DRA. DÑA. PINO CABALLERO GIL, CATEDRÁTICA DEL DEPARTAMENTO DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS DE LA UNIVERSIDAD DE LA LAGUNA

Y

DR. DON CÁNDIDO CABALLERO GIL, PROFESOR CLI DEL DEPARTAMENTO DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS DE LA UNIVERSIDAD DE LA LAGUNA

CERTIFICAN:

Que la presente memoria titulada "Algoritmos criptográficos y aplicaciones seguras para escenarios de transporte" ha sido realizada bajo su dirección por el Ingeniero Francisco Martín Fernández y constituye su Tesis para optar al grado de Doctor en Informática.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos que haya lugar, firman el presente en

San Cristóbal de La Laguna, a 26 de Mayo de 2016

Algoritmos criptográficos y aplicaciones seguras para escenarios de transporte

TESIS DOCTORAL

Francisco Martín Fernández

Departamento de Ingeniería Informática y de Sistemas
Grupo de Investigación CryptULL
Universidad de La Laguna

Junio 2016

Algoritmos criptográficos y aplicaciones seguras para escenarios de transporte

Memoria que presenta para optar al titulo de
Doctor en Informática con mención Internacional

Dirigida por la Catedrática Pino Caballero Gil
Y por el Doctor Cándido Caballero Gil.

Departamento de Ingeniería Informática y de Sistemas
Grupo de Investigación CryptULL
Universidad de La Laguna

Junio 2016

Cada persona fue traída aquí por una razón, para todas ellas...

Agradecimientos

La gratitud es una vacuna, antitoxina y un antiséptico.

John Henry Jowett

96.336.000 Segundos, 1.605.600 Minutos, 26.760 Horas, 1.115 Días, 159 Semanas, 37 Meses, algo más de 3 Años... ese es el tiempo que he pasado en un maravilloso grupo de investigación llamado CryptULL. Junto a todas las personas que conforman este grupo, he podido elaborar mi Tesis Doctoral que en esta memoria plasmo.

Por ello, el primer agradecimiento es para el mejor grupo de investigación de la Universidad de La Laguna, que con tan poco hacen tanto. Para la mejor investigadora y líder que he conocido jamás, mi directora Pino, porque no todo es saber investigar, sino que también hay que tener mano diestra para coordinar a un grupo y hacer sentir importante a cada persona que lo conforma. Emulando a Daniel Faraday, tengo claro que si algo sale mal, Pino será mi constante. Para mi co-director Cándido, que más allá de involucrarse de forma incondicional en mi Tesis, se ha involucrado en mil y un fregados que le he propuesto: ¡la de eventos que hemos hecho para la comunidad!. Para el resto de integrantes de CryptULL, para Cande, para Jeza que poco a poco va siendo cada vez más chicharrera... para Álvaro ¡el futuro del grupo!, para los demás integrantes que se han unido paulatinamente, Ale, Iván y Néstor, memorable el Hackathon de Intel en Barcelona (Paul dixit!).

También agradecer a mi familia, a mi Madre, mi Padre, mis hermanos, sobrinos... y a mis amigos, que saben quiénes son de sobra, aquellos que están más cerca o a los que me soportan desde la distancia.

A todo el mundo que me haya ayudado alguna vez en esta Tesis, ya sea directa o indirectamente...

¡Gracias!, porque ¡Funcionó!

Resumen

Cada vez es más frecuente ver cómo la tecnología se funde con la realidad en el uso cotidiano. Esta tendencia se conoce como la Internet de las Cosas o IoT (Internet of Things) y surge ante la necesidad de tener monitorizados e interconectados los dispositivos electrónicos que sean útiles para el ser humano. En esta nueva dimensión aparecen nuevos retos relacionados con la seguridad inalámbrica, ya que ésta es la vía convencional de comunicación entre dichos objetos hiperconectados. En consecuencia, se necesitan algoritmos criptográficos ligeros acordes a la capacidad reducida de cómputo de estos dispositivos.

La hipótesis de partida, y principal motivación de este trabajo, ha sido la consideración de que en el nuevo paradigma de la Internet de las Cosas la seguridad actual no se adapta, en la mayoría de ocasiones, a las necesidades de este tipo de redes. Se ha elegido para este estudio una de las redes inalámbricas que más impacto está teniendo en la investigación de los últimos años: las redes vehiculares. Diseñar nuevos algoritmos criptográficos ligeros para otorgar confiabilidad y veracidad a los usuarios legítimos y garantizar su correcta y eficiente autenticación, y para la gestión de usuarios malintencionados, han sido las grandes líneas seguidas en la investigación descrita en este trabajo.

Por tanto, esta Tesis se centra en la concepción, diseño e implementación de nuevos algoritmos criptográficos para aplicaciones seguras en escenarios de transporte con objeto de resolver varios problemas actuales en este campo. De este modo, entre los objetivos principales de este trabajo se encuentran el análisis de los antecedentes recientes sobre algunas primitivas criptográficas relevantes para Internet de las Cosas, el estudio de las aplicaciones más destacables del área en el campo del transporte, la propuesta de nuevos algoritmos criptográficos, la adaptación de las propuestas realizadas para su utilización sobre redes inalámbricas, la implementación de los algoritmos diseñados en la plataforma Android Open Source Project, la integración de las soluciones en redes heterogéneas formadas por dispositivos móviles con diferentes tecnologías, la evaluación de la seguridad de los sistemas propuestos frente a diversos ataques maliciosos, la comparación de los resultados obtenidos con otros sistemas relevantes parecidos, y el desarrollo de una o

varias aplicaciones móviles que contengan las mejores soluciones propuestas para ser accesibles en entornos reales.

