hand, they will allow discarding warning events beyond a certain range because in most cases, information generated at a certain location in a VANET is not interesting out of a radius distance. Thus, fo instance, an event data packet can be generated in coordinates (X,Y) and certain range of interest for this packet can be defined within a radius R. In this way, such a packet will not be broadcast when it reaches R. The particular size of the radio R must be fixed by the source node according to the type of road and warning event.

- **Speed.** A parameter that the GPS device uses to detect the fastest or shortest route to the destination is the maximum limit of all speeds in the used via. In this sense, our system can detect whether there is a traffic jam in a specific highway through the speeds at which vehicles move on it. With this information, the GPS device will be able to make calculations to determine if going through the traffic jam will take less time than modifying the route. Otherwise, it may propose a new way to reach the destination in the shortest time possible.

- **Next via.** Information about the next via let us know whether a possible traffic jam is across the entire highway or only in a given lane of the highway.

- **TimeStamp.** The parameter that gives the time at which the packet was sent allows determining whether the received information is new or old. This makes possible to have updated information about the road all the time.

False Information Detection

The next paragraphs explain how the aforementioned information contained in data packets can be used to detect attempts of false information retransmission:

- **GPS coordinates:** If a vehicle A provides information about dense traffic in a road where another vehicle B is driving at an appropriate speed, the vehicle B can report that A is introducing false information. Moreover, if a vehicle is sending an event data packet outside the fixed radius R, this can be reported as a DoS attempt.

- **Speed and Next via:** If a vehicle sends information about a traffic jam in certain coordinates where another vehicle is circulating at a high speed, this should be reported as a fraud. Indeed, specific cases such as when the next via in its route is nearby, should be dealt as an exception for detecting a traffic jam warning.

- **TimeStamp:** If a vehicle is transmitting information with an expired TimeStamp, this is considered as a DoS attempt.

Non-Cooperation Detection

A vital aspect for the operation of the network is to provide real-time and trustful information to others nodes. This is achieved through node cooperation in relaying packets to their neighbouring nodes. In this work we identify cooperation of nodes through timestamped ACKnowledgment (ACK) packets.

In particular, if a node A receives or produces some event data packet, before providing it to another node B, it asks B about B's cooperation in the network. The node B answers A by providing it with the last ACK it has received. If the timestamp included in such ACK exceeds the threshold T defined by the protocol depending on the network size, the node A does not send the packet to B. Thus, nodes are motivated to cooperate in order to upgrade their ACKs.

In order to get the ACK after a retransmission and to avoid selfish behaviour, the system follows the process detailed in Figure B.3 where node A, who wants to send the data packet M to B, splits it into two parts (M_1, M_2) in order to ensure that node A receives at least one ACK as proof that it is cooperating before B receives the complete packet. When B receives the first part M_1 of the packet encrypted with its public-key $E_B(M_1)$, it sends an acknowledgment signed with its private key, $D_B(ACK_1)$. Then, A sends to B the second encrypted part of the packet $E_B(M_2)$ so that B can recover the full content of the data packet. Finally, B sends a second signed acknowledgment $D_B(ACK_2)$ to node A. It is possible that after receiving the first acknowledgment, A does not send M_2. Also, B might not send the second acknowledgment. In any of these cases, the fraud must be reported through the reputation mechanism based on reputation list described below.

B.3.3. Malicious Node Isolation

Each network node maintains two reputation lists, one providing an overview of all malicious network nodes and the other providing the vision about own experience with malicious nodes during its life in the network. The purpose of these lists is to use them to exclude misbehaving nodes from the network, so that nodes are motivated to contribute to the network operation in order not to be isolated from it.

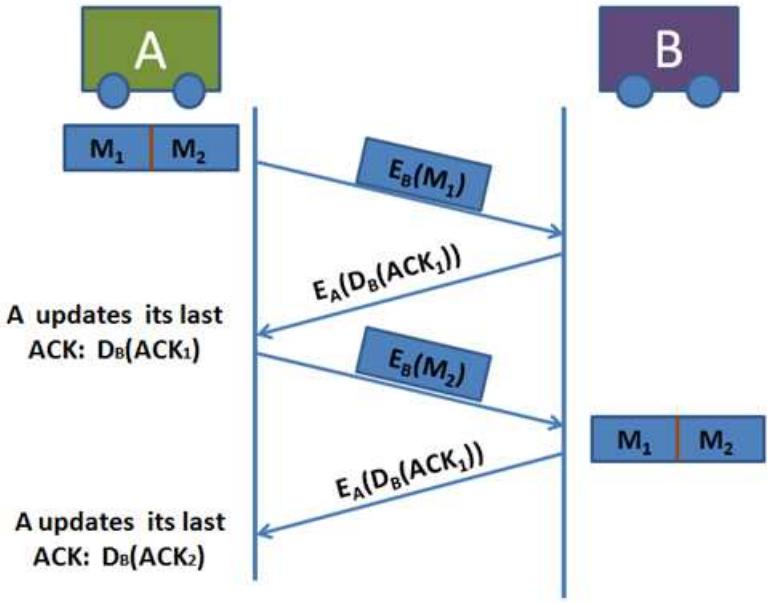


Figura B.3: Sending packets and receipt confirmations.

Each vehicle stores a list called General Reputation List (GRL) containing revoked pseudonyms corresponding to vehicles that have bad behaviour against the operation of the network. In the absence of a CA, the GRL must be updated through the exchange of the GRLs among neighbouring nodes. Such an update is done each time two cooperative and authenticated neighbour nodes starts a communication. If a vehicle A has an event data packet and meets another node B that is in its GRL, A does not provide B with such a packet. Thanks to this procedure, nodes will not have selfish behaviour within the network. Moreover, if a node A receives a packet from a node who is in A's GRL, A will discard B's packet so that the misbehaving node B will not be able to continue attacking the network. In order to update this list, it is important that the process is efficient and based on a fast search algorithm. Table 1 shows four possible fields in this list. Each record in this list will contain the misbehaving vehicles' pseudonyms. The timestamp of a bad behaviour is used to keep the list updated by deleting old records. Another field including the signature of the node who presented the complaint is also stored. Furthermore, the GPS coordinates field corresponding to the misbehaviour location is included so that they can be used to simplify

the GRL if it grows too much.

Tabla B.1: Fields of the GRL.

Selfish node's pseudonym	Misbehaviour TimeStamp	Complainant's Signature	GPS Coordinates (X,Y)

As previously discussed, a vital aspect for the operation of the network is that nodes cooperate in relaying packets of their neighbouring nodes. To meet this need, we propose the use of the so-called Individual Reputation List (IRL). It allows the node to store information about cooperation got from the different nodes it meets during its life on the network. This list is maintained by each node to store information about its direct own experience with other nodes of the network, so it is totally reliable for the node. Hence, thanks to the IRL the node can make clear decisions on whether to cooperate or not with other nodes. The information in IRL is directly added to the GRL during GRL's exchange.

From the stored GRL, each node can compute a misbehaviour rating r_j for each other node j corresponding to the number of stored complaints from different complainants against j. Thus, the higher misbehaviour rating the lesser probability to send/accept data packets to/from j. In particular, such a probability $prob_j = 0$ if $j \in$ IRL. Otherwise $prob_j = 1/r_j, \forall r_j \geq 1$, and $prob_j = 1$ if $r_j = 0$.

The IRL is updated whenever the node detects misbehaviour of a vehicle that tries either to forge a message that does not correspond to its real environment information or not to cooperate in the retransmission acknowledgment process explained above.

During the aforementioned exchange of packets and acknowledgments, if some node decides not to relay all necessary packets, it is introduced in the IRL of the other node. This would happen if for example A does not send the second part of the packet after B has sent the first acknowledgment, or if B does not send some of the corresponding acknowledgments. Figure B.4 shows the flowchart corresponding to the procedure when a node A sends a data packet to a node B while Figure B.5 shows the flowchart corresponding to the process when a node B receives a data packet from a node A.

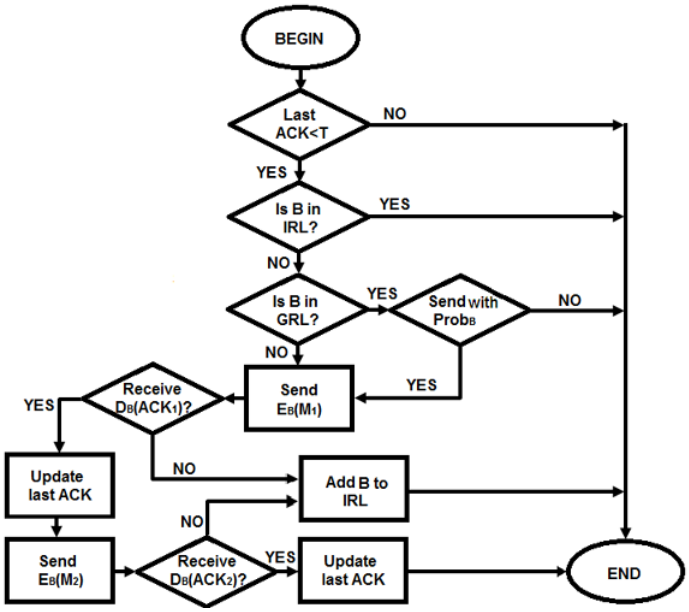


Figura B.4: Sender A's Flowchart.

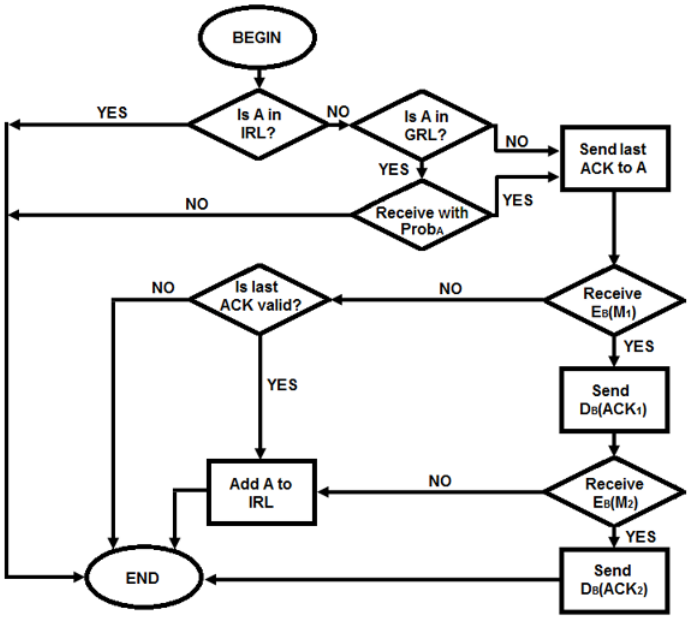


Figura B.5: Receiver B's Flowchart.

B.3.4. Generation of Road Events

In the absence of a central authority, certificate revocation must be done through cooperation. As explained above, when a node detects an incident, it must generate a

detection message in order to allow that other vehicles in its environment sign it if they agree with the information. This message M will be signed with the private key of the node i that generated the packet PrKi(M). If the vehicle i is an attacker, it will try to send a message containing false information (see Figure B.6A). Nodes that are within its cooperative group, detect that it is an attack by checking the information and include the node i in the IRL. Hence, one of the nodes that detected the attack will start forming a new cooperative group with all nodes in its range excluding the attacker (see Figure B.6B). The new leader sends a message of complaint containing the false information and the signed message with the private key of the attacker node. Nodes check this information and if they also detect that it is an attack, they send a signed complaint, in response to agree with the fact that node i is an attacker. Finally, the node that generated the complaint packet adds all received complaints and sends the information to all nodes that are in scope. At this time and following the process of the previous section, the nodes include the node i in the GRL, they delete the node i of the IRL to optimize space and they revoke node i, deleting its key from their key storage. During their life on the network, nodes will exchange the information of events that are stored on their database. At this meeting, they also updated their repositories and exchange the GRLs among their neighbours so that they isolate malicious nodes from the network. In this case, it is impossible for an attacker to revoke a honest node, since it is impossible to have the honest node's private key to sign the information.

It is impossible to ensure that during communications, there are a sufficient number of nodes to revoke another node on the network. Therefore, when the number of complaints is not enough but the nodes detect an attack, they store the attacker node in the IRL (see Figure 6C). This prevents nodes to perform the isolation attack on the same nodes. When a node tries to communicate with another node that is in the IRL, the honest node will reject the communication. In an exchange of lists, the node, which is in the IRL and in the neighbour node GRL, is removed from the IRL and revoked.

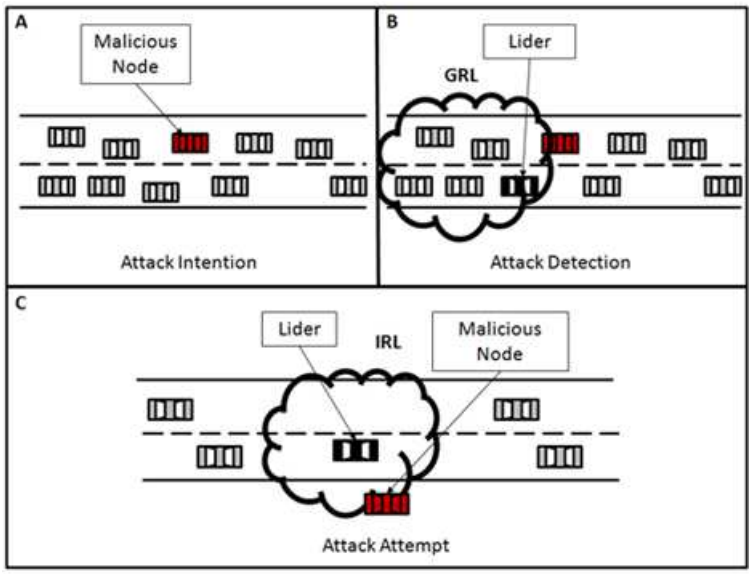


Figura B.6: Attack and Detection

B.3.5. Analysis of the Proposal

Both the feasibility and effectiveness of the proposal have been tested through several simulations. In particular, we used NS-2 and SUMO taking as starting point the simulations analyzed in [32]. We simulated the proposed scheme based on a combination of rewarding ACK-based exchanges and reputation lists in a random environment to see its effects on network and cooperation performance. In order to make a performance analysis of the proposal, several VANET simulations were implemented. This section presents some details and results of these simulations. The aim of our proposal is to determine and isolate from the network all malicious nodes. An interesting analyzed parameter is the time required for all network nodes to know which nodes are malicious in order to isolate them and prevent communications with them. The first simulation consists of a network of 100 nodes that make communications between them in a totally random way. Each simulation was performed 100 times for different percentages of malicious nodes. If a node meets for the first time a malicious node, it includes it in its IRL and GRL. If it meets a malicious node who is already in its IRL, the connection is stopped. Otherwise, if the neighbour j is in the GRL, then according to $Prob_j$, it can either stop the communication or continue with it.

Both in this last case, and if it connects to a node that is not malicious, make an ACK-based exchange of their GRLs. Figure B.7 shows the average time (in min.) required for all the simulated networks with different percentages of malicious nodes until all nodes determine who are the malicious nodes. As we can seen, as the number of malicious nodes increases, the time to detect them decreases. This is because there is a greater probability of encountering a malicious node and therefore the number of complaints on those nodes increases. Therefore, the mechanism works faster to isolate malicious nodes when the number of malicious nodes is greater.

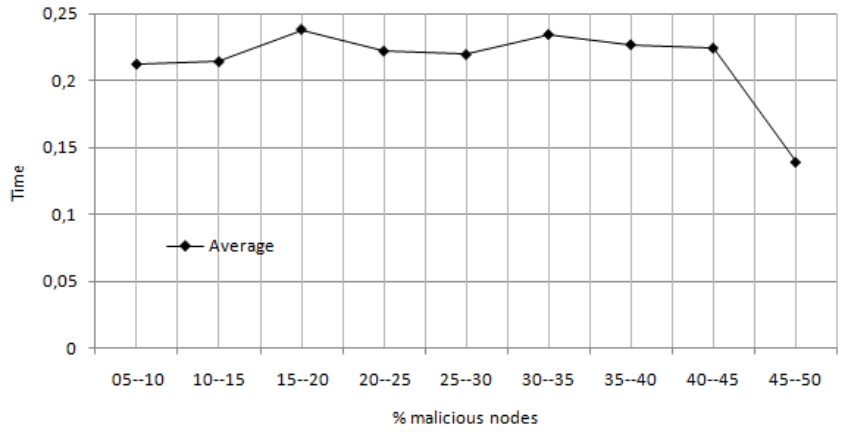


Figura B.7: Average warning time vs. percentage of malicious nodes

Figure B.7 shows that where more time is needed by the method is in networks with 15 to 20 percent of malicious nodes so we took this value and varied the number of nodes in the network from 100 to 1000 nodes to determine how the number of nodes influence in the time needed to isolate malicious nodes from the network. In this case the average warning times (in min.) for 100 simulations performed for each different network sizes is shown in Figure B.8. As we can see, the time to alert all nodes increases with the increase of the network size. However, the results show that it is possible to isolate the malicious nodes, in a reasonable time and independently of the network size because the growth is linear rather than exponential.

Therefore, for all the implemented simulations we can conclude that the proposed cooperative system using reputation and rewarding works properly for the analyzed

VANETs.

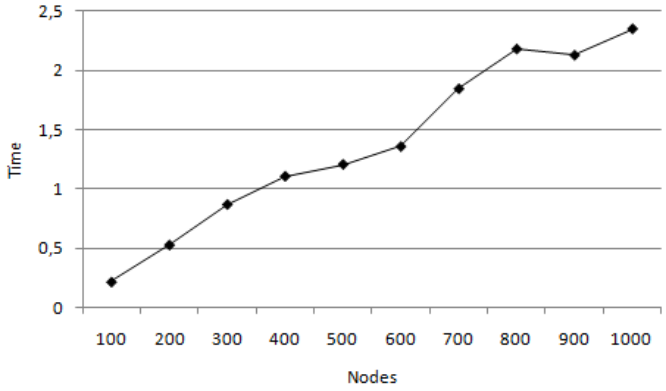


Figura B.8: Average time for warning vs. network size

B.3.6. Flexibility and Robustness

A good detection mechanism for cooperation must have two characteristics: flexibility and robustness. With regard to flexibility, note that a hardware malfunction can make the device sends messages with an incorrect or expired timestamp. Therefore we should not be too strict and allow nodes to recover from that problem. Moreover, it would be unfair to prevent the access of misbehaving nodes to the network forever after only one bad behaviour. In order to solve this problem, any node with a reported misbehaviour has two possibilities:

1. To get a new key pair and a pseudonym from a legitimate node belonging to the network. When a node has been marked as an attacker, it will be isolated from the network and will not receive any traffic information. In case of malfunction, the node can request a new key pair from a legitimate network node. Before the node receives it, it must explain the situation to the legitimate node that will provide the new key pair if it thinks it is appropriate.

2. To wait till records in nodes lists are old enough to be deleted from them. Once the threshold T is reached, the complaint expires and the node is removed from the lists so that the node is able to re-join the network with its credentials.

The robustness of the proposed mechanism ensures that the information that reaches any node is true, what prevents that nodes can impersonate other nodes by sending fake packets on their behalf. To ensure this, each intermediate vehicle must be able to determine whether the information generated by the source node has not been altered. In this case, the source node signs the packets with its private key. Thus, if the information is altered, the received node will be able to detect it. Furthermore, thanks to these detection mechanisms, selfish nodes can be isolated from the network, what ensures that the nodes involved in the network are reliable and so the information they send.

During the security analysis of the proposed protocol, we realized that there exists the possibility of incorrect false information detections. Thus, for instance, there could happen that a vehicle has detected a traffic jam in a road where another vehicle is travelling at an appropriate speed within the same road. This could be a common situation where the left lane works properly but there is a traffic jam in a deceleration lane on the right corresponding for example to an exit to a city. In this case, besides the GPS coordinates and movement direction, the lanes have to be determined.

As discussed in the previous section, cooperative authenticated neighbour nodes exchange their GRLs. This implies that a node can try to attack other nodes by inserting false records in their lists. This is the reason why the criterion for determining whether a node must be isolated or not depends on its misbehaviour rating according to the GRLs. On the one hand, if a high number of nodes agree that a particular node is selfish, then it is probably true and consequently the reported node is isolated. On the other hand, if the GRL grows too much, an additional parameter that could be taken into account is the GPS coordinates (X,Y) so that at least two complaints against the same node should have different coordinates (X,Y) to be considered in the GRL.

An unusual situation appears when a vehicle is stopped on the roadway due to an accident, car malfunction or phone conversation for instance. In any of those situations, the automatic mechanism detects a vehicle at 0 km/h on a road and sends a warning about a traffic jam that does not exist. A drastic option to solve this problem would be to revoke the car, which then should ask for a new key pair after explaining what has happened to its revocation.

Another analyzed problem comes from the use of ACK as a cooperative enforcement mechanism. New nodes that have not participated in any packet retransmission have no ACK to receive packets from the network. The simplest solution would be that the authenticator node gives an ACK to them. Another option would be to wait till the new nodes generate own data packets, so that after sharing them with other vehicles, they get an ACK and are able to participate in the network.

The best practical solutions to all aforementioned special situations will be determined during the practical implementation of the proposal.

B.4. User Recruitment

Security of communications in vehicular ad-hoc networks currently represents an important challenge to be solved because it is expected that these networks in the future will imply a major revolution for the safety and comfort of road transport. In this section, a secure communication system is presented in a spontaneous and self-managed vehicular ad-hoc network, without any infrastructure on the road or vehicles, by using only mobile phones. The operation mode is completely distributed and decentralized. This proposal avoids passive behaviour of users who intend to take advantage of the VANET without cooperating in its operation so we must ensure that only those vehicles that belong to the network and help in its operation, will get benefit from information relayed in it. In this paper we propose using the encrypted exchange of data as method to strengthen cooperation in VANETs. In particular, we describe a new pseudorandom generator to produce a secret key to encrypt sent information. This procedure prevents that passive nodes that do not cooperate in relaying packets, can get benefit from this information.

In these networks, traffic data size is generally very large. For this reason, we propose a symmetric encryption that uses a pseudorandom number generator for a stream cipher [42]. A PseudoRandom Number Generator (PRNG) is an algorithm for producing a sequence of numbers that approximates the properties of random numbers. In order to be considered useful for cryptographic purposes, the resulting sequence must fulfill at least three properties: large period, pseudorandomness and high linear complexity. All these properties

are here analyzed for the proposed PRNG.

B.4.1. Proposed Generator

This section describes a new PRNG proposed that can be used both for symmetric encryption through stream cipher and for challenge-response authentication based on symmetric cryptography in VANETs.

A PRNG can be seen as a deterministic function whose output is computed from the previous output. It is initialized with a randomly chosen seed. The strength of the PRNG depends on the period and the probability distribution of the output sequences. Many PRNGs are based on LFSRs because this basic generator can be applied efficiently in stream ciphers. In this paper, we define a new LFSR-based PRNG.

An LFSR is defined by a feedback function and a string of binary cells that share the same clock signal. To produce a bit, the register contents are shifted one position, extracting as output the most significant bit of the register in the previous state. The feedback function allows computing every new bit from some bits of the register, so that this new bit is the new least significant bit of the new state. The feedback function of an LFSR is basically an XOR operation on some contents of the cells of the state, given by the feedback polynomial over $GF(2)$ denoted $C(x) = 1 + c_1x + c_2x^2 + \ldots + c_Lx^L$, whose degree L is the length of the register. The period of sequences produced with an LFSR of length L is less than or equal to $2^L - 1$, value that is reached when its feedback polynomial $C(x)$ is primitive, in which case the produced sequences have optimum statistical properties. The LFSR is commonly used as pseudorandom generator in cryptography due to the good characteristics of the produced sequences and also because its hardware implementation is efficient and its computational requirements are simple. However, LFSRs have significant drawbacks that must be solved in order to be used safely [35]. The worst problem comes from its linearity as the initial state or seed of the LFSR can be easily determined with a simple system of linear equations by using the polynomial function $C(x)$ and a $2L$-bit output keystream. Thus, in order to use an LFSR to build a PRNG, the linearity problem must be solved, what is usually performed with methods such as the so-called non-linear filtering or the non-linear combination of several LFSRs.

Our proposal is based on an LFSR mainly because it is an ideal system for both energy and computational constrained environments. The design is based on an LFSR with primitive feedback polynomial to achieve maximum period and to fulfil the pseudorandomness of the generated sequences. In particular, in the proposal the number of non zero coefficients of the LFSR feedback polynomial $C(x)$ is the smallest possible integer greater than $0.1 \cdot L$, in order to avoid correlation attacks and to ensure efficiency.

The proposed generator consists of two main building blocks: an LFSR and a filter function. The order of the nonlinear filter function has been chosen to be the greatest prime number p less than or equal to the number $L/2$, in order to ensure a large linear complexity.

The nonlinear filter function includes a linear term corresponding to the stage indicated by the function order. The number of terms in each order $i = 2, 3, ..., p$ is given by the integer part of L/i. These terms are obtained by multiplying successive disjoint stages to achieve pseudorandom and confusion.

Below we specify the concrete details of our design generally sketched in Fig. B.9 The LFSR used in the exemplification of the proposed design is of length L=20. Its contents are denoted by $s_j, s_{j+1}, ..., s_{j+19}$. In particular, the proposed feedback polynomial $C(x)$ of the LFSR is a primitive polynomial of degree 20 defined as $C(x) = 1 + x^3 + x^{20}$ so that the update function is $s_{j+20} = s_{j+17} + s_j$. The contents of the 20-bit LFSR represents the state of the cipher and input of the nonlinear filter function f. In particular, from such a state, 20 variables are taken as input to the boolean function $f(x)$ of algebraic degree 7. Such a filter function has been chosen to be balanced, first-order correlation-immune, and with high nonlinearity.

The filter function is defined as

$$f(x) = x_{i+7} + \sum_{i=1}^{10} x_{2i-1} \cdot x_{2i} + \sum_{i=1}^{6} x_{3i-2} \cdot x_{3i-1} \cdot x_{3i} + \cdots + \sum_{i=1}^{2} x_{7i-6} \cdot x_{7i-5} \cdot x_{7i-4} \cdot x_{7i-3} \cdot x_{7i-2} \cdot x_{7i-1} \cdot x_{7i} \tag{B.7}$$

where the variables $x_0, x_1, ..., x_{19}$ correspond to the tap positions s_j, s_{j+1}, ..., s_{j+19}, respectively. The design of the cipher, shown in Fig.B.9, has been chosen to be as simple as

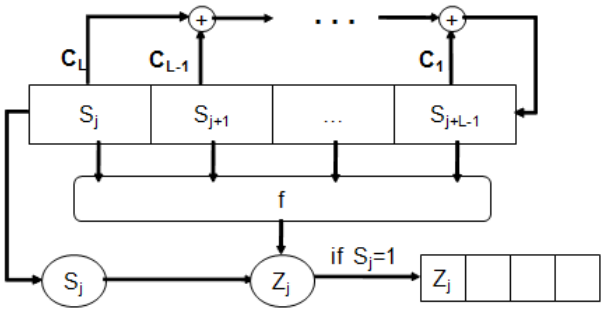


Figura B.9: General Description of the PRNG

possible for a hardware implementation. Since the LFSR in the cipher is of length 20 and its polynomial is primitive we know that the period of the LFSR keystream is $2^{20} - 1$.

B.4.2. Analysis of the Generator

The most important issue of the analysis of any PRNG for stream ciphers is that any possible attacker must not be able to find any regularities in the output stream. If this were the case, a prediction attack might be launched to predict additional bits of the output stream. For this reason, it is required that the output stream is indistinguishable from a random sequence. This concept is formalized through the Golomb's randomness postulates [66], which are conditions that a sequence should fulfil in order to appear random. A binary sequence that satisfies Golomb's postulates is called a Pseudo-Noise (PN) sequence. Each postulate has an immediate translation into some test of randomness. In order to prove pseudorandomness of our generator, we have generated with it a large number of sequences and subject them to a battery of statistical tests [112]. Since most sequences pass most tests, the confidence in the pseudorandomness of the sequences is large and so is the confidence in the generator. In particular, the generator has passed the following tests: Frequency Test, Serial Test, Poker Test, Runs Test and Autocorrelation Test.

The proposed PRNG has been implemented in software to check the pseudorandomness of the output sequences. We generated 3.9 Gb of data with our PRNG in order to check its statistical properties.

Our experiments involved $2^{20} - 1$ bits produced with our generator for every pos-

Test	Frequency	Serial	Poker	Runs	Autocorrelation
% of Success	100	100	99,01	98,55	98,68

Figura B.10: Results of Statistical Tests

sible seed of the LFSR. The results of all aforementioned statistical tests for a significance $\alpha = 0.05$, shown in Fig.B.10, led to the conclusion that our proposed PRNG passes all the studied tests. In particular, the frequency test, which is based on the proportion of zeroes and ones, checks the closeness of the proportion of ones to 0.5. In this case we obtained 100 % of positive results over all possible inputs. On the other hand, the serial test, whose focus is to determine whether the frequency of all possible 2^m m-bit overlapping patterns across the sequence is the same, 100 % of all possible outputs pass the serial test. For the poker test we divide the sequence into subsequences of a certain length, and then check whether these sequences appear the same number of times. It results in 99,01 % of positive results with our generator over all possible inputs. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test produced 98,55 % of positive results for the sequences produced with the proposed generator over all possible inputs. Finally, in the autocorrelation test, the purpose is to check for correlations between the sequence and its shifted versions. It produces 98,68 % positive results with the proposed generator over all possible inputs.

Ten thousand random and different seeds have been used to initialize the PRNG and the correlation between the obtained sequences has been computed concluding that two simultaneous identical sequences do not appear.

In order to check the unpredictability property, the serial correlation test has been implemented. Such a test measures the extent to which each m-bit output depends upon the previous m-bit output. For our sequences, this value is obtained close to zero so we conclude the fulfilment of the property.

Finally, since a PRNG is a finite state machine with at most 2^L states, an output sequence must become cyclic after at most 2^L output bits. As a consequence, the more significant bits of the sequence can be modelled as a function of the less significant bits by a

suitable recurrence relation. In this work we computed the period for the analyzed example and obtained always values around $2^{20} - 1$ for every possible seed.

On the other hand, the linear complexity of any sequence is the length of the smallest LFSR that generates the sequence. Thus, high linear complexity is a necessary requirement for all sequences generated by a PRNG. There exists an efficient algorithm by Berlekamp and Massey [93] that constructs the shortest linear recurrence describing the sequence. We computed the linear complexity of the produced sequences through the Berlekamp-Massey Algorithm over $2^{20} - 1$ bits of the keystream, confirming that it is always maximum and equal to half the number of analyzed bits.