Las contribuciones aportadas por esta Tesis van en la línea del análisis profundo y la mejora de los algoritmos criptográficos actuales para entornos móviles y volátiles que se generan con la aparición de Internet de las Cosas, poniendo especial énfasis en la aplicabilidad a escenarios de transporte. Las contribuciones se pueden resumir brevemente en los siguientes logros conseguidos: una revisión pormenorizada del estado del arte de la seguridad en las comunicaciones inalámbricas, un estudio profundo de los métodos de autenticación y de revocación de usuarios más relevantes para entornos vehiculares, un nuevo método de autenticación de usuarios basado en demostraciones de conocimiento nulo no interactivas, una nueva familia de métodos de revocación de usuarios basados en estructuras de datos autenticadas y árboles hash, un algoritmo innovador que permite obtener un grado cuantitativo de la confiabilidad de un determinado usuario frente a otro basado en la teoría de los seis grados de separación y las redes sociales, y un nuevo sistema de auto-denuncia anónima de infracciones en semáforos.

Este documento comienza con una introducción a los conceptos y fundamentos que sustentan las investigaciones realizadas. Continúa con propuestas de algoritmos diseñados e implementados durante los últimos cuatro años. Se propone un nuevo método de autenticación basado en demostraciones de conocimiento nulas no interactivas. Dicho método ha sido diseñado para entornos donde los dispositivos se mueven de forma constante, como los escenarios de transporte. Se aportan nuevas soluciones para la gestión de usuarios fraudulentos y malintencionados en las redes vehiculares. Haciendo uso de las denominadas estructuras de datos autenticadas y de los árboles hash, se proponen diferentes esquemas de gestión de certificados revocados usando desde árboles k-arios hasta códigos de Huffman. Además, se introducen una serie de aplicaciones móviles que se han diseñado y desarrollado haciendo uso de las investigaciones anteriores y de otras, como un algoritmo propio confiable y seguro de verificación de la confianza de los usuarios. Finalmente, se concluye la memoria con un resumen de las contribuciones más relevantes extraídas de las investigaciones, y una serie de objetivos futuros que se han abierto a partir de ellas.

Esta Tesis ha sido desarrollada bajo el paraguas de un proyecto de investigación del plan Nacional, denominado TUERI [245] gracias a la beca FPI BES-2012-051817 asociada a dicho proyecto, y en conexión también con otros proyectos de investigación financiados por el Ministerio de Economía y Competitividad: DEPHISIT [114], ATLAS [222] y CASUS [242].

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Fundamentos

Hoy en día la innovación y el desarrollo tecnológico están protagonizados por dispositivos minúsculos y móviles que permiten crear entornos inteligentes que hasta hace un par de años eran impensables. Este nuevo paradigma que surge con la proliferación de la Internet de las Cosas abre un abanico inmenso de posibilidades a la hora de idear y diseñar un mundo tecnológico disruptivo. Sin embargo, la velocidad de crecimiento de este tipo de redes es mayor que la puesta en escena de algoritmos criptográficos seguros que auguren un futuro prometedor para ellas [146]. De hecho, entre las redes pertenecientes a este paradigma, quizás las más populares y más críticas de manejar son las redes vehiculares, donde los vehículos crean una malla inteligente de nodos interconectados para monitorizar todas las incidencias de tráfico. Desplegar este tipo de redes hoy en día puede parecer una utopía debido a todos los requisitos necesarios que conllevaría adaptar nuevo hardware en los vehículos. Sin embargo, se puede sustituir este alto coste de instalación por la utilización de dispositivos cotidianos como son los teléfonos móviles inteligentes. Este capítulo se centra en introducir el concepto de Internet de las Cosas desde el punto de vista de la seguridad. Se describe la necesidad actual requerida por los elementos de estas redes para que su uso sea factible y aconsejable. Se explica la evolución de la seguridad en este campo para poner en situación el estado real que nos atañe en la actualidad. Nótese que este capítulo no tiene intención de ser un análisis exhaustivo del estado del arte de la seguridad en Internet de las Cosas, sino una introducción generalista e inteligible para adentrar al lector en las motivaciones que preceden a los capítulos posteriores. Además se da una explicación pormenorizada de las redes vehiculares y se detalla alguno de los mayores retos en materia de seguridad a los que hay que hacer frente a corto plazo. Por último, se aportan conceptos básicos sobre tecnologías móviles y sobre cómo pueden ayudar a conseguir un despliegue más rápido de este tipo de redes sin necesidad de un alto coste de instalación.

1.1. Internet de las Cosas

El concepto de Internet de las Cosas o IoT (Internet of Things) se basa en la interconexión digital de objetos físicos para proveer un nuevo tipo de red que permite su gestión eficiente, como muestra la Figura 1.1. Por ejemplo, la Internet de las Cosas potencia objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores y demás, y les permite comunicarse globalmente. Su principal objetivo es mimetizar el mundo real con el mundo virtual para facilitar las tareas cotidianas. Según la empresa Gartner [90], en 2020 habrá en el mundo aproximadamente 26 mil millones de dispositivos bajo el paradigma de Internet de las Cosas. Abi Research [212], por otro lado, asegura que para el mismo año existirán 30 mil millones de dispositivos inalámbricos conectados a Internet.