We now briefly consider some general attacks on stream ciphers in order to investigate to what extent they can be applied against the proposed generator. Indeed, resistance against known cryptanalytic attacks is the most important basis for the design of a new encryption algorithm because there should be no faster successful attack than the exhaustive key search. Due to the good statistical properties of the PN-sequences produced by the basic LFSR, and because the function f is first-order correlation-immune, the correlations between the output of the generator and the bits of the LFSR are so small that they can not be exploited for correlation attacks. In addition, a filtering based on a function $f(x)$ of degree 7 is not vulnerable to algebraic attacks as the algebraic degrees of the output bits when expressed as a function of LFSR-bit are large in general, and varying in time so this defeats algebraic attacks. In addition, the cost of time/memory/data tradeoff attacks on stream ciphers is $O(2^{L/2})$, where L is the number of inner states of the stream cipher. To comply with the margins set by this attack, L =20 has been chosen. The sampling resistance of $f(x)$ is reasonable because this function does not become linear in the remaining variables by fixing less than half of its 20 variables.

According to the aforementioned analysis, most sequences produced by the proposed PRNG pass most statistical analyses, have short hardware requirements and are resistant to known attacks, what confirms the validity of the proposal and the hypothesis that the proposed nonlinear filtering and decimation solve the linearity problem of LFSR-based generators.

Appendix C

Aggregation

Vehicular Ad-hoc NETworks (VANETs) are usually defined as wireless networks formed among vehicles and roadside infrastructure, which are used to provide drivers with information to increase safety, efficiency and comfort in road travel. In this type of networks, warning messages affect decisions taken by drivers so that any false disseminated information could lead to higher transportation times, fuel consumption, environmental contamination and impact of road constructions, and, in the worst-case scenario, more traffic accidents.

As prospective intelligent transport system technology, VANETs have become a very hot topic in the research on networks. In the near future, this type of networks will allow the reduction of the number of deaths due to car accidents, and the provision of real-time information on traffic and on roads. For example, drivers will be able to exchange information with their neighbours and with the road so that they can receive warnings about potentially dangerous events such as accidents, obstacles on the road, etc. Another practical application of VANETs is, for instance, the ability to find free car parking spaces.

Nowadays, several GPS applications offer information of the traffic on a chosen route and compute alternative routes based on feedbacks from local road authorities, police departments and systems that track traffic flow. However, in most cases the information given to the driver is not real-time because it does not reflect the events that have just produced, and/or implies the lack of user privacy. Besides, most of that software, like Google traffic application, requires a 3G connection, what represents an additional cost for the

users. Thus, our motivation to study the secure, efficient and self-organized deployment of VANETs to assist drivers instead of those GPS applications is clear.

A classical VANET is composed of two different types of nodes: On-Board Units (OBUs) that are wireless devices installed on vehicles, and Road-Side Units (RSUs) that form the network infrastructure on the road. While in other resource-constrained wireless networks such as sensor networks, data aggregation is used mainly to save energy, in VANETs data aggregation can be used both to ensure that the transmitted information is reliable and to minimize the number of repeated event warnings.

To overcome false content generation problems, a new data aggregation protocol is here proposed. In particular, we combine the ideas of reactive groups and data aggregation with a probabilistic verification scheme to check the authentication of warning messages quickly and reliably in self-organized and decentralized VANETs.

C.1. State of the Art

Regarding the protection of VANET communications, in the literature we can find several papers proposing the use of asymmetric cryptography in VANETs so that thanks to the use of digital signatures, both the source and the integrity of messages can be verified [64]. Other works propose the use of symmetric encryption to provide location privacy [139]. We can also find proposals based on the use of pseudonyms to protect user identities [39]. However, none of those mechanisms protect the system against malicious attacks such as false content packet generation, which is one of the targets of the present thesis. A legitimate but adversary node could try to inject false information that does not correspond to what it is really detecting. For example, a driver who wants to reach its destination as soon as possible might try to disseminate information about a false congestion on a road in its route in order to decrease the number of vehicles on it. In order to face this problem, the system described in [46] uses a mechanism based on threshold signatures, which prevents internal attackers from attempting to send fake messages. Three privacy-preserving variants of the system are there described to provide message trustworthiness and vehicle unlinkability under different road conditions. However, such a proposal requires

the participation of a trusted governmental authority, which is not available in fully distributed and decentralized networks like the ones here analyzed. Other works also propose data aggregation to address this problem but under different conditions. For instance, [114] discusses the relationship between security and data aggregation in wireless sensor networks.

The topic of this work is about the need to act when false information is sent in VANETs. Thus, we are talking about trust management. Related to this topic, the survey [148] discusses the challenges, identifies some desired properties towards effective trust management and concludes the lack of effectiveness of the existing models. On the other hand, with respect to malicious attack identification, a work that discusses this topic is [65], where each node compares the received data with the stored information, as we assume here, but in that paper it is assumed that each vehicle has the global knowledge of the network, condition that is not assumed here.

With respect to data authentication in VANETs, [147] introduces a novel message authentication scheme that makes the RSU responsible both for verifying the authenticity of messages sent from vehicles, and for notifying the results back to the vehicles. On the contrary, the model proposed here does not require any RSU. The authors of [47] propose that not only fresh data are considered for warning, but also aggregated data histories are maintained for disseminating knowledge among vehicles, what is an approach different from the one used here. Another proposal can be found in [52], where the aggregation of multiple messages describing the same event and the use of revocation messages allowing vehicles to report false information are proposed. However, such a mechanism has an important weakness because real messages can be also revoked. In [118] the proposed solution is based on the use of a tamper-proof device and consists in asking an aggregator vehicle about a random aggregated record. The main disadvantage of this method is the dependency on a tamper-proof device since an attacker could easily skip this service in order to compose malicious aggregated data. [119] proposes another mechanism to provide security through aggregation in a scheme where streets are divided into fixed size segments corresponding to Wi-Fi signal coverage. The authors of [144] also outline an aggregation scheme that combines all known information on each fixed-length road segment to one average value. However, both aggregation criteria use a fixed segmentation of the road, what has been

shown that does not work properly with a high number of vehicles in large areas, like for example in big traffic jams covering kilometers.

Recently, the authors of [49] proposed the notion of data-centric trust for event validation, but their scheme produces a high dissemination delay. Another recent work closer to the present topic is [90], which proposes an algorithm to choose one of multiple aggregates for the same area based on a probabilistic approximation to underlying data. The difference with our proposal is clear because we propose a probabilistic approach applied on the verification of aggregated data, and not on the aggregation phase. Besides, they use aggregation to combine observations concerning large areas into one single value, instead of several aggregated packets produced by different reactive groups, which is one of the bases of the proposal here described.

Finally, regarding the formation of groups of vehicles, which is also a topic discussed in this work, there are many papers with different proposals. So, for instance, [74] presents a protocol for relaying information under the assumption that vehicles form groups on a highway, there called clusters, and some details about speed and traffic information are exchanged within nodes in the same cluster. In their proposal, aggregated information contains also the relative positions of all cars to a cluster head and an average speed, what is not necessary in the scheme here described. Their proposal also reduces the amount of data transmitted about a cluster of cars, but it does not include any mechanism for merging aggregates.

C.2. Preliminaries

A natural approach to address the data authentication problem implies providing nodes of a mechanism to store and process received information, including data about type of warning, road where it was created, traffic direction, and source node that generated the packet, among others. When a warning message is received by a node, the device has basically two options: either to alert the driver of the danger even if the information is not true, or do not alert the driver and wait to be able to compare the received data in order to verify the contents of the packet accuracy, although this delay might cause an accident.

The first option might affect the driver's decision and result in a waste of its time and/or money if the information is not true. Furthermore, it can increase distrust about other messages from the network. Thus, the recommended option in VANETs is the second one, even though it implies comparing received data with other information packets received before and related to the same event but provided by different vehicles, what could cause a considerable delay until receiving a sufficient number of packets with the same content from different sources. Apart from such a delay, this option requires that the vehicles have a large storage space as well as a fast mechanism to compare different records. Therefore, the implementation of such an option must take into account that the waiting time should be short enough to warn the driver in time to avoid the problem, and large enough to ensure that the content of the information is true.

In a basic model, each vehicle detecting an event, signs a warning message and broadcasts it, what means a considerable network overhead. Nodes that receive the signed packet, have to verify its signature and compare the content of the message with other related messages previously received, what also causes a major delay (see Fig. C.1). To solve these problems, a simple model that combines the signatures generated by different vehicles to alert about the same danger can be used. However, the direct combination of signatures in a single packet would increase the packet size as the number of vehicles confirming the information increases. This would imply an overload of the channel too. Furthermore, the receiver in such a scheme must verify all the signatures, which also means another delay that would equal or even exceed the time required by the basic model.

In order to try to solve all the aforementioned problems a priori, we propose a new combined model based on two main ingredients. On the one hand, it uses reactive groups to provide trustful information through the limited combination of signatures in an aggregated packet. On the other hand, to solve the signature verification delay, we propose a probabilistic scheme that defines the verification of only a few signatures chosen at random among the signatures included in the packet. These two security mechanisms form the basis of the new model and will be fully detailed below.

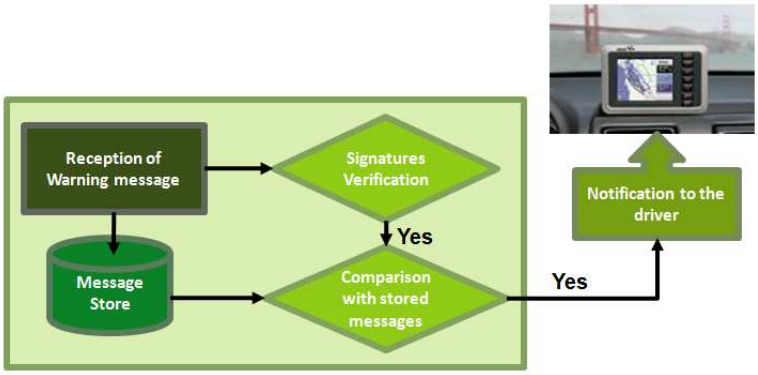


Figura C.1: Basic Model

C.3. Data Aggregation Based on Fuzzy Logic

C.3.1. Fuzzy Logic System

Aggregation schemes are usually based on three sequential phases. The first phase is decision-making, when the system must allow deciding whether two pieces of information are similar enough to add them or not. The second phase is data aggregation. Once the system has concluded that the received packets must be added, it has to apply an aggregation method for combining them. The third and last step is the aggregated data delivery. After the two previous steps, the system must allow the dissemination of the aggregated data.

In the literature we can find several proposals for the two last stages. However, schemes for making decision are normally based on predefined clusters, trees or fixed structures, which are not suitable for their use in VANETs. Thus, a new aggregation scheme is here proposed where decisions regarding whether two received pieces of information are similar enough to be added are based on the content of such packets. Such decisions take into account possible inaccuracies or approximations regarding space and time data to define an event. Therefore, our proposal for fuzzy logic is focused especially on the description of the decision criteria of the first phase of the aggregation scheme.

In the proposed scheme information is aggregated based mainly on two dimensions: space and time. On the one hand, the approximate location of an event in a road or a map is a fundamental parameter. On the other hand, a time interval in which the persistence of an

announced event must be taken into account. When applying a fuzzy approach on such parameters the best decision on aggregation is made because it allows a flexible reasoning that takes into account all possible values in packets announcing the same event. This enables a dynamic approach when considering clusters for aggregation. Before the implementation of the system, and depending on the particular application, it is possible to identify other possible parameters apart from spatial and temporal location, with possible influence on the decision of aggregation, such as distance between agents or speeds. The input values corresponding to all influential parameters are real numbers, which the system has to assign to adjectives. The next step is to formulate rules that express the combination of the chosen influential parameters. In particular, according to such rules, the degrees of the influential parameters' adjectives are combined using fuzzy Boolean operators such as AND, OR and NOT, defined as the minimum, maximum, and complement, respectively. The output of the application of every rule is defined as two possible values: YES and NO. This fuzzy decision making process allows combining all possible influential parameters in order to conclude whether two received warnings refer to the same event because the complexity of the original real values of such parameters are hidden behind the mapping to adjectives.

C.3.2. Control Rules

As an example of fuzzy decision making based on rules we propose to choose and combine the influential parameters of spatial and temporal location denoted Space-Distance (SD) and Time-Distance (TD). Fig. C.2 exemplifies this process for both variables representing the influence of both parameters considering as x-coordinate respectively SD in meters or TD in minutes, and the y-coordinate is the probability corresponding to the adjectives LOW, MEDIUM and HIGH. Each of these adjectives is described by a membership function that maps the real valued input value of the corresponding influence factor to a membership degree corresponding to the adjective described by the function. In the depicted example, according to the fact that a typical error in normal GPS is about 23 meters of ambiguity, an input SD of less than 3 meters is fuzzified as being LOW with a degree of 1. For example, the output of the function for a SD of about 9 meters is classified at the same time as LOW and as MEDIUM with a degree of 0.5. From 27 meters on, SD is considered HIGH with

probability 1. The same is considered for TD in minutes.

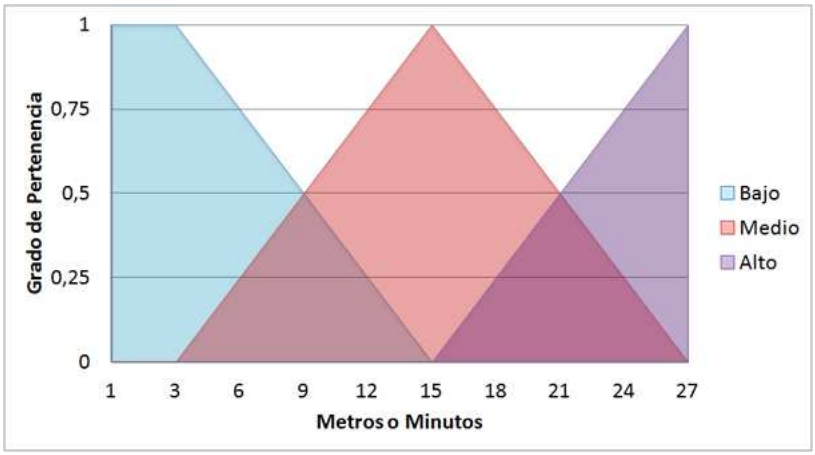


Figura C.2: Fuzzification Function of SD and TD

The next step after fuzzification is to formulate particular rules that express the combination of the influences. As an example, a possible simple basic structure of such fuzzy logic rules is as follows:

Algorithm 1 Fuzzy Rules

01: **if** (SD **is** LOW) **OR** ((SD **is** MEDIUM) **AND** (TD **is NOT** HIGH))

02: **then**

03: Aggregation-Decision **is YES**;

04: **else**

05: Aggregation-Decision **is NO**;

06: **endif**

The above rules can be also represented according to Table 1.

There can be more than one rule assigning values to the aggregation decision. In this case, the assignments to the aggregation decision are combined by an implicit AND,

Tabla C.1: Fuzzy Rules

HIGH	NO	NO	NO
MEDIUM	YES	YES	NO
LOW	YES	YES	YES
SD/TD	LOW	MEDIUM	HIGH

so that the corresponding probability of the Aggregation-Decision is given by the minimum between both input probabilities for SD and LD. After all rules have been evaluated, the decision will be either YES or NO depending on which of the two has got the higher assigned probability by the rules.

For example, if the SD is 10 m and so it is fuzzified to LOW with a degree of 0.32 and to MEDIUM with a degree of 0.68, and for the same pair of packets the TD is 20 min and so the TD is fuzzified to MEDIUM with a degree of 0.58 and to HIGH with a degree of 0.42, the Aggregation-Decision would be YES with a degree 0.58 and NO with a degree 0.42, so the final decision is YES and we aggregate both packets because we concluded they refer to the same event.