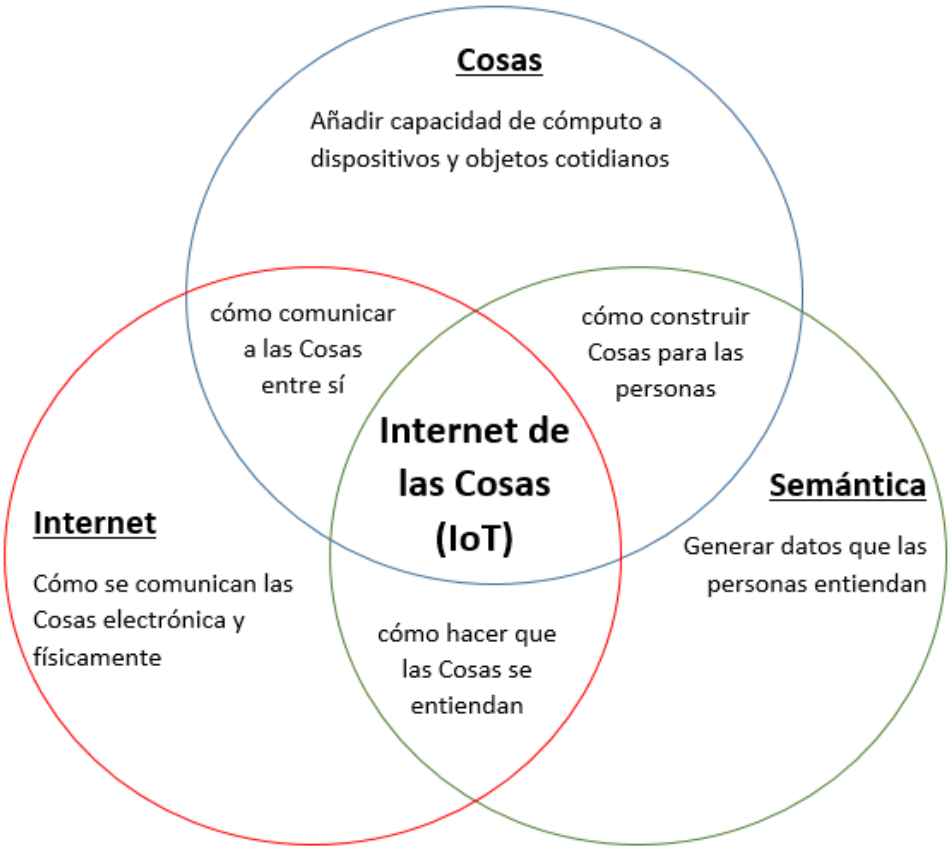

Figura 1.1: Concepto de Internet de las Cosas

La Internet de las Cosas se caracteriza por ser no determinista y abierta, y porque entidades inteligentes auto-organizadas u objetos virtuales son interoperables y capaces de actuar de forma independiente, en función del contexto, las circunstancias o el ambiente [220]. La arquitectura que mejor

define a este tipo de redes es la que está orientada a eventos, y que se construye de abajo hacia arriba, basándose en el contexto de los procesos y las operaciones que tienen lugar en tiempo real. En este nuevo paradigma se genera una ingente cantidad de datos que son aprovechados mediante técnicas de big data para intentar modelar el comportamiento de los objetos interconectados. Además se utilizan técnicas de machine learning y deep learning para predecir dichos comportamientos.

Entre las principales propiedades que definen a la Internet de las Cosas [9], se pueden destacar las siguientes:

- Comunicación y cooperación: Los objetos tienen la capacidad de conectarse a los servicios de Internet y/o entre sí, pudiendo intercambiar y actualizar datos entre ellos, mediante el establecimiento de conexiones descentralizadas.
- Capacidad de direccionamiento: Esta clase de dispositivos pueden ser configurados y localizables desde cualquier lugar de la red.
- Identificación: Los objetos pueden ser identificados mediante tecnologías tales como RFID (Radio Frequency Identification), NFC (Near Field Communication), códigos QR (Quick Response), entre otras muchas.
- Localización: Se puede tener conocimiento sobre la ubicación física de los objetos, pudiendo saber dónde se encuentra en todo momento.
- Actuación: Determinados objetos son capaces de manipular su entorno, creando una nueva dimensión de espacios inteligentes.

Por ello, gracias a la Internet de las Cosas y sus aplicaciones, la evolución tecnológica cotidiana está evolucionando a pasos agigantados [187] y muchas son las posibles aplicaciones que se pueden desarrollar sobre Internet de las Cosas, como muestra la Figura 1.2.

Así, existen sistemas que, por ejemplo, permiten el control de la contaminación del aire mediante la mejora del transporte público, como el proyecto RESCATAME [134], liderado por Libelium, una de las empresas españolas más potentes en soluciones de IoT. El programa, puesto en marcha por el Ayuntamiento de Salamanca, ha permitido la obtención de una gran cantidad de datos del tráfico de la ciudad mediante la colocación de sensores basados en una placa base Waspmote que facilita información sobre temperatura, humedad relativa, monóxido de carbono, dióxido de nitrógeno, ozono y niveles de partículas en el aire. Otro ejemplo similar es la capital de Finlandia, Helsinki, que dispone de un sistema de transporte público en autobús conocido como HelB [103], que permite la recogida de datos a través de sensores colocados en los vehículos. Gracias a su análisis, la administración