Data received by an agent must be stored in a local database with the objective not only of aggregation but also because the channel could get overload, restricting the amount of data that can be sent, and in this case, the agent is allowed to send only a subset of its database. In order to solve the problem of selection, we might use an approach also based on fuzzy logic to take into account relevant parameters for spreading the more adequate and accurate information that is currently available in the database. Then, each agent should carry out a qualification process of the data in its database in order to conclude which are the most relevant data stored that must be sent according to the restrictions of the channel. Several factors, such as severity or antiquity of an event, can play an important role in this selection. As of the aggregation decision, rating relevance of different parameters of a piece of information often depends on the specific application. Relevance ranking provides an order on the parameters to be considered and on data stored so that according to this order, the agent always knows which packages should be sent without collapsing the channel.

C.4. Data Aggregation with Probabilistic Verification

In this section we describe in detail a new data aggregation scheme that contains both a security mechanism based on reactive groups created on demand to ensure a priori that vehicles generate trustworthy information, and a probabilistic verification scheme to detect attack attempts a posteriori in an efficient way, with minimal overhead and delay.

C.4.1. Geographic Zones

Now we introduce the definition of geographic zone, which is a key concept of the group-based mechanism of the proposal.

Due to specific characteristics of vehicular networks like high mobility and frequently changing topology, it is especially difficult to protect data in such networks. Thus, the security mechanisms in this environment should not assume the existence of any stable and centralized infrastructure, but only the existence of mobile functional nodes within the network. In our data aggregation scheme, we consider three different possible situations of nodes:

- Vehicles that find an event on the road and automatically generate a warning message.
- Vehicles that receive a warning packet and can directly confirm that it corresponds to a true event.
- Vehicles that receive packets with the event warning and its respective confirmation, but do not have direct contact with the reported event.

In most cases, information generated at a certain location in a VANET is not interesting out of a radius distance. For example, if an accident happens in a city center, in most cases it has not any sense that the corresponding warning message reaches a neighbour city. Consequently, in this thesis three different geographic zones are defined depending on where warning messages about an event is considered interesting by the receivers. In particular, different parts of the protocol must be run depending on the geographic zone where the receiver node is. As shown in Fig. C.3, three geographic zones are defined with respect to a reported event:

- Danger zone, is the area defined by the innermost distance from the event, so that the event can be directly detected by vehicles in this zone.
- Uncertainty zone, where nodes cannot confirm the information directly, but they have to make decisions quickly because in a short period of time they can be in the danger zone.

- Security zone, where nodes cannot confirm the information directly, but they have enough time to collect evidences about the event in the form of aggregated packets.

The particular size of the radio of these zones is fixed by the source node, according to factors such as the type of road and event.

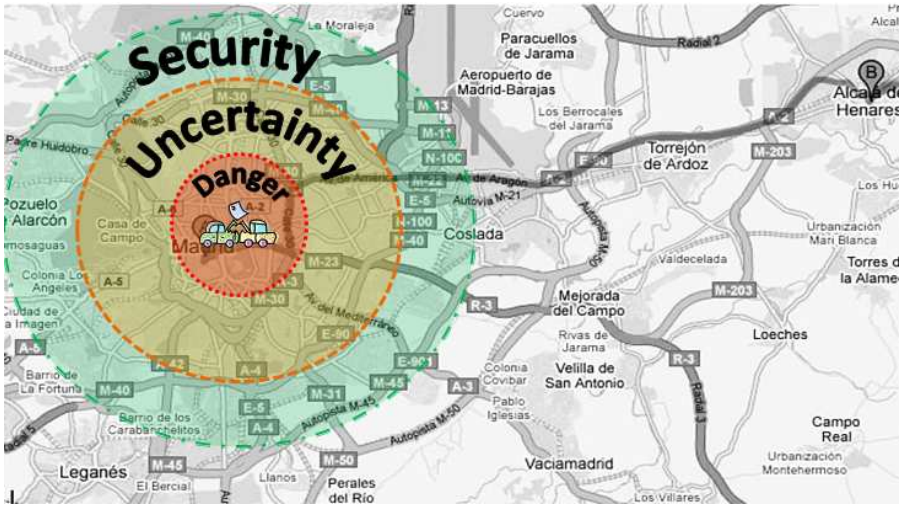


Figura C.3: Geographic Zones

Aggregated and signed warning messages can be only produced by vehicles in the danger zone, while aggregated message verification is only run by vehicles that are unable to verify directly the information reaching them, that is to say, by vehicles out of the danger zone. In particular, when one of these vehicles receives a warning message about an incident that is not under the coverage of its antenna, and wants to confirm the authenticity of the received message, it has to act differently depending on the geographic zone where it is:

- In the uncertainty zone, the decision must be made quickly to allow notifying the driver, who must be sure that the information is true. In this area, if a vehicle receives a signed aggregation packet, the vehicle should use a verification mechanism fast enough to verify all the signatures contained in the packet. However, as aforementioned, on the one hand it is inefficient to verify all the signatures contained in a packet, but on the other hand it is necessary to verify the information before accepting it as valid. In order to fix this problem, only a few signatures are proposed to be verified.

- In the security zone, since this region is quite far from the reported event, the receiving vehicle has more time to collect aggregated messages before it has to make a decision. In this case, the vehicle has to verify the signatures of the received packet, as in the previous case, but vehicles may also perform other verifications to provide a higher level of certainty on the received information. Being in this area, it is possible to receive different warning packets about the same event. Taking into account that these packets are independently generated, the more received and verified warning packets, the greater certainty and accuracy of the provided information. An additional verification that the vehicles in this zone can perform is to recreate the different cells where the received packets could be created from the data they contain, what provides value-added information because all the signatures in every packet should correspond to the same cell. This simple procedure gives an additional way to detect malicious nodes.

C.4.2. Reactive Groups

In order to provide real-time and trustful information about events on the road, since no technical infrastructure is available to coordinate vehicles to act as a group, a reactive establishment of vehicle groups in a self-organized way is here presented. Thus, groups are formed only when they are necessary. This tool prevents any packet to grow indefinitely because it implies a limit in the number of signatures contained in it. Here we propose a mechanism where group formation is not required a priori but when a vehicle detects an event, it automatically tries to form a group with other vehicles within its range in the geographic danger zone centered on the event. Note that otherwise, in dense environments, if all vehicles detecting an event sent the same warning message, the communication overhead in the network would be very high. In this way, the organization of vehicles in reactive groups to aggregate information allows avoiding repeated warnings. The event location defines the center of the danger zone, area that is proposed to be divided into cells to form groups. The leader of each group will be in charge of constructing the signed aggregated warning message.

As aforementioned, groups proposed in this work are reactive and created on de-

mand because they are formed only once an event is detected on the road. In particular, when a vehicle detects a static event, it generates a packet with information about the event such as its geographic coordinates (X,Y,Z), timestamp, traffic direction, etc. This packet is broadcast to all nodes that are in the range of the danger zone. For instance, the radius could be 100 meters, which is roughly the transmission range of a Wi-Fi network. This danger zone is then divided into cells where reactive groups can be created as shown in Fig. C.4. Thus, a cell is a geographical area limited by the maximum number of vehicles that can form a group, while a group is a set of vehicles inside a cell, which produces an aggregated packet. In the maximum case, the number of vehicles in a group corresponds exactly to the maximum number of vehicles that fit into a cell. Since the dimensions both of the cells and of the danger zone are included in the message, all the receiver nodes in the danger zone define the same cells with respect to the event coordinates (X,Y,Z).

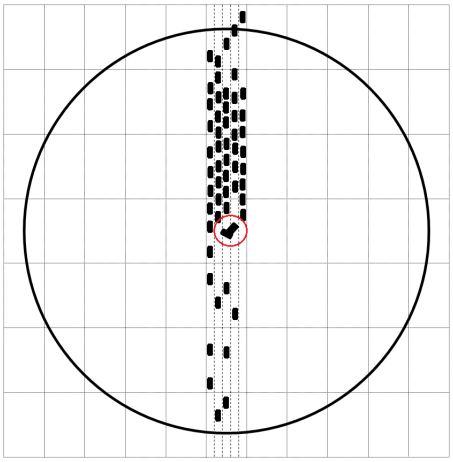


Figura C.4: Cells in the Danger Zone

When a node receives a warning packet P from a node with ID, it runs the following process algorithmically described in Algorithm 1. First, it checks whether its vehicle is within the danger zone of the packet. Then, it checks whether it is able to detect the event, and in that case, it calculates the dimensions of the cell to which it belongs according to the coordinates (X,Y,Z) so that, if it has not received any group formation request, it launches a new one to all nodes within its cell. It is possible that two or more vehicles start the group formation request approximately at the same time, but this problem is easily solved

as all vehicles always choose the request with the oldest timestamp and in the rare event of a timestamp collision, they choose the neighbour closest to the center of the cell. The chosen node becomes the leader of the reactive group and is in charge of generating the corresponding aggregated packet. Once the group is created, the leader signs the warning message and sends it to all members of its group.

When the vehicles of the group receive a signed warning message from the leader, they check whether they can validate the received information by verifying data such as time of the reported event and whether they are in the range to detect it. If they meet the above requirements but they do not detect the reported event, they might consider it as an attack attempt. In this case, the nodes would mark the leader node as malicious according to the scheme proposed in [98] and discard the packet. On the other hand, if they agree with the received information, they sign the message and send it back to the leader. The received signatures provide evidences so that the leader aggregates them to generate an aggregated warning message. This implies that the leader creates a packet where different vehicles from its group alert about the same event, what can be used as proof that the content of the information is true. Finally, the leader broadcasts this aggregated message and all the receivers verify the signatures according to the scheme described below and store the message.

Algorithm 1 Reactive Group Formation

```
01: function Main(Packet P, Node ID)
02:   bool GroupFormationRequest= false;
03:   ////Checks whether it is a new warning message
04:   if (Warning(P)) then
05:     // Gets packet information
06:     double X = P.X;
07:     double Y = P.Y;
08:     double Z = P.Z;
09:     string EventType = P.EventType;
10:     string Direction = P.Direction;
```

```
11:     string Road = P.Road;
12:     double TimeStamp = P.TimeStamp;
13:     //Checks whether its vehicle is within the danger zone
14:     if (InDanger(X,Y,Z,EventType,Direction,Road,TimeStamp)) then
15:       //Checks whether it is able to detect the event
16:       if (DetectEvent(X,Y,Z,EventType,Direction,Road)) then
17:         //Computes the dimensions of its cell
18:         double Cell = CellDimensions();
19:         // If it has not received any group formation request, from
20:               // its cell it launches one
21:         if (!GroupFormationRequest()) then
22:           //Sends a group formation request
23:           SendRequest(X,Y,Z,EventType,Direction,Road,TimeStamp);
24:         end if
25:       else
26:         MarkMalicious(ID); //Identifies attack attempt
27:       end if
28:     else
29:       //Cannot detect the event
30:     end if
31: end Main
```

It might happen that a node is alone in its group. In this case the node generates a packet with a single signature and sends it like an aggregated packet. As the packet has not enough evidence about the truth of the information, it is not taken into account in the uncertainty zone. However, this packet could be used combined with other warning messages about the same event received in the security zone.

C.4.3. Types of Packets

The described process of aggregation of data involves four types of packets:

- Packet type W: warning packet broadcast by a source node who detects an event to all nodes in the danger zone. This packet contains the message M formed by (X,Y,Z) coordinates, type of event, traffic direction, name of road and timestamp, together with node ID and the signature of M using the private key Pr of the node.

- Packet type R: corresponds to the request for group formation after the reception of a packet type W. This packet is sent by a node being proposed as leader of a reactive group to all the vehicles in its cell. In this case the content includes M together with the self-nominated leader's ID, signature of M, coordinates and timestamp in order to allow other nodes to determine whether the sender node is the leader of the group or not. The node sending this packet can know whether another node is in its own group or not thanks to its geographic coordinates, so it waits for a fixed time to receive all the signatures of the nodes belonging to its group.

- Packet type S: contains the signature of each node who agrees with the received packet type R, both with the event warning and with the group information. It contains the message M and the data corresponding to the sender.

- Packet type A: containing all the signatures received in packets type S by the leader of a group from its members in order to provide higher evidence about the existence of a reported event. This packet is broadcast by the leader through the network.

Figure C.5 shows a summary about the contents of each type of packet.

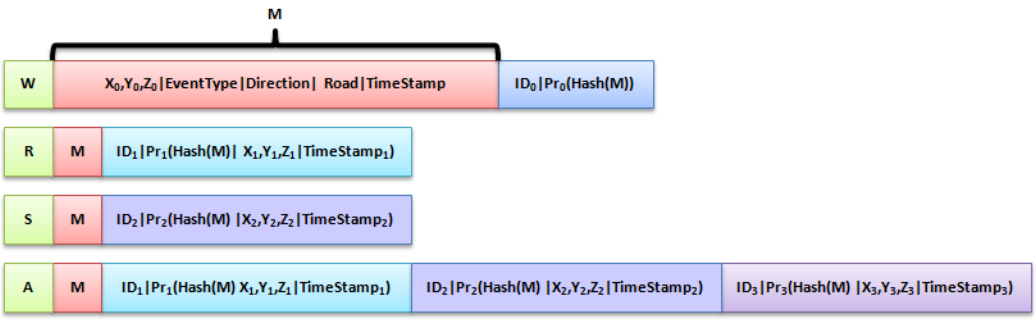


Figura C.5: Types of Packets

C.4.4. Cell Size

The division of the danger zone into cells have been proposed here as a way to avoid channel overhead due to huge packets containing all the signatures of nodes detecting the same event. The aggregated packets produced in each cell should provide enough evidence about the validity of the reported event. Thus, the cell size must be chosen properly in order to address both issues because it is directly related to the packet size and to the number of signatures. In fact, it defines both the usual minimum and maximum number of vehicles inside it, what determines respectively the reliability and size of the corresponding packet type A.

According to [75], a fixed cell size of 16 meters wide and 126 meters long is the optimal cell size for VANETs. However, in a dynamic environment, these parameters should be variable in order to adapt them to the different environmental characteristics. In a VANET, vehicles are affected by different factors that vary while they circulate. Some of them are the number of lanes of the road, their speed, vehicle density supported by each road, etc. Taking into account these changing characteristics, it is necessary to define specific criteria for determining the size of a cell based on this information.

It is impossible for every node to determine exactly how many vehicles exist on the road where it is at each moment, and the distance and speed of such vehicles, but each road has a maximum speed and minimum safety distance that vehicles must maintain, what give an idea of the maximum number of vehicles that should be on it in normal conditions. Based on these parameters we can propose a metric to calculate the size of the cells.

The goal is to find an optimal cell size that minimizes the aggregated message size while maximizing the reliability of the information. In the simulations, we established that the length of each cell is twice the safety distance, value that can be easily calculated from the square of the road speed, while its width is the number of lanes of the road multiplied by the width of each lane, 4 m, what provides the area of the cell:

$$CellArea = (2 * safety\_distance) * (4 * number\_of\_lanes)$$

This expression produces approximately the fixed optimal cell size indicated in [82] for four-lane roads with 80 km/h speed limit.

In this way, the danger zone can be divided into cells so that the center of the danger zone (X,Y,Z) corresponds to the center of the central cell. The other cells are calculated from the central cell as shown in Fig. C.4.

The number of signatures that can be generated within a group in any of those cells is easy to compute. For instance, in a three-lane road with 120 km/h speed limit, CellArea= 3456 m^2 and a maximum of 9 signatures per cell are generated in the assumed conditions. Note that the influence of CellArea value when conditions are not those of the approximation formula is high. For instance, in the event of high density like traffic jam, approximately 170 signatures might be generated in the same cell, whilst in low traffic density, it might happen that only one vehicle exists in a cell and no more signatures can be generated there. However, if those traffic conditions are not stable, we must not either reduce the cell size to reduce the number of possible signatures in the cell because vehicles could move too fast to allow group formation, or increase the CellArea to increase the number of possible signatures because that could lead to too large packets. On the other hand, CellArea formula should not be dependent on the traffic conditions because such conditions may vary over time, and its value has to be clearly defined using only geographic coordinates.

A possible solution to the high density case, where too many signatures are generated in a cell, could be based on a probabilistic approach to data aggregation in such a way that instead of generating a complete aggregated packet, only a few signatures chosen at random by the leader from the nodes in its cell are included in the probabilistically aggregated packet so that its size is decreased.

C.4.5. Probabilistic Verification

The probabilistic verification of the signatures contained in a packet type A is the second basis of our proposal. In particular, the probabilistic aspect is on the choice of the signatures to be verified. Once the signatures are chosen, their verification is done through the usual method based on a Public Key Infrastructure (PKI), which involves obtaining the public key certificates of the signatures, and performing three steps: computation of the hash value of the signed message, decryption of the digital signature with the sender's public

key, and comparison between both values. Since in VANETs it is not possible to rely on mechanisms that require a centralized system such as a centralized Certification Authority, public key certification must be done through some distributed solution like the one used in the implementation of [26].

Probabilistic verification is only used by vehicles out of the danger zone, because those vehicles are unable to verify directly the information that reaches them and their only source of information in through the received packets type A. The proposed probabilistic verification algorithm uses threads, which are lightweight processes that allow a concurrent execution for a faster execution of the whole protocol. In the algorithm shown below, Th[i] denotes a thread for the variable i that takes an integer value between 1 and n, where n denotes the number of aggregated signatures in a received packet. When a vehicle receives an aggregated message, the main process launches as many threads as signatures the message contains but before that, it checks whether there are enough signatures to determine whether the message has been confirmed by a significant number of different vehicles. In the implementation, this minimum number of signatures was set to 3 but in general such a number should be fixed by the source node depending on the traffic density. In our particular simulation, if less than 3 signatures are received the packet is dropped, otherwise each thread Th[i] determines whether to verify the signature S corresponding to position i with a probability Prob(Th[i]), which is expressed using percent with a random number between 0 and 99. If Th[i] defines verification, and the signature is proved to be valid, Th[i] returns a true value informing that it is a valid signature. Otherwise, it returns a false value. The result of all these threads are stored in a structure St. If all fields in the structure St are proved to be valid, it is interpreted as evidence that all the verified signatures are correct so the message is accepted as valid. On the other hand, if St contains some fields that are invalid, this could be interpreted as false message. Before signing a packet, legitimate nodes check whether they can validate the information. Otherwise, they discard the packet.

Prob(Th[i]) will be discussed in detail in the following section, where the reason of its specific limit in the Algorithm 2 is given.

Algorithm 2 Probabilistic Verification of Signatures

```
01: function Main(...)
02:   bool St[n];
03:   Thread Th[n];
04:   for (i=0;i<n;i++) do
05:     ThreadStart[i].CheckSignature(n, St);
06:   end for
07:   if (IsTrueMajority(St)) then
08:     return ReliableMessage;
09:   else
10:     return NotReliableMessage;
11:   end if
12: end Main

13: bool function CheckSignature(n, St)
14:           int j=0;
15:   if(n > 3)
16:     for (i=0;i<n;i++)
17:       Prob(Th[i])=rand(0..99);
18:       if (Prob(Th[i]) > ((1-10/n)*100)) then
19:         string M=RecoverMessage(); //Gets the message
20:         Signature S=RecoverSignature(i); //Gets a signature
21:         St[j]=(VerifySignature(S, M));
22:         j++;
23:       end if
24:     end for
25:   else
26:     //Not enough signatures to verify
27:     return NotEnoughSignatures;
28:   end if
```

```
29: end function

30: bool function VerifySignature(Signature S, String M)
31:   if (IsValid(S, M)) then
32:     return true;
33:   else
34:     return false;
35:   endif
36: end function
```

C.4.6. Verification Probability

To guarantee the validity of a specific message, a first approach would be that at least one thread should produce the verification of a signature. In this way, the probability that at least one thread leads to some signature verification must be as close to 1 as possible. However, from the aggregation's point of view, only one thread verifying a signature is not enough. Suppose that a message with several signatures is received so that only one of them is true and the rest are false. Then, if a thread Th[i] verifies only the true signature, it would ensure that this message is valid. Therefore, there should be more that one thread verifying the signatures of a message sent by a vehicle. In the following we analyze the optimal values to fulfill this restriction.

Each thread can be seen as an independent Bernoulli experiment that with probability p given by the percentage Prob(Th[i]) produces the verification of the i-th signature of an aggregated packet [53]. Let X be the variable given by the number of successes of the n threads which follows a binomial distribution with parameters n and p. Thus, the probability of the event B according to which there are at least two threads that verify the signatures of the packet can be expressed in function of n and p:

$$P\{B\} = 1 - (1-p)^n - n \cdot p \cdot (1-p)^{n-1} \qquad (1)$$

Eq. (1) follows from that $P\{X = 0\} = (1-p)^n$ is the probability that none of the signatures is verified and $P\{X = 1\} = n \cdot p \cdot (1-p)^{n-1}$ is the probability that exactly one of the n signatures is verified.

Our objective is to make $P\{B\}$ as close to 1 as possible. Fig. C.6 shows the relationship among $P\{B\}$, p and n. It can be seen that $P\{B\}$ increases as either p or n increases and quickly approaches 1. Our goal is to choose a p value that makes $P\{B\}$ approaches 1 as much as possible for a fixed n and at the same time is as small as possible because a small value of p implies that a vehicle can potentially save processing time. In conclusion the parameter p must be adequately chosen so that both conditions are fulfilled. This balance is analyzed in the next section.

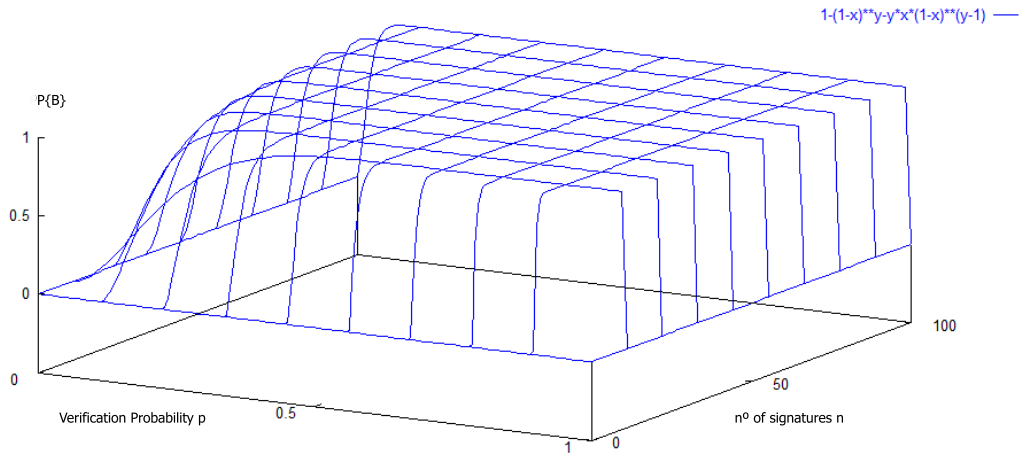


Figura C.6: Probability of Verification of at Least two Signatures

C.4.7. Discussion About Packet Size

When the number of signatures contained in a packet is selected, both the maximum packet size that can be used for VANETs and the minimum number of signatures that is necessary for ensuring information must be taken into account. According to the first condition, packet sizes from 256 bytes to 1500 bytes might be considered in VANETs due to the capacity of the wireless channel. Since in such networks a large number of packets can be generated, it would be advisable not to use the maximum possible packet size because in that case a small number of packets can saturate the channel. In this work we consider the use of about 100 bytes for the message content and the rest for the signatures, so that we can use for the signatures 156 bytes in the worst case and 1400 bytes in the best one.

Given that the result of encrypting with private key almost does not change the size of the input, in this section we only take into account the result of applying the hash function to the message. For instance, the hash function SHA-1 [113] produces an output of 20 bytes, so that with it we could generate 7 signatures per packet of 256 bytes and 70 signatures per packets of 1500 bytes. These data are used as a starting point for the discussion on the optimal values of the parameters.

In order to choose an appropriate value of p for different values of n, the variable $k = n \cdot p$ could be used to leverage the inversely proportional relationship between p and n. Notice that k represents the average number of signatures that a vehicle verifies because n is the total of signatures in the packet and p is the verification probability. If we can find a suitable k, then the corresponding p can be determined. Based on Eq. (1), we can obtain the relationship between $P\{B\}$ and n in terms of different values of k, so that the value of p can be determined. Given that the probability p has a maximum value of 1, and we have that using SHA-1, n would be greater than 6, we use this value to conclude through Fig. C.7 if we choose k=*6*, $P\{B\}$ is close to 1 but not enough. However, with k=10, $P\{B\}$ is sufficiently close to 1 when the packet contains 10 or more signatures. Therefore, we can conclude that k can be set to a constant value, for instance $k = 10$ and once k is fixed, p can be computed from k/n so that it takes the value $10/n$, which is the limit used in the Algorithm 1. In other words, we can express p in terms of the value of n. For example, a vehicle that receives a message with 20 signatures, verifies each signature with probability 0.5. Thus, if n is less than 10, p is ≈ 1, what looks correct because with so little evidence of an event, all signatures should be verified to avoid potential attacks.

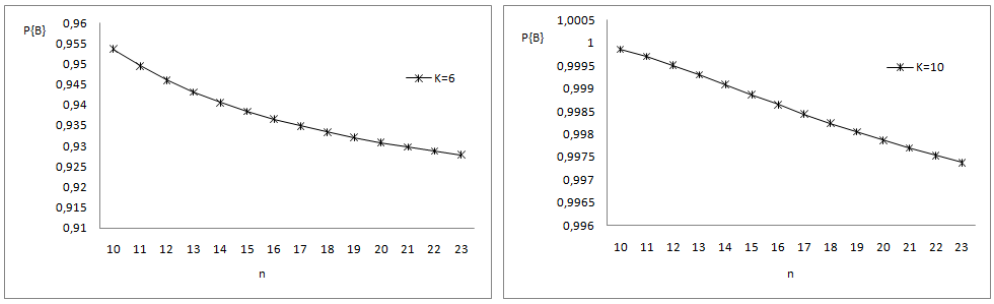


Figura C.7: Probability of Enough Verifications with k=6 and k=10

Considering the minimum number of signatures that a packet can contain to maximize the probability $P\{B\}$, and calculating the probabilities and the maximum number of signatures that fit in a packet, we can discuss the optimal hash functions to be used in this type of network with the proposed scheme. Table 2 shows the corresponding data for different hash functions, MD5 [122] producing 16 bytes, SHA-1 producing 20 bytes and SHA-256 producing 32 bytes. Consequently, if 100 bytes are used for the message content, the table shows the maximum number of signatures that can be appended in each aggregated packet. For example, a packet of 512 bytes can include a total of 9 signatures with SHA-1. Even the use of the hash function SHA-256 with 512-byte packets is possible and increases the security of the scheme.

Tabla C.2: Parameters of Hash Functions for a 100-byte Message

Hash Function	**Packet Size**	**Signature Size**	**N. of Signatures**
MD5	256	156	9
SHA-1			7
SHA-256			4
MD5	512	412	25
SHA-1			20
SHA-256			12
MD5	1024	924	57
SHA-1			46
SHA-256			28
MD5	1500	1400	87
SHA-1			70
SHA-256			43

C.4.8. Security Analysis

In order to analyze the effectiveness and robustness of our proposal, we performed a security evaluation of the proposal. This section briefly analyzes several possible adversarial attacks and how our system can resist them.

In particular, eight attacks are briefly discussed below: Sybil attack, false information generation, aggregated message discarding, false aggregated message generation,

impersonation attack, and attacks on privacy.

- **Sybil attack.** This type of attack occurs when a malicious node creates different false identities in the system in order to get more influence in the network. Our system only allows one identity per node using as unique identity a parameter such as its telephone number, so this attack is impossible.

- **Generating false information.** An attacker may forge a message that does not correspond to the true characteristics of its real environment information. This case is dismissed by the data aggregation structure because the other vehicles sign the message only if they detect the same event and conditions that are specified in the message.

- **Discarding aggregated messages.** Attackers may try to discard some aggregated message, resulting in biased information dissemination. To solve this problem some cooperation scheme could be used like the one proposed in [12]. Anyway, the damage that may result from the removal of one or a few data aggregation packets is not very high, since more than one aggregated packet related to each event are usually generated in the proposed system.

- **Modifying aggregated messages.** An attacker might modify aggregated messages transmitted through the network. However, when a vehicle has no direct contact with the information contained in a received aggregated message because it is not in the danger zone, it has to perform some verification of the signatures. First, a vehicle in the uncertainty and security zones must verify that some signatures according to the verification probability match the message. Besides, the vehicles in the security zone must verify the existence of different aggregated packets from different reactive groups warning about the same event. In any case, any false aggregated message would be detected with a high probability.

- **Impersonation attack.** In order to protect the scheme against possible attacks consisting in claiming to be a legitimate node, robust public key management scheme and strong authentication are used in the implementation of the proposal.

- **Attacks on privacy.** Privacy is an important concern in VANETs. Simple PKI-based communication systems do not protect nodes privacy because the broadcasting of any message usually includes the signer signature and certificate. Although these data do not contain any sensitive information on the sender, the receiver might be able to track it. This issue was addressed in the implementation of the proposal through the use of changing pseudonyms in beacon broadcast announcing presence so that to link a beacon to the corresponding nodes certificate is only possible after strong bidirectional authentication. In this way, tracking all the movements of specific vehicles is not possible in the proposed scheme.

- **False trust increase.** A legitimate node who does not detect an event could try to add its signature to the corresponding aggregated packet in order to increase the trust about the event. This attack is useless in the proposed scheme because the verification of only one signature is not enough to accept the validity of the warning message.

- **Leader's attack.** A group leader could try to add a false signature to an aggregated packet about a true event in order to forge the event discarding by the other nodes. Such an attack would be easily detected by the other nodes in the danger zone where they can correctly detect the event.

C.4.9. Performance Evaluation

In order to analyze how fast and effective is the data aggregation module, several NS-2 simulations have been done based on data got from the real device implementation. This section presents some details and results obtained by averaging 100 simulations using different network sizes over the same area of 1000 square meters, that is to say, considering different situations regarding traffic density. Due to computational constraints, our simulations are formed by networks between 10 and 40 nodes. The most relevant parameters selected for the demonstration have been: total number of lanes for each direction= 3, simulation time= 1000 s, moment when motions starts= 0 s, moment when retransmissions begins= 40 s, retransmission period= 10 s, transmission range= 100 m, traveled distance before the event happens= 800 m. Speeds and directions of nodes in the NS-2 simulations

were random. Two different types of roads and two different types of events were considered in such a way that the considered storage time of packets type A ranged from 3 min for a parking event announced in a conventional road, to 5 and 10 min for a traffic jam event in a highway and in a conventional road, respectively.