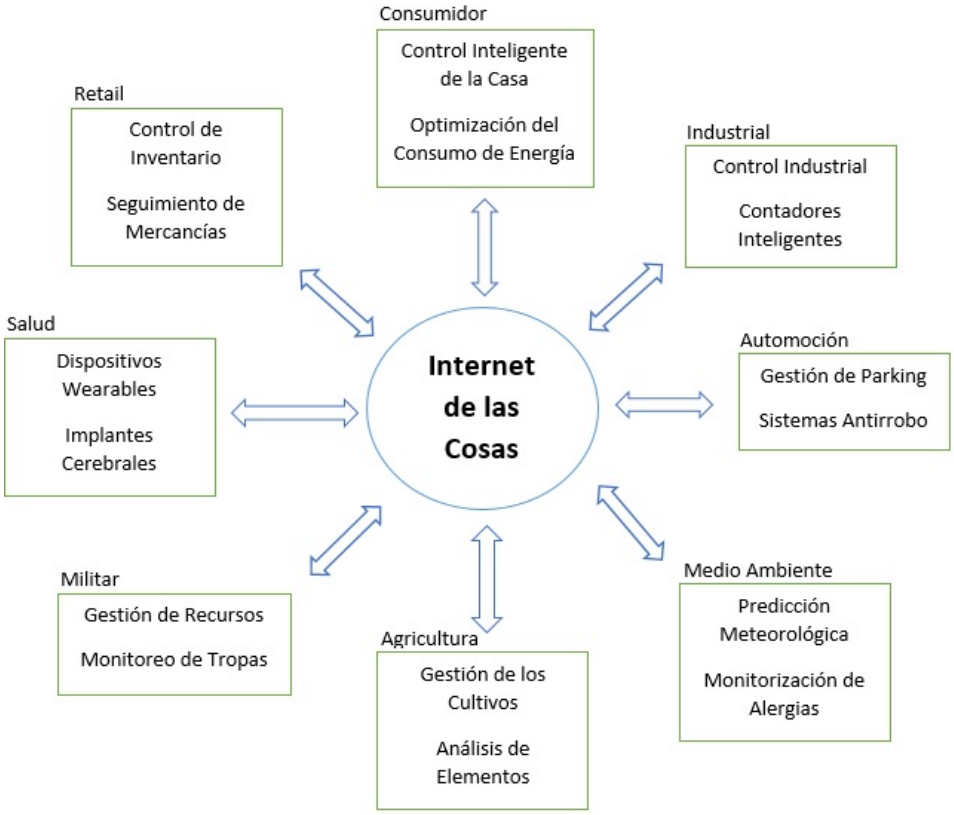

Figura 1.2: Aplicaciones de la Internet de las Cosas

ha conseguido reducir el consumo de combustible, mejorado los trayectos y perfeccionado los métodos de conducción de sus profesionales. Existen aplicaciones que ayudan a alertar de incendios forestales, como la de la empresa valenciana NTForest, que lleva tiempo desarrollando soluciones propias de Internet de las Cosas para el cuidado y la preservación del medio ambiente, especialmente la lucha contra los incendios forestales. En dicho entorno se dispone de unos sensores que se encargan de recoger en tiempo real datos sobre humedad, temperatura relativa, gases en combustión y otros factores ambientales que permiten avanzar cómo evoluciona un fuego y luchar para extinguirlo. El proyecto, conocido como Senticnel [85], utiliza la tecnología WSN (Wireless Sensor Network) para la detección temprana de los incendios. También es común el uso de Internet de las Cosas para la recogida y el análisis de los datos en entornos agrícolas. Esto ha permitido lo que se conoce como la Agricultura Inteligente. Hoy en día existen proyectos como el liderado por la empresa iSoftStone [115], con sede en China, que facilita la instalación de unos sensores con cámaras en las granjas de Hebei, una de las provincias del norte del país. Esos medidores recogen información sobre

temperatura, humedad y precipitaciones en los cultivos. Por otra parte, la gestión de los aparcamientos públicos es un problema que tratan de solventar algunas aplicaciones actuales de Internet de las Cosas. Este problema, relacionado con la gestión eficaz de los aparcamientos, contribuye especialmente al aumento de la polución en las grandes ciudades. La búsqueda de aparcamiento provoca una mayor congestión en las carreteras, un desperdicio de combustible y una pérdida de tiempo. El Proyecto SMART SANTANDER [58] ha distribuido casi 1.200 sensores Waspmote por toda la ciudad, de los que 375 están enfocados a la recogida de información sobre el número de plazas libres de aparcamiento gratuito. Otras grandes ciudades como Barcelona ya han conseguido ahorrar 50 millones al año gracias a este tipo de aplicaciones de Internet de las Cosas. De este estilo existen una gran variedad de aplicaciones que aparecen día a día para aprovechar todas las ventajas que ofrece la IoT. Aplicaciones de ahorro en la iluminación de las grandes ciudades [203], aplicaciones para el mundo del retail [241] [195], aplicaciones para la gestión de la recogida inteligente de basuras [135], y un sinfín de posibilidades que hace más fácil la vida cotidiana.

Sin embargo, muchos de estos proyectos no se preocupan lo suficiente de uno de los factores más importantes para hacer viable y escalable estas aplicaciones: la seguridad. En este nuevo tipos de redes, la seguridad debe ser un elemento que se parametrice y diseñe desde un inicio. Muchas compañías, como Google, están utilizando el principio de "Seguridad por defecto"(o "Secure by default"), en los sistemas operativos que están desarrollando para la Internet de las Cosas, como es el caso de Brillo [98]. De esta manera, la seguridad es el eje de todo el sistema, y sobre ella orbitan el resto de funcionalidades de las aplicaciones desplegadas en estos entornos. Aún así, la seguridad debe ser tenida en cuenta por las propias aplicaciones en su momento de desarrollo, y no delegarlo sólo en el sistema operativo que las engloba.