The aim is to evaluate on the one hand the number of generated packets using reactive groups, and on the other hand, the effects of our proposal in the computational complexity shown through the time that our aggregation mechanism takes to warn all network nodes about the existence of an event. Finally, we analyzed the time spent in verifying the signatures contained in a packet for different packet sizes according to the proposed probabilistic verification. For the purpose of these simulations we used and compared our proposal with a basic scheme without any aggregation mechanism.

The first simulations correspond to a traffic jam on the road and the corresponding warning packet forwarding with and without reactive groups that is to say with and without aggregation. In Fig. C.8 we can see that the number of generated packets in the simulation without aggregation is much higher than when using our aggregation scheme, even though in this case such a number includes the packets generated by the group formation algorithm. The decrease in the number of generated packets allows making better use of the channel. Thus, Fig. C.8 shows how using aggregation with reactive groups we can reduce the number of generated packets and improve the channel usage.

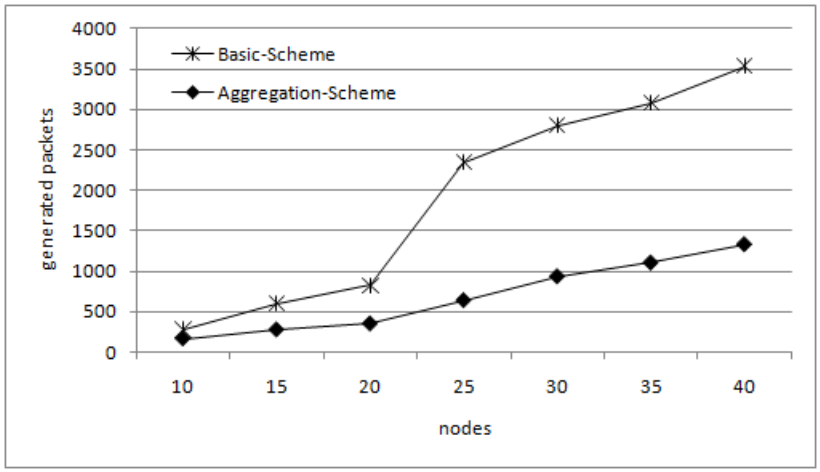


Figura C.8: Number of Generated Packets

Connections between vehicles in VANETs are usually short so any proposed mechanism requiring communication between vehicles has to be fast enough to prevent data loss in communications. In Fig. C.9 we can see the impact of node density in time cost of communication both with the basic scheme and with the proposed aggregated scheme, and conclude that using this aggregation mechanism does not involve a significant increase in time cost of management when the packet is received. We can see that when the network size is between 10 and 20 nodes, the mechanism is even better. This effect is surprising but can be explained by the fact that communication is better organized when using aggregation and reactive groups, and permits to alert a greater number of nodes in less time. As the number of nodes in the network increases, the time it takes to process aggregated packet to warn all nodes also increases.

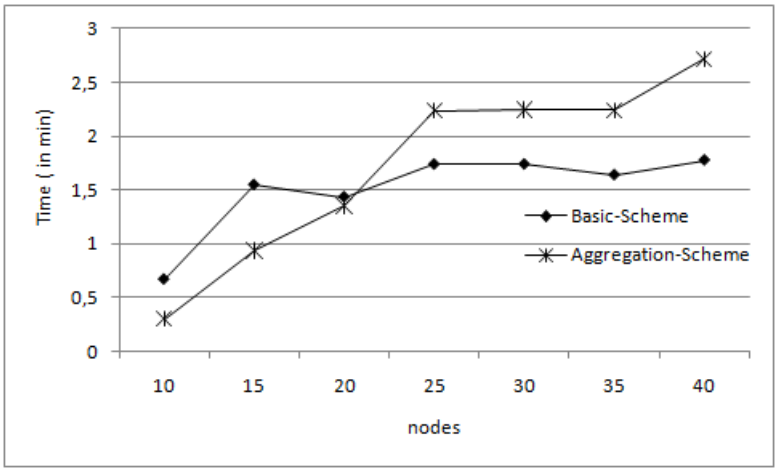


Figura C.9: Time Cost

Another sensitive aspect of this implementation is the verification of signatures and the delay it causes. As we discussed in previous sections, it is impractical to verify all the signatures that a packet contains so in this work we propose a probabilistic algorithm in which only some of them are checked. To analyze the delay and find out whether our proposal improves the verification time we have made several simulations. We took into account the different sizes of packets and the maximum number of signatures that fit in each one with the different hash functions.

Table C.3 shows the simulation results. For different packet sizes and numbers of signatures, it shows the average times in minutes of 100 simulations. We can see that when the number of signatures contained in the packet is lower than 10, the test time is approximately equal for both methods although our method has a slightly higher time cost due to the calculation of probabilities. In the last column we can see the average of the number of signatures that our scheme verifies. Due to the proposed probabilistic approach, as the number of signatures increases, the time consumed by our scheme decreases, which makes our system more efficient. According to these results, our scheme maintains the verification time in about 0.138 min $\approx$ 8.3 s regardless of the number of signatures contained in the packet. Therefore, we can conclude that the use of a probabilistic approach greatly improves the time spent on signature verification, and consequently demonstrates the scalability of the proposal.

Tabla C.3: Simulation for Different Hash Functions

Hash Function	**Time Basic Scheme**	**Time Aggr. Scheme**	**N. of Verified Signatures**
MD5	0,14121162	0,148480218	9
SHA-1	0,093453982	0,096037364	7
SHA-256	0,058070086	0,064743644	4
MD5	0,379342788	0,133570748	10
SHA-1	0,289680197	0,111626543	8
SHA-256	0,182566924	0,171111552	10
MD5	0,81221526	0,144320619	9
SHA-1	0,728451986	0,146179821	9
SHA-256	0,452038378	0,132944675	9
MD5	1,246871085	0,135551543	11
SHA-1	1,007615639	0,143795704	10
SHA-256	0,679152238	0,135551543	11

Appendix D

VANETs in Phones

In this section an implementation analysis of the above proposal is included. To check the effectiveness of the aggregation and verification processes, the best option would be by testing them in a large scale implementation. However, doing this with a high number of real devices is not easy. Therefore, the chosen alternative has been to implement the scheme in a few devices to take real data from them and then use those data in NS2 simulations.

D.1. VAiPho Structure

We have implemented the proposal mechanism in a real VANET environment in order to obtain real data that are used in a software simulation. As aforementioned, the real device implementation of the proposal was combined with the implementation of public key management, strong authentication and pseudonym schemes in order to prevent several types of attacks.

VANETs are traditionally defined in bibliographic references as a set of special devices called On Board Units, which are installed in vehicles, and Road Site Units, which are deployed on the roads. These networks make possible the detection of events and the exchange of messages about such events between nodes. Both issues were assumed to be solved in 2011, but the economic crisis has prevented this VANET deployment because the aforementioned installations are too costly both to governments and to users.

In order to try to deploy VANETs without additional costs either on users or on

governments, a new model has been proposed of a fully self-organized VANET where no RSUs are assumed and the OBU role is partially played by a software application called VAiPho (VANET in Phones) [26], running on mobile phones in vehicles. Given that many references suggest that vehicle OBUs connect via Wi-Fi using 802.11 protocols, VAiPho implements VANETs in smartphones, because these are widespread devices that provide Wi-Fi connectivity, GPS and enough computing capability. In particular, VAiPho is a free secure communication system for spontaneous and self-managed vehicular networks that use smartphones and do not require any infrastructure in vehicles or roads. The operating mode is completely distributed and decentralized and takes into account the protection of privacy and integrity against possible attacks, as aforementioned. Below the different modules implemented in VAiPho including the aggregation scheme here proposed are shown in Fig. D.1:

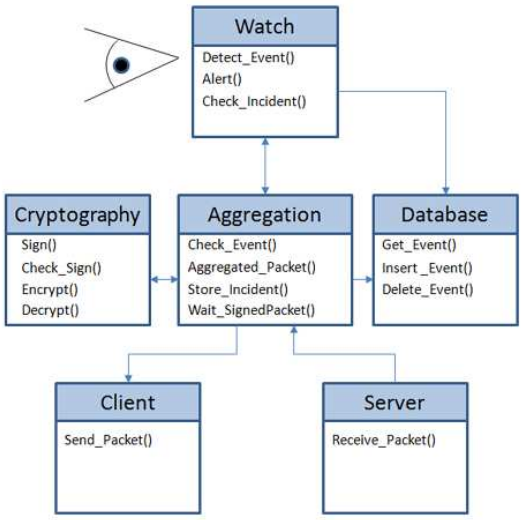


Figura D.1: Modules

- **Watch Module** is responsible for detecting events on the road, such as traffic jams. The module is continuously consulting the GPS in order to get the type of road, minimum and maximum allowed speed and car speed. If this speed is abnormally low according to the road speed, the device generates an alert about a possible traffic

jam. After a period of time, if the vehicle continues at an abnormally low speed, it generates a traffic jam warning that launches the Aggregation Module and is stored in the database.

- **Database Module** contains all the functions responsible for interacting with the database like event insertion, creation, consult and deletion. The database stores event warnings produced by the Watch Module or included in information packets received by the device.

- **Aggregation Module** implements the aggregation functionality explained in this paper on an event warning produced by the Watch Module. The first task is to check whether the event indicated by the Watch Module exists or not in the database so that if it is not in the database, it means that it is a new event so it launches an aggregation procedure starting a group formation process.

- **Client Module** sends messages either in broadcast mode or in unicast mode to a node, in order to form a reactive group with information about an event and waits until vehicles in the neighborhood sign the messages to form an aggregated packet.

- **Server Module** is responsible for receiving and classifying packets into three groups: packets type R to be signed if the receiving node agrees with the reported event; packets type S with signatures to generate aggregated packets; and aggregated packets type A. Depending on the type of received packet, the module proceeds with the corresponding action.

- **Cryptography Module** is responsible for signature, encryption and decryption of information.

D.2. Real Device Implementation

The real device implementation was made with four mobile devices each inside a different car, of which three were used to detect a traffic jam and create the corresponding signed aggregated packet type A after the corresponding exchange of the packets type W,

R and S during the formation of the reactive group. The fourth device was inside a vehicle in the uncertainty zone so it could not see the reported event but received the aggregated packet type A and verified the signatures. In this case, since there were so few devices, they verified all the three signatures. Fig. D.2 shows in detail the simulated situation.

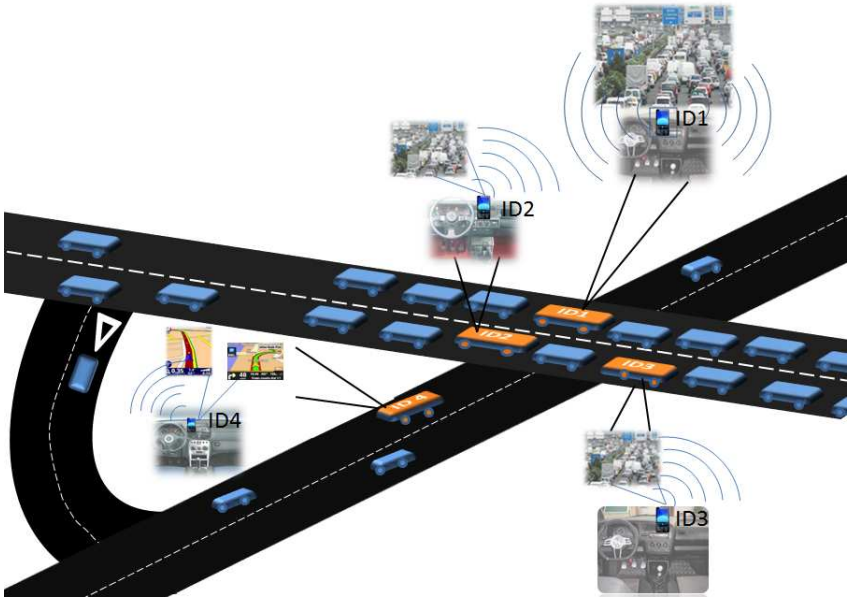


Figura D.2: Simulated Situation

In Fig. D.2 the three mobile devices capable of detecting the traffic jam are denoted ID1, ID2 and ID3. Node ID1 starts a data aggregation procedure by sending a packet type R to all nodes in the network through broadcast. Nodes ID2 and ID3 and even ID4 may receive this packet because they are in the neighborhood. However, only ID2 and ID3 are able to detect the traffic jam and belong to the corresponding reactive group, so they sign the warning message and send the corresponding packet type S back to node ID1. Once node ID1 receives the information, it generates a packet with the signatures of ID2 and ID3 and broadcasts the resulting aggregated packet type A. When node ID4 receives the information, it verifies the signatures contained in the packet and if they are correct, it warns the driver and suggests an alternative route.

When the system starts, both the Server and the Watch Modules begin. As shown in the left image of Fig. D.3, the Watch Module collects the GPS information and when

the vehicle speed is lower than 4 km/h, it indicates that there is a possible jam. After a few seconds it checks the conditions again in order to discard the possibility of traffic lights or a stop sign and if the situation remains equal, it inserts the reported event in the database as shown in the right image of Fig. D.3. All nodes are alerted but the process is only initiated by who first detects the event. In particular, the mobile node ID1 first detects a possible traffic jam and after certain time it checks again its speed to finally determine that it is a traffic jam. Then, the node stores the event in the database and sends a message with the information to form a reactive group. An aggregated packet is created with the signatures of all the nodes that form its reactive group and answer to its request.

Figura D.3: Traffic Jam Detection

Nodes ID2 and ID3 receive the packet and check their Watch information. If they agree, they return a message with the signed information such as can be seen in Fig. D.4. As explained above, the signature of the message is computed with the private key of each node and a hash function. Once signed, the message is forwarded to node ID1 indicating conformity with the received information.

ID1 receives two signatures from ID2 and ID3. Then, it generates the aggregated packet because there are no more devices in the reactive group. Finally, it broadcast the signed packet to the network so that it can be relayed to reach other vehicles that cannot detect the event directly.

The aggregated packet is received by node ID4 who verifies the signatures before taking the information as valid. After such verification, the information is provided to

Figura D.4: Traffic Jam Confirmation

the GPS to determine whether the vehicle should continue in the same route or take an alternative one. Fig. D.5 shows the coordinates of the reported event and how the warning is received by the device. From the implementation we conclude that the proposed aggregation scheme works well in VAiPho solution for VANET deployment. Several videos explaining VAiPho operation and implementation details can be found on the website [26].

Figura D.5: Signature Verification

Among the parameters obtained from the described real implementation, the most remarkable ones are the maximum generation time for packets type A, which was 2 min including the creation of the reactive group; the time required for the exchange of messages between mobile devices, which was less than 1 s; and the maximum distance between communicating devices, which was 300 m. The time for signature verification depends on the used hash function. All these data were used in the NS2 simulation described below.

Appendix E

Conclusions and Future Works

E.1. Conclusions

The secure and efficient exchange of information in VANETs, which is the main theme of this Thesis, involves more challenges than in any other type of ad-hoc networks, like MANETs and sensor networks, due to several unique features of VANETs. Firstly, the lack of infrastructure and centralized authority to control and manage the network increases the vulnerability of these networks, because those tasks have to be done by the users. On the other hand, the retransmission of packets requires the cooperation of nodes so that communications are dependent on the fact that their individual behaviour does not decrease the efficiency and reliability of transmitted information. Besides, the limited resources of mobile devices used in the implementation proposed here, such as memory, battery, computational capacity or bandwidth, are an additional difficulty in proposing solutions to the problem of retransmission. Finally, the content of the information generated by individual nodes within the network must considered inherently unreliable because in general it is easy to generate packets with false content, leading to distrust on the future received information, and in the worst case, to accidents.