1.1.1. La Seguridad: Pasado

Hasta ahora la seguridad informática se ha presentado contextualizada en el ámbito de proteger a *los buenos* de *los malos*. En particular, esta visión binaria definía de forma global lo que la gente entendía, y muchos siguen entendiendo a día de hoy, como seguridad en redes. Esta perspectiva está tomando un giro radical en los últimos lustros con el incandescente concepto de Internet de las Cosas. Aunque parezca que todo el paradigma de seguridad debe dar un cambio bastante grande en su contenido para adaptarse a la Internet de las Cosas, el cambio más importante es conceptual [108], en su definición. Se estaba acostumbrado a pensar en la red como una fortaleza cual castillo en la edad media. Los buenos permanecían en el interior mientras los malos esperaban en el exterior su oportunidad para entrar de formas poco ortodoxas. Los elementos constructivos de la fortaleza, como

puertas, paredes y demás, en conjunción con los aguerridos guardianes que vigilaban el tránsito hacia el interior de la fortaleza, intentaban garantizar que sólo las personas autorizadas pudieran entrar en los dominios de la fortaleza. Análogamente sucede lo mismo con Internet y sus componentes, pero con la salvedad de que Internet no es una ciudad medieval estática, sino más bien una ciudad completamente dinámica y extremadamente compleja, con infinidad de límites cambiantes en su interior haciendo el símil con las ciudades medievales. Estos límites entre el interior y el exterior son cada vez más difusos, por lo que continuamente surgen nuevas dudas que el modelo no puede resolver en la Internet de las Cosas. Por ello, el acceso, autorización y relaciones de confianza dentro de Internet son muy complejos.

Paul Simmonds [231] acuñó un término que da explicación a este símil entre la seguridad actual y los castillos medievales. Él lo nombró *deperimeterization*, intentando definir un modelo en el que el administrador simplemente debe comunicar los componentes informáticos que necesite conectar por red, sin preocuparse de la seguridad de estos en dicha red, ya que el sistema estaría configurado de tal forma que cualquier dispositivo conectado a él estaría asegurado sin configuraciones adicionales.

Han sido muchos los trabajos de investigación en los últimos años que han abordado el tema de la seguridad en Internet de las Cosas desde un punto de vista conceptual. En [187] hacen un análisis detallado de los retos y oportunidades abiertas más importantes que tiene el paradigma de Internet de las Cosas. Destacan por encima de todas, la seguridad, indicando que representa el componente más crítico para la adopción de IoT. La confidencialidad de los datos, la autenticación, la privacidad y la fiabilidad son los cuatro pilares sobre los que se debe sustentar cualquier sistema IoT.

La confidencialidad es la propiedad que impide la divulgación de información a usuarios no autorizados, definiendo autorización como el otorgamiento de privilegios a un usuario para realizar una operación. Para conocer la identidad de los elementos, se establecen métodos de control de accesos que aseguran el acceso a la información únicamente a aquellos usuarios que cuenten con la debida autorización. Estos métodos de control de accesos requieren de algoritmos y técnicas de autenticación que permitan demostrar que la identidad de un elemento es la que dice ser. La privacidad es el aspecto de la tecnología de la información que se ocupa de la capacidad que un elemento tiene para determinar qué datos pueden ser compartidos con otros elementos. Por último, la fiabilidad es la capacidad de un elemento de realizar su función de manera adecuada.

Los proyectos de investigación pasados más relevantes en esta materia, son descritos en el trabajo de investigación [187]. Así, en 2008 la American National Science Foundation lanzó el programa Cyber-Physical Systems, que pretendía desplegar la primera red de objetos físicos interconectados de forma segura. En Europa, desde 2005 se están diseñando iniciativas relacionadas

con Internet de las Cosas a través de los programas marco de investigación, antiguos FP7, actuales Horizonte 2020. Proyectos como HYDRA, RUNES, IoT-A o iCORE han puesto en la vanguardia de la Internet de las Cosas al viejo continente. En Asia, países como Japón, siempre han apostado fuerte por el cambio generacional que involucra IoT, y con proyectos como UNS financiados por la estrategia de investigación del marco e-Japan, han creado sistemas ubicuos disruptivos para estos escenarios.

En lo referente a la legislación que involucra el tema de la seguridad en dispositivos inteligentes de Internet de las Cosas, en el trabajo [251] se describe el estado histórico de las leyes auto reguladas necesarias para adecuar correctamente el uso de estas redes. Mediante una serie de escenarios, se proponen las acciones a llevar a cabo para compaginar seguridad, privacidad y legalidad gubernamental.

1.1.2. La Seguridad: Presente

La definición del nuevo concepto de Internet de las Cosas, por sí sola, no sirve para entender cómo debe ser la seguridad en ella. Hay que sumarle, al menos, dos conceptos más. El primero, conocido como *consumerization* [108], es el concepto que mejor define la situación actual de la informática de consumo. Los consumidores tecnológicos han surgido con gran auge gracias al abaratamiento de las tecnologías, con lo que cada día existen más devoradores de gadgets que compran todo tipo de dispositivos específicos como portátiles, tabletas, smartphones o similares para su consumo personal. El gran inconveniente es que las empresas en las que trabajan, que hasta hace poco tiempo les suministraban gadgets configurados bajo las directrices que imponían los administradores de red de las empresas, están perdiendo la partida ya que entonces la seguridad estaba de una u otra forma garantizada, mientras que ahora el enfoque de los empleados es preguntarse para qué quieren portátiles o smartphones adicionales si ya tienen los suyos particulares. Así, el administrador de redes, cada vez más, está perdiendo el control sobre los dispositivos que debe asegurar en la red corporativa. Con el tiempo, esta tendencia de que el mismo gadget sirva para lo personal y lo profesional no hará sino aumentar de forma casi exponencial. Todo este proceso se acelerará gracias al consumismo, y a los nuevos productos que estarán de moda, que saldrán mejor de precio y que serán aún más específicos en sus cometidos, integrándose de forma más natural con el ser humano. A todo esto hay que sumarle que los grandes consumidores tecnológicos están aún por llegar. Estos no son otros que los niños y jóvenes que ahora se pasan horas delante de una videoconsola, o los que ya manejan a las mil maravillas el arte de la escritura en smartphones o tablets. Ellos dentro de poco harán de este concepto una auténtica forma de vida. Es simplemente una recapitulación de lo que sucedió en la década de los 80 y 90 con los ordenadores personales, pero elevado a la máxima potencia, con más dispositivos distintos y mayor

versatilidad en todos los aspectos.

El otro concepto a tener en cuenta es el de *decentralization* [108], que se corresponde con la apoteosis de la computación en la nube. La tendencia actual es almacenar nuestros datos en un nodo central del cual conocemos poco o casi nada. Correos electrónicos, fotos, libros, música, documentos y un largo etcétera, se almacenan cada vez con más frecuencias en plataformas virtuales, a las que tenemos acceso desde nuestros gadgets de última generación simplemente mediante un navegador web o aplicación diseñada para ello. Esto conlleva replantearse si estamos ante el final de largas y duras batallas que se han disputado desde la aparición de los ordenadores personales. ¿Es necesario ya replantearse si vale la pena comprar un dispositivo con mayores prestaciones a nivel de hardware y que contenga como sistema operativo Windows?, o por el contrario ¿dejar de lado la potencia del hardware y apostar por la conjunción entre desarrollo software para un determinado hardware, como hace Apple? ¿Para qué discutir este tema, si con un navegador web podemos tener acceso a infinidad de aplicaciones, como hace Google con su sistema operativo Chrome? De forma análoga, las copias de seguridad personales, los ordenadores llevados al límite en prestaciones, etc. se convierten en aspectos irrelevantes, ya que es suficiente tener un terminal tonto, que es como se le denomina a un ordenador que basa todas sus características en un nodo central que suministre lo que se necesite, sin que se disponga de grandes capacidades de cómputo, sino que sólo sea capaz de mostrar lo que ese nodo central le devuelve.

En la actualidad existen múltiples proyectos de investigación en ejecución que abordan la complejidad de la seguridad en Internet de las Cosas [230]. Uno de los más relevantes es el OWASP Internet of Things Project [53] que está diseñado para ayudar a los fabricantes, desarrolladores y consumidores a comprender mejor los problemas de seguridad asociados a Internet de las Cosas. Butler [55] es un proyecto financiado con fondos del antiguo FP7, cuyo propósito es permitir el desarrollo de aplicaciones seguras e inteligentes por medio de un sistema de información basado en el contexto y la ubicación de los elementos. El proyecto presentado en [121] propone un Sistema de Detección de Intrusos o IDS (Intrusion Detection System) para los sistemas de Internet de las Cosas utilizando el protocolo IPv6 a través de dispositivos de baja potencia para redes de área personal (6LoWPAN). Además es cada vez más frecuente ver proyectos coordinados por varios continentes, como la colaboración entre Europa y Asia (a través de China y Corea) para el diseño de una arquitectura de Internet de las Cosas dentro del proyecto FIRE (Future Internet Research and Experimentation) [56] [57], que tiene como objeto la búsqueda de soluciones para el despliegue de tecnologías en Internet de las Cosas (por ejemplo, para la seguridad pública, para la seguridad social, para servicios médicos y de salud, para la gestión urbana, etc.), con especial atención a la seguridad de la información, la privacidad y

los derechos de propiedad intelectual.

En lo referente a estandarización [187], en los últimos años se han conseguido estándares como el RF-layer y la tecnología NFC en diversos órganos (ISO 18092, 21481, 22536 y 23917; ECMA 340, 352, 356 y 365; ETSI TS 102 190). Además, ECMA 340/352 e ISO 18092/21481 describen las versiones 1 y 2 de NFC. Los estándares ECMA 356/362 y las normas ISO 22536/23917 también se centran en NFC. Además, otras asociaciones como GSMA han establecido grupos de trabajos en NFC desde 2006, para diseñar las directrices necesarias para que el NFC pudiera introducirse en teléfonos móviles. En el campo de la tecnología RFID, la solución más adoptada es el EPC (Electronic Product Code). La especificación EPC constituye un estándar abierto y de libre acceso, basado en el trabajo realizado en la última década en el MIT. Otros estándares han ido apareciendo como el ONS (Object Naming Service) que representa un mecanismo para leer información sobre un objeto determinado a partir de su EPC. En cuanto a las comunicaciones entre los objetos inteligentes, se ha estandarizado la comunicación de las capas inferiores (PHY y MAC), permitiendo la definición de especificaciones como la 802.15.4, utilizadas por tecnologías como ZigBee. Más recientemente, la atención se ha centrado en las comunicaciones inalámbricas ópticas, dentro del estándar 802.15.7. Respecto a las comunicaciones en las capas superiores, ETSI posee un comité técnico para estandarizar las comunicaciones M2M (Machine to Machine). Por último, remarcar que la W3C tiene un grupo de trabajo especializado en Redes de Sensores Semánticas para estandarizar la forma de dar inteligencia a los objetos.

1.1.3. La Seguridad: Futuro

Los conceptos mencionados anteriormente son insuficientes para definir la complejidad de las redes futuras. En los próximos años surgirán nuevos cambios conceptuales para intentar adaptarse a las nuevas situaciones. Tres de estos conceptos (ver Figura 1.3) a tener en cuenta, se comentan a continuación.

El primer concepto, que Bruce Schneier llama *deconcentration* [108], se refiere a que poco a poco, nuestros queridos ordenadores de propósito general se están muriendo, siendo reemplazados por dispositivos de propósito específico. No en vano, en 10 años, la mayoría de los equipos serán dispositivos pequeños y especializados, ubicados en cualquier lugar imaginable. Por ejemplo, visitar una página web como la Wikipedia era hasta ahora una labor que se realizaba con un navegador web. Actualmente están surgiendo aplicaciones específicas para gadgets específicos, que permiten acceder al contenido de esa página web, sin necesidad de usar un navegador web convencional.