With regard to cooperation, we have made various proposals based on different structures to encourage nodes in retransmitting packets. In preliminary studies, we proposed the form of motivation based on rewards in such a way that the rewards had to

be supported either by an operator, or an advertiser, based on their resources. The use of rewards suffered a major problem of uncontrolled spending, and led to other possible attacks in order to get more benefit than what corresponds to each node. Therefore, we designed a new proposal based on reputation where malicious nodes are isolated from the network firstly individually by the node who detects the bad behaviour, and so finally by the global network through the sharing of such information. In addition, other aspects that can lead to failure in cooperation, such as storage or battery consumption, were taken into account when designing the proposed protocol. The participation of nodes in these networks may decrease if they are able to receive network information without participating in it. It has also been shown that the greater the number of nodes that enter the network, better performance we have of the network due the wider coverage. With both goals, the use of encrypted communications has been here proposed so that any node who wants to read transmitted information, must obtain the secret key to decrypt communications. In order to implement this encryption, a new pseudorandom generator has been proposed that aims to promote the recruitment of network users.

When the network grows in size, performance of communications and in general network management falls drastically. Therefore, it is necessary to propose some structure to carry out this process in an ordered way. In this Thesis two structures have been proposed or retransmission, as a tree structure and as a group structure, being the latter the best option for our proposals. In fact, we focus on the formation of reactive groups, which are formed only when necessary. The group structure allows not only to improve communications by decreasing the number of packets on the network but also generates reliable information within the network. In particular, in the proposed group structure, a special node exists that is known as the group leader, who is responsible for managing the communications within its group as well as for generating aggregated packets. While some nodes may try to attack the network by generating false information, the group structure will prevent that attack, because all nodes belonging to the same group have the same view of their environment in a promiscuous mode. In conclusion, both the formation of reactive groups together as well as the data aggregation protects the network against such attacks.

The problem of the generation of false information is not only attacked with the

proposal of the reactive group structure but also with an aggregation mechanism. Given the high mobility of the nodes in the network, the proposed aggregation must be fast in both generation and verification of information. The generation of information is solved by the group structure, which allows to make it effectively. The verification is done once the node receives the information because to its location with respect to the received event, will urge more or less the provision of the information to the driver. It is therefore proposed a new probabilistic verification protocol where not all but a sufficient number of signatures will be checked to determine the veracity of the received information. The complete process is fully protected against attacks of both generation of false information, and of repetition of information.

The scientific community has been seeking a solution for efficient traffic management through the creation of vehicular networks that facilitate communication between vehicles to provide fresh information on different real events that occur on the road. However, implementing such a solution through RSUs on roads and OBUs in vehicles, as it has been stated the current literature, would imply a high cost for both the governments responsible for installing and maintaining road communication infrastructures, and for users who would have to adapt or change their vehicles. Thanks to the application proposed in this thesis, called VAiPho, these economic investments can be avoided by making available the use of a vehicular network to any driver. Also, VAiPho has offered us the opportunity to test all the proposed protocols quickly and economically in in a real environment where effects that were not possible to reflect in a simulator, such as signal fading, loss of communication, interferences, etc., arise naturally. We have also shown that a smartphone is able to run the proposed algorithms so their computational cost may be considered acceptable. Therefore, we can ensure that the use of cryptographic schemes are not a bottleneck for the autonomy and availability of devices.

E.2. Results

The main contributions of this Thesis are summarized in the following points:

- We have addressed the issue of cooperation for VANETs in general, and for a new

proposal of fully distributed and decentralized VANETs in particular. Different countermeasures have been proposed to prevent uncooperative behaviour by offering new practical solutions for VANETs where there is no need for any centralized authority. Thus, one of the objectives of this study has been to develop tools to form a self-managed network by using existing technology, so that nodes can send and receive traffic information via their devices in order to ensure the cooperation of participating nodes. This will allow them to address various security issues in broadcast and performance of such networks via free-cost solutions based on the cooperation of those users who have implemented the proposed schemes on their devices. In particular, we have proposed the use of two reputation lists and of acknowledgment messages as well as different mechanisms based on parameters such as time and distance to allow nodes to detect bad behaviours automatically in order to isolate malicious nodes. Many practical simulations of the proposal have been done to show its strength and usefulness in VANET scenarios, especially in heavy traffic conditions such as traffic congestions, so that the proposals have shown to be successful in reducing the number of selfish nodes in VANETs.

- The design of a new keystream sequence generator based on a non-linear filter of a linear feedback shift register has been presented. The objective of the proposal has been to use it in VANETs in order to prevent the existence of nodes outside the network that can benefit of it by trying to access the transmitted information. Our generator has been analyzed through various statistical tests, and the obtained results about the pseudorandom properties of the output stream have been positive. In addition, this generator has been modified to operate in accordance with the RFID EPC Gen 2 standard with 16-bit technology.

- A new solution has been proposed to meet the need to address the security issue in VANETs consisting in determining whether the traffic information available to the driver is reliable or not. In particular, we have described a new scheme to generate aggregated packets that can not be replaced by any attacker, because they contain the signatures of the vehicles that agree with the reported event. Both in order to prevent

that the warning packets can grow indefinitely, because the scalability is a key aspect in VANETs, and to add confidence to warning messages, signatures are generated according to the formation of reactive groups. On the other hand, when an aggregated packet reaches a vehicle, in order to avoid the delay caused by the verification of signatures in dense environments, we have proposed a probabilistic scheme under which only a few signatures are selected for be verified. The performance of the reactive groups for event generation and detection of misbehaviour has been evaluated, and the results have confirmed that this is a promising approach to increase channel efficiency and confidence in the transmitted information. Moreover, the proposed system has been tested in a real environment, and the results have allowed solving several real problems that do not appear in simulation environments, and obtaining data that have been used for large scale NS-2 simulations, that have also produced promising results on the time required by the system.

- In order to perform the real implementation of the protocols proposed in this Thesis, a new software application for driver assistance, called VAiPho (VANET in Phones), has emerged. VAiPho automatically detects congestions and other traffic events and in real time by using only real mobile phones. The broadcasting of these events takes place in VAiPho by using secure communications between network devices that represent the vehicles. Its main objective is to provide the driver with information about the events that are automatically detected on the road and that may be interesting for him/her. Specifically, jam detection, detection of possible free public parking, location of the parked car and supply of geolocated advertising are functionalities already incorporated in VAiPho. VAiPho implementation has shown the validity of many of the proposals presented in this Thesis in a real environment with real devices that might be considered the first real deployed VANET using only mobile phones within vehicles. In particular, VAiPho has been implemented in a Beta test version on devices with Windows Mobile, Android and Symbian operating systems, that allows sharing information on various events between mobiles on the same platform, concluding that it works well in this type of networks. The media impact of VAiPho after being patent-

ed 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 [134].

E.3. Future Works

- **Cooperation**

  Cooperation in the retransmission of packets within the network depends, among other factors, on the behaviour of nodes, so that it is necessary compensating them not only through the access to transmitted information, but also through rewards for their good behaviour, or through penalties for their malicious behaviour. Since it is impossible to determine a priori what will be the behaviour of network users, it would be useful studying the social behaviour of the network in order to improve the proposals. The best way to do this is by implementing the reward and punishment systems in VAiPho. However, this is a complex task since at the beginning of its deployment the number of users will be low, so that security measures such as certificate revocation should not be very restrictive. So, our main task in terms of cooperation in the next future will focus on the implementation of the IRL and GRL lists in VAiPho application. As for the generator, since we have checked it, only through some statistical tests, because it was not the main objective of this Thesis, then many questions have remained open, such as prove the generator through other test batteries, and if the obtained results were not satisfactory enough, then modify its structure.

- **Aggregation**

  The proposed schemes for automatic event detection provides confidence about its accuracy because it allows the detection of false messages, wrong or malicious. Those algorithm have provided some pretty good results. However, it would also be interesting validating the systems not only regarding the time needed to perform the aggregation and verification of data, but also in terms of their ability to detect different types

of attacks. Another future goal is to check the accuracy of the proposed schemes in practice by implementing large-scale tests with real devices and in real environments to assess the influence of speed and buildings in the wireless communication, and the performance in dense scenarios. Also we plan to implement the proposal based on fuzzy logic for decision making.

- **VAiPho**

  Both in cooperation as in aggregation, a part of future work is checking the performance of the proposals in real environments on a large scale test. This will require the commercialization and diffusion of VAiPho, in order to allow that many users can access it. Therefore, a remarkable future work is the full, efficient and multiplatform implementation of VaiPho, as a fully functional tool, that might provide us with feedback from users in order to enhance the tool by solving any problem that might arise. That would be an opportunity to test the algorithms in real circumstances, where a wide variety of users could provide different points of view to the proposed solutions.

  Finally, the immense range of possible new applications of VAiPho is really spectacular. Now VAiPho allows the detection of congestions and bottlenecks, possible free parking spaces, exact location where the parked vehicle is, and provision of a geolocated advertising platform. However, in the future many more features might be added to VAiPho in an easy and intuitive way, thanks to the modular design of its implementation. In addition, the tool might be used in any other application non-related with VANETs, since the basis of the system is to allow secure communications between mobile phones. Therefore, by now the applications of VAiPho might be considered unlimited.

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: 3rd International Symposium on Wireless Communication Systems, ISWCS, 2006, pp. 307–311.

[2] L. Anderegg, S. Eidenbenz, Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents, in: Proceedings of the 9th annual international conference on Mobile Computing and networking, MobiCom, 2003, pp. 245–259.

[3] S. Bansal, M. Baker, Observation Based Cooperation Enforcement in Ad-hoc Networks, in: Technical report, 2003.

[4] T. Batz, K. Watson, J. Beyerer, Recognition of Dangerous Situations within a Cooperative Group of Vehicles, in: Proceedings IEEE Intelligent Vehicles Symposium, 2009, pp. 907–912.

[5] S. Buchegger, J. Coudec, Nodes Bearing Grudges: Towards Routing Security, Fairness and Robustness in Mobile Ad-hoc Networks, in: 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, PDP, 2002.

[6] BuscaCar, http://es.androidzoom.com/androidapplications/tools/buscacarovyl.html.

[7] L. Buttyan, J.-P. Hubaux, Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks, in: ACM/Kluwer Mobile Networks and Applications, MONET, Vol. 8, 2003.

[8] L. Buttyan, J.-P. Hubaux, Security and Cooperation in Wireless Networks, Cambridge University Press, 2008.

[9] L. Buttyán, T. Holczer, I. Vajda, On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs, in: European Workshop on Security and Privacy in Ad Hoc and Sensor Networks, ESAS, 2007, pp. 129–141.

[10] P. Z. C. Frese, J. Beyerer, Cooperation of Cars and Formation of Cooperative Groups, in: Proceedings IEEE Intelligent Vehicles Symposium, 2007, pp. 227–232.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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, 2010.

[16] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Merging Subnetworks in VANETs by Using the IEEE 802.11xx Protocol, Submitted to Eurasip Journal of Wireless Communications and Networking, 2012.

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

[18] 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, 2011.

[19] C. Caballero-Gil, P. Caballero-Gil, J. Molina-Gil, Self-organizing Life Cycle Management of Mobile Ad Hoc Networks, Accepted by Security and Communication Network, 2012.

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

[21] 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.

[22] C. Caballero-Gil, P. Caballero-Gil, A. Peinado-Domínguez, J. Molina-Gil, Lightweight Authentication for RFID Used in VANETs, in: Computer Aided Systems Theory EUROCAST, Lecture Notes in Computer Science, Vol. 6927, Springer-Verlag, 2011.

[23] 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).

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

[25] 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: In-

ternational Conference on Computer Systems and Technologies, CompSysTech, 2011, pp. 16–17.

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

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

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

[29] 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, 2010.

[30] 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.

[31] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, C. Hernández-Goya, Flexible Authentication in Vehicular Ad Hoc Networks, in: Proceedings of the 15th IEEE Asia-Pacific Conf. Communications, APCC, 2009, pp. 576–879.

[32] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, A. Quesada-Arencibia, A Simulation Study of New Security Schemes in Mobile Ad-hoc NETworks, in: Computer Aided

Systems Theory EUROCAST 2007, Lecture Notes in Computer Science, Vol. 4739, 2007, pp. 73–81.

[33] 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, SALT, 2011.

[34] P. Caballero-Gil, C. Caballero-Gil, J. 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, TRANSNOVA, 2011.

[35] P. Caballero-Gil, A. Fúster-Sabater, Improvement of the Edit Distance Attack to Clock-Controlled LFSR-based stream ciphers, in: Computer Aided Systems theory EUROCAST, Lecture Notes in Computer Science, Vol. 3643, 2005, pp. 355–364.

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

[37] 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, Vol. abs/1005.3143, 2009, pp. 346–349.

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

[39] G. Calandriello, P. Papadimitratos, J. Hubaux, A. Lioy, Efficient and Robust Pseudonymous Authentication in VANET, in: Proceedings of the 4th ACM International workshop on Vehicular ad hoc networks (VANETs), 2007, pp. 19–28.

[40] 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, 2003.

[41] G. Chang, J. Sheu, C. Chung, Zooming. A Zoom-Based Approach for Parking Space Availability in VANET, in: IEEE 71st Vehicular Technology Conference, VTC, 2010, pp. 1–5.

[42] Cipher, ECRYPT Stream Cipher Projec, http://www.ecrypt.eu.org/stream/index.htm.

[43] O. Consulting, Smart Device Framework, http://www.opennetcf.com/CompactFramework/-Products/SmartDeviceFramework/tabid/65/Default.aspx.

[44] Y.-G. D., End-term Project Directed by Caballero Gil P., Molina Gil J. Implementación de comunicaciones seguras en la plataforma Android para asistencia a la conducción. ETSI Ingeniería Informática. Universidad de La Laguna. Sobresaliente (10) (por unanimidad), June 2011.

[45] W. .Dabbous, C. Huitema, PROMETHEUS: Vehicle to Vehicle Communications, in: Research Report, INRIA-Renault collaboration, 1988.

[46] V. Daza, J. Domingo-Ferrer, F. Sebe, A. Viejo, Trustworthy Privacy-Preserving Car-Generated Announcements in Vehicular Ad Hoc Networks, in: IEEE Transactions on Vehicular Technology, Vol. 58, 2009, pp. 1876–1886.

[47] T. Delot, N. Cenerario, S. Ilarri, Vehicular Event Sharing with a Mobile Peer-to-Peer Architecture, in: Transportation Research Part C: Emerging Technologies, Elsevier, Vol. 18, 2010, pp. 584–598.

[48] Detector, Grupo Detector S.A., http://www.grupodetector.com/.

[49] S. Dietzel, E. Schoch, B. Konings, M. Weber, F. Kargl, Resilient Secure Aggregation for Vehicular Networks, in: IEEE Network, Vol. 24, 2010.

[50] S. Dornbush, A. Joshi, StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's, in: Proceedings of the VTC2007-Spring Vehicular Technology Conf. IEEE 65th, pp. 11–5.

[51] F. Dotzer, L. Fischer, P. Magiera, VARS: A Vehicle Ad-hoc Network Reputation

System, in: Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM, 2005, pp. 454–456.

[52] S. Eichler, C. Merkle, M. Strassberger, Data aggregation system for distributing inter-vehicle warning messages, in: Proceedings of the 31st IEEE Conf. on Local Computer Networks, IEEE Computer Society, 2006, pp. 543–544.