El segundo concepto que está surgiendo, Schneier lo denomina *decustomerization* [108] y se basa en la idea de que se está rompiendo la relación comercial entre las empresas de TIC (Tecnologías de la Información y Co-

municación) y sus usuarios.

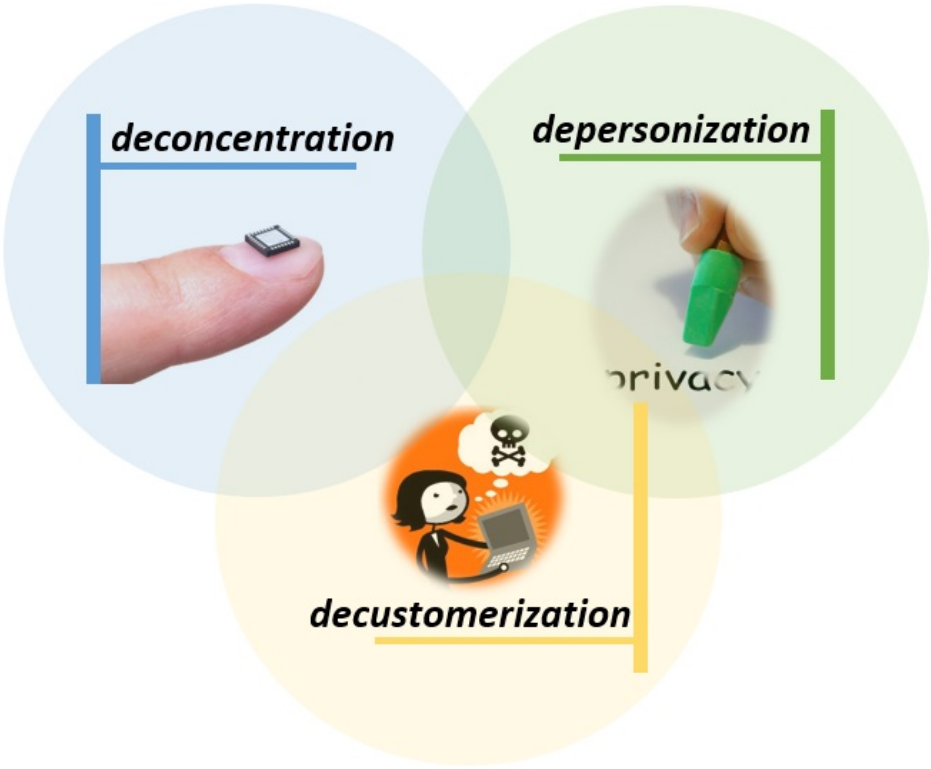

Figura 1.3: Cambios Conceptuales con la Aparición de IoT

Por ejemplo, los motores de búsqueda regalan los servicios a cambio de que aparezca publicidad en las búsquedas que realiza un usuario, de forma que a este le sale gratis realizar una búsqueda a cambio de visualizar anuncios en dicha búsqueda. Esto se extrapola a cuentas de correo electrónico, redes sociales, aplicaciones para smartphones, etc., conllevando que estén diseñadas con el fin de ser meras plataformas publicitarias. Además, en breve veremos cómo se dirige esta tendencia hacia el hardware. Regalarán tablets y notebooks, a cambio de publicidad, o smartphones que contengan una marca de agua con publicidad en las fotos que saquen y se publiquen en las redes sociales. Por supuesto siempre habrá excepciones que cumplan la regla, y no todo el hardware será gratuito o subvencionado, como lo están siendo ahora los teléfonos móviles gracias a las compañías telefónicas. Si hacemos autocrítica de esta situación, poniendo como ejemplo lo que Google está haciendo con sus servicios, podemos darnos cuenta de que no somos el cliente de Google, sino que nos hemos convertido en el producto que Google usa para vender el negocio de la publicidad a sus clientes. Es una relación comercial a tres bandas: nosotros (los consumidores de los servicios), el proveedor de

dichos servicios, y el comprador o anunciante de los datos que se difunden a través de los servicios. Destacable es el caso de Facebook que juega con la privacidad de los datos que maneja, para satisfacer a sus verdaderos clientes, que son los anunciantes y consumidores de datos.

Un tercer concepto en plena fase de nacimiento, es el de *depersonization* [108], y su explicación en muy simple. La idea no es otra que la de pensar o imaginar que el usuario informático convencional está siendo eliminado de forma parcial o total. Cada vez existen más agentes inteligentes que facilitan la labor de los usuarios, por ejemplo, filtrando y gestionando los correos electrónicos que le llegan, tomando como pautas sus preferencias, dejando en un segundo plano la propia decisión del usuario. Otra técnica de este estilo es la de la publicidad personalizada, que tan de moda está últimamente. Aunque dichos agentes inteligentes no sean lo suficientemente sofisticados como para conseguir los resultados que se quieren, de una manera u otra cumplen con lo que las compañías que se anuncian piden.

El punto de inflexión de toda esta evolución no es otro que la reducción de precios en las tecnologías. La conexión de objetos a Internet pronto será lo suficientemente barata como para ser viable. En breve tendremos numerosos objetos conectados, como los dispositivos médicos, las redes inteligentes de energía, los teléfonos inteligentes o smartphones o los automóviles. Con la aparición de la Internet de las Cosas, muchas acciones que hasta ahora realizaba de forma personal el ser humano, pasarán a ser labor de los objetos inteligentes. Los electrodomésticos inteligentes se comunicarán directamente con la compañía eléctrica para gestionar un consumo más eficiente, el coche inteligente interactuará con los sensores de carretera y eventualmente, con otros coches para mejorar la seguridad vial y el confort, la ropa se comunicará con la tintorería, el teléfono hablará con las máquinas expendedoras para realizar las peticiones de producto y los pagos, la nevera se comunicará con el supermercado para pedir los productos que falten, etc. Algunas de estas situaciones ya suceden en algunos países. Las posibilidades que estos objetos inteligentes interconectados ofrecen son difíciles de imaginar. En resumen [107], se unirán una vieja tendencia: *deperimeterization* con dos tendencias actuales: *consumerization* y *decentralization* y tres tendencias futuras: *deconcentration*, *decustomerization* y *depersonization*. En un futuro no muy lejano, una década a lo sumo, posiblemente se perderá el control sobre la información personal ya que no se podrá asegurar lo que estarán haciendo las cosas sin nuestro conocimiento y consentimiento. Por tanto, la seguridad se convertirá en un asunto muy delicado y extremadamente necesario de abordar. No estará orientado tanto al modelo de Paul Simmonds [231] que se explicó antes, sino más encaminado a proteger a los usuarios de los modelos de negocios empresariales.

El paradigma de la *deperimeterization* asume que no se puede confiar en nadie hasta que se demuestre lo contrario, mientras que la *consumeriza-*

tion asume que los dispositivos de los usuarios en las redes no son confiables hasta que se demuestre lo contrario. Por otra parte, la *decentralization* y la *deconcentration* no van a tener éxito si alguien es capaz de hackear los dispositivos para ejecutar software no autorizado o acceder a datos no autorizados. Tampoco la *decustomerization* será viable a menos que se esté dispuesto a aguantar cantidad de anuncios, o cualquier otro modelo que el proveedor utilice para obtener beneficios económicos. Finalmente la *depersonization* requiere que el usuario se fíe de la autonomía total que adquirirán los objetos inteligentes.

Bruce Schneier apuntilla que en la década de 2020, en menos de 5 años, según predice la Ley de Moore los ordenadores serán 100 veces más potentes. Eso cambiará las cosas de una manera que no podemos ahora predecir, pero sí sabemos que la naturaleza humana nunca cambiará. Cory Doctorow [70] ha señalado con razón que todos los ecosistemas complejos tienen parásitos. Los mayores parásitos tradicionales de la sociedad son los criminales. En esta nueva Internet, los usuarios perderán el control de los sistemas TIC y los proveedores se harán con ese control para sus propios fines, por lo que esa definición de parásito cambiará perceptiblemente. Ya sean delincuentes que traten de dejar a cero su cuenta bancaria, piratas cinematográficos tratando de eludir las políticas de piratería multimedia, usuarios de Facebook que traten de utilizar la red social sin renunciar a su privacidad o se vean obligados a ver anuncios, etc., de una manera u otra, los parásitos seguirán tratando de beneficiarse de los sistemas informáticos sin atenerse a las reglas del juego. Estos parásitos y otros mucho más peligrosos van a existir, como siempre han existido, por lo que se deberá prestar especial atención a la seguridad en este paradigma de la Internet de las Cosas, que crece a un ritmo muy lento comparado con el ritmo de aceleración que siguen las tecnologías de la información. En este nuevo mundo, las empresas se preocuparán de la seguridad legal para proteger sus modelos de negocios que es lo que les afecta. Así, si no eres un usuario de su modelo, definitivamente no estarás seguro.

1.1.4. Retos

Internet de las Cosas es un concepto que abarca muchos campos. Por ello, para analizar de forma más detallada los retos de seguridad de la Internet de las Cosas, es necesario estructurar de alguna manera el concepto de Internet de las Cosas. La EASST (European Association of Software Science and Technology) ha estructurado los conceptos que rodean a la IoT [178] de una manera lo más global posible, dividiendo en los siguientes tópicos generales el concepto amplio de Internet de las Cosas:

- **Comunicaciones**: Para permitir el intercambio de información entre los objetos inteligentes.

- **Sensores**: Para capturar y representar el mundo físico en el mundo digital.
- **Actuadores**: Para llevar a cabo acciones en el mundo físico a raíz de órdenes realizadas desde el mundo digital.
- **Almacenamiento**: Para tener un medio donde almacenar los datos obtenidos por los sensores, por los sistemas de localización y por el proceso de identificación.
- **Dispositivos**: Los objetos inteligentes para la interacción con los seres humanos en el mundo físico.
- **Procesamiento**: Para la extracción de datos y los servicios o aplicaciones.
- **Localización**: Para determinar la ubicación del dispositivo en el mundo físico.
- **Identificación**: Para representar de forma unívoca un objeto inteligente en el mundo digital.

La EASST va más allá y propone una tabla identificando la sensibilidad de las características de seguridad en la Internet de las Cosas con cada uno de esos tópicos, proporcionando así una serie de retos a solventar. De esta manera se detalla cómo están en cada campo las características que definen la seguridad de la IoT [178] [120]:

- **Comunicaciones**: Las investigaciones en este campo han llegado a propuestas de soluciones que proporcionan integridad, autenticidad y confidencialidad. Las necesidades de privacidad han sido abordadas en diferentes esquemas de encaminamiento, pero no se ha generalizado lo suficiente. Una cuestión que aún sigue abierta, a pesar de las investigaciones llevadas a cabo en este aspecto, es cómo asegurar disponibilidad frente a ataques de tipo de denegación de servicios o DoS (Denegation of Service).
- **Sensores**: La protección de la integridad y autenticidad de los datos de los sensores es un objetivo de la investigación actual que puede ser solucionado mediante marcas de agua. Por