[53] W. Feller, An Introduction to Probability Theory and Its Applications, 3rd ed., Wiley, 1968.

[54] Finder, Car Finder, http://es.androidzoom.com/androidapplications/transportation/car-findersmsf.html.

[55] H. Fischer, Digital Beacon Behicle Communications at 61 GHz for Interactive Dynamic Traffic Management, in: Eighth International Conference on Automotive Electronics, 1991, pp. 120–124.

[56] E. Fonseca, A. Festag, A Survey of Existing Approaches for Secure Ad Hoc Routing and Their Applicability to VANETS, in: Technical Report NLE-PR-2006-19, NEC Network Laboratories, 2006.

[57] M. Frank, P. Marini, M. Plaggemeier, CineMA: Cooperation Enhancements in Manets, in: The 29th Annual IEEE Conference on Local Computer Networks, LCN, 2004, pp. 86–93.

[58] W. Franz, H. Hartenstein, M. Mauve, Inter-Vehicle-Communications Based on Ad Hoc Networking Principles, in: The FleetNet Project, 2005.

[59] R. French, Historical Overview of Automobile Navigation Technology, in: 36th IEEE Vehicular Technology Conference, Vol. 36, 1986, pp. 350–358.

[60] R. French, Automobile Navigation in the Past, Present and Future (1987) 542–551.

[61] G-Park, http://itunes.apple.com/es/app/g-park/ id284943236?mt=8.

[62] W. Gillan, PROMETHEUS and DRIVE: Their Implications for Traffic Managers, in: Vehicle Navigation and Information Systems Conference, Conference Record, 1989, pp. 237–243.

[63] D. J. Golic, On the Linear Complexity of Functions of Periodic GF(q) Sequences, in: IEEE Trans on Inform theory, Vol. IT-35, 1989.

[64] L. Gollan, C. Meinel, Digital signatures for automobiles, in: Proceedings of Systemics, Cybernetics and Informatics (SCI), 2002, pp. 225–230.

[65] P. Golle, D. Greene, J. Staddon, Detecting and Correcting Malicious Data in VANETs, in: Proceedings of the 1st ACM International workshop on Vehicular ad hoc networks, 2004, pp. 29–37.

[66] S. Golomb, Shift Register Sequences, Revised Edition, in: Aegean Park Press, Laguna Hills, CA, 1982.

[67] Google, Google Maps, Traffic option, http://maps.google.com/.

[68] K. Hartman, J. Strasser, Saving Lives Through Advanced Vehicle Safety technology: Intelligent Vehicle Initiative Final Report, in: FHWA-JPO-05-057, 2005.

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

[70] C. Hernández-Goya, P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Cooperation Enforcement Schemes in Vehicular Ad-hoc Networks, in: Computer Aided Systems Theory - EUROCAST 2009: 12th International Conference, Lecture Notes in Computer Science, Vol. 5717, Springer-Verlag, 2009, pp. 429–436.

[71] C. Hernández-Goya, P. Caballero-Gil, J. Molina-Gil, C. Caballero-Gil, Cooperation Requirements for Packet Forwarding in Vehicular Ad-hoc Networks, in: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, 2009, p. 56.

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

[73] J.-P. Hubaux, L. Buttyán, S. Capkun, The Quest for Security in Mobile Ad Hoc Networks, in: proceedings of the 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing, 2001.

[74] K. Ibrahim, M. Weigle, Accurate Data Aggregation for VANETs, in: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks, 2007, pp. 71–72.

[75] K. Ibrahim, M. Weigle, Optimizing CASCADE Data Aggregation for VANETs, in: Proceedings of the IEEE International Workshop on Mobile Vehicular Networks MoVeNet, 2008, pp. 724–729.

[76] O. Ileri, S.-C. Mau, N. B. Mandayam, Pricing for Enabling Forwarding in Self-Configuring Ad Hoc Networks, in: IEEE Journal on Selected Areas in Communications, IEEE J-SAC, Special Issue on Wireless Ad Hoc Networks, Vol. 23, 2005, pp. 151–162.

[77] J. Isaac, S. Zeadally, J. Camara, Security Attacks and Solutions for Vehicular Ad Hoc networks, in: IET Communications, Vol. 4, 2010, pp. 894–903.

[78] M. Jakobsson, J.-P. Hubaux, L. Buttyan, A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks, in: Proceedings of Financial Cryptography, 2003.

[79] D. Jiang, L. Delgrossi, IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments, in: IEEE Vehicular Technology Conference Spring, VTC, 2008, pp. 2036–2040.

[80] R. Jurgen, Smart Cars and Highways Go Global, in: IEEE Spectrum, Vol. 28, 1991, pp. 26–36.

[81] H. Kawashima, Japanese Perspectives of Driver Information Systems, in: Transportation, Vol. 17, 1990, pp. 263–284.

[82] P. Kolios, V. Friderikos, K. Papadaki, Ultra Low Energy Store-Carry and Forward Relaying Within the Cell, in: Proceedings of the Vehicular Technology Conference Fall, IEEE, 2009, pp. 1–5.

[83] S. Lee, G. Pan, J. Park, M. Gerla, S. Lu, Secure Incentives for Commercial ad Dissemination in Vehicular Networks, in: Proceedings of ACM MobiHoc, 2007.

[84] LGParking, http://es.androidzoom.com/androidapplications/lifestyle/lg-parkingueamscreenshots.html.

[85] F. Li, J. Wu, A Winning-Probability-based Incentive Scheme in Vehicular Networks, in: Proceedings of IEEE International Conference on Network Protocols, ICNP, 2008.

[86] F. Li, J. Wu, FRAME: An Innovative Incentive Scheme in Vehicular Networks, in: Proceedings of IEEE International Conference on Communications, ICC, 2009.

[87] J. Liu, V. Issarny, Enhanced Reputation Mechanism for Mobile Ad-hoc Networks, in: iTrust, 2004, pp. 48–62.

[88] N.-W. Lo, H.-C. Tsai, A Reputation System for Traffic Safety Event on Vehicular Ad Hoc Networks,10 pages. doi:10.1155/2009/125348, in: EURASIP Journal on Wireless Communications and Networking,, Vol. 2009, 2009.

[89] LocalizadorGPS, http://es.androidzoom.com/androidapplications/communication/localizador-gps-aparcar-cochetwda.html.

[90] C. Lochert, B. Scheuermann, M. Mauve, A Probabilistic Method for Cooperative Hierarchical Aggregation of Data in VANETs, in: Ad Hoc Networks, Elsevier, Vol. 8, 2010, pp. 518–530.

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

[92] J. Massey, Cryptography: Fundamentals and Applications, in: ATS Seminar, Zürich,Switzerland, 1994.

[93] J. L. Massey, Shift Register Synthesis and DCH Decoding, in: IEEE TRans, on Inform theory, Vol. IT-15, 1969.

[94] 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.

[95] S. Milgram, The Small World Problem, in: Psychology Today, Vol. 2, 1967, pp. 60–67.

[96] 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, 2007.

[97] 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, pp. 94–97.

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

[99] 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, pp. 10–15.

[100] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, A Vision of Cooperation Tools for VANETs, in: IEEE International Workshop on Data Security and PrivAcy in wireless Networks DSPAN-IEEE WoWMoM, 2010, pp. 1–4.

[101] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Countermeasures to Prevent Misbehaviour in VANETs, In second round review at Journal Universal Computer Science, JUCS, 2011.

[102] 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, pp. 91–102.

[103] J. Molina-Gil, P. Caballero-Gil, C. Caballero-Gil, Probabilistic Aggregation for Data Authentication in VANETs, In third round at Transportation Research Part C: Emerging Technologies, Elsevier.

[104] 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, 2011, pp. 460–465.

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

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

[107] J. Molina-Gil, P. Caballero-Gil, C. Hernández-Goya, C. Caballero-Gil, Agregación de datos para autenticar información en VANETs, in: XI Reunión Española sobre Criptología y Seguridad de la Información, RECSI, 2010.

[108] 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, pp. 33–40.

[109] H. Mousannif, I. Khalil, H. A. Moatassime, Cooperation as a Service in VANETs, in: Journal of Universal Computer Science, Vol. 17, 2011.

[110] T. Nakahara, N. Yumoto, ITS Development and Deployment in Japan, in: Proceedings of IEEE Conference on Intelligent Transportation Systems, 1997, pp. 631–636.

[111] M. Newman, A.-L. Barabási, D. J. Watts, The Structure and Dynamics of Networks, U.S.A, Princeton University Press, 2006.

[112] NIST, Random Number Generation Technical Working Group, http://csrc.nist.gov/rng/.

[113] NIST, Secure Hash Standard. FIPS 180-1, nist, us department of commerce ed., Springer-Verlag, 1996.

[114] A. Olteanu, Y. Xiao, Security Overhead and Performance for Aggregation with Fragment Retransmission (AFR) Very High-Speed Wireless 802.11 LANs, in: Proceedings of IEEE Transactions on Wireless Communications, Vol. 9, 2010, pp. 218–226.

[115] R. Panayappan, J. Trivedi, A. Studer, A. Perrig, VANET-Based Approach for Parking Space Availability, in: VANET, Proceedings of the fourth ACM international workshop on Vehicular Ad Hoc Networks, 2007, pp. 75–76.

[116] S. Panichpapiboon, W. Pattara-Atikom, Connectivity Requirements for a Self-Organizing Vehicular Network, in: IVS, 968-972., 2008.

[117] K. Paul, D. Westhoff, Context Aware Detection of Selfish Node in DSR based Ad-hoc Network, in: Vehicular Technology Conference, Proceedings, VTC, IEEE 56th, Vol. 4, 2002, pp. 2424–2429.

[118] F. Picconi, N. Ravi, M. Gruteser, I. L., Probabilistic Validation of Aggregated Data in Vehicular Ad-hoc Networks, in: Proceedings of the 3rd ACM International Workshop Vehicular Ad Hoc Networks (VANET), 2006, pp. 76–85.

[119] M. Raya, A. Aziz, J.-P. Hubaux, Efficient Secure Aggregation in VANETs, in: International ACM Conference on Mobile Computing and Networking, 2006, pp. 67–75.

[120] M. Raya, J. Hubaux, The Security of Vehicular Ad Hoc Networks, in: ISWCS 307-311.SASN 2005, Proceedings of the 3rd ACM workshop on Security of Ad Hoc and Sensor Networks, 2005, pp. 11–21.

[121] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, Eviction of Misbehaving and Faulty Nodes in Vehicular Networks, in: IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks, Vol. 25, 2007, pp. 1557–1568.

[122] R. Rivest, The MD5 Message-Digest Algorithm, Request for Comments (RFC) 1321, in: Internet Activities Board, Internet Privacy Task Force, 1992.

[123] D. Rosen, F. Mammano, R. Favout, An electronic Route-Guidance System for Highway Vehicles, in: EEE Transactions on Vehicular Technology, Vol. 19, 1970, pp. 143–152.

[124] S. Sachs, P. Varaiya, A Communication System for the Control of Automated Vehicles, in: Path Technical Memorandum, 1993.

[125] N. B. Salem, L. Buttyan, J.-P. Hubaux, M. Jakobsson, Node Cooperation in Hybrid Ad Hoc Networks, in: IEEE Transactions on Mobile Computing, TMC, Vol. 5, 2006.

[126] R. K. Schmidt, T. Leinmuller, E. Schoch, A. Held, G. Schafer, Vehicle Behavior Analysis to Enhance Security in VANETs, in: Proceedings 4th Workshop on Vehicle to Vehicle Communications, V2VCOM, 2008.

[127] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, N. McKeown, Automated Vehicle Control Developments in the PATH Program, in: IEEE Transactions on Vehicular Technology, Vol. 40, 1991, pp. 114–130.

[128] C. Spotter., http://itunes.apple.com/us/app/car-spotter/id293089131?mt=8.

[129] J. Sun, Y. Fang, A Defense Technique Against Misbehavior in VANETs Based on Threshold Authentication, in: IEEE Military Communications Conference, Milcom, 2008.

[130] F. Surbock, H. Weinrichter, Interlacing Properties of Shift-Registers Sequences With Generator Polynomials Irreducible Over GF(p), in: IEEE Trans on Inform theory, Vol. IT-24, 1978, pp. 386–389.

[131] SYGIC, Real-Time Traffic, http://www.sygic.com.

[132] TOMTOM, HD Traffic, http://www.tomtom.com/services/ service.php?id=2.

[133] Urban, Urban Mobility Report, Texas Transportation Institute, Texas A&M University, 2010.

[134] VAiPho, Medios de Comunicación, http://www.vaipho.com/prensa.

[135] G. S. Vernam, Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications, in: Journal of the American Institute of Electrical Engineers, Vol. 55, 1926.

[136] J. Walker, Drive, Prometheus & GSM, in: Proceedings of the Mobile Radio Technology, Marketing and Management Conference, 1992.

[137] Z. Wang, C. Chigan, Cooperation Enhancement for Message Transmission in VANETs, in: Wireless Personal Communications: An International Journal, Vol. 43, 2007, pp. 141–156.

[138] Z. Wang, C. Chigan, Countermeasure Uncooperative Behaviors with Dynamic Trust-Token in VANETs, in: Proceedings of IEEE International Conference on Communications, ICC, 2007, pp. 3959–3964.

[139] A. Wasef, X. Shen, REP: Location Privacy for VANETs Using Random Encryption Periods, in: ACM Mobile Networks and Applications (MONET), Vol. 15, 2010, pp. 172–18.

[140] WAZE, Real-Time Maps and Traffic Information Based on the Wisdom of the Crowd, http://www.waze.com.

[141] A. Weyland, T. Braun, CASHnet-Cooperation and Accounting Strategy for Hybrid Networks, in: Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2004, pp. 423–424.

[142] M. Williams, PROMETHEUS, The European Research Programme for Optimising the Road Transport System in Europe, in: IEE Colloquium onDriver Information, 1988, pp. 1–9.

[143] L. Wischhof, Self-Organizing Communication in Vehicular Ad Hoc Networks, in: Technische Universitat., 2007.

[144] L. Wischhof, A. Ebner, R. H., Information Dissemination in Self-Organizing Intervehicle Networks, in: IEEE Transactions on Intelligent Transportation Systems, Vol. 6, 2005, pp. 76–85.

[145] N. Xiong, A. Vasilakos, L. Yang, W. Pedrycz, Y. Zhang, Y. Li, A Resilient and Scalable Flocking Scheme in Autonomous Vehicular Networks, in: ACM/Springer Mobile Networks and Applications (MONET), special issue on Advances and Applications in Vehicular Ad Hoc Networks, Vol. 15, 2010.

[146] Y. Yoo, S. Ahn, D. Agrawal, A Credit-Payment Scheme for Packet Forwarding Fairness in Mobile Ad-hoc Networks, in: IEE International Conference on Communications, Vol. 5, 2005, pp. 3005–3009.

[147] C. Zhang, X. Lin, R. Lu, P.-H. Ho, X. Shen, An Efficient Message Authentication Scheme for Vehicular Communications, in: IEEE Transactions on Vehicular Technology, Vol. 57, 2008, pp. 3357–3368.

[148] J. Zhang, A Survey on Trust Management for VANETs, in: Proceedings of the 25th IEEE International Conference on Advanced Information Networking and Applications (AINA), IEEE Computer Society, 2011, pp. 105–112.

[149] S. Zhong, Y. Yang, J. Chen, Sprite: A simple, Cheat-proof, Credit-Based System for Mobile Ad-hoc Networks, in: Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM, IEEE Societies, Vol. 3, 2003, pp. 1987–1997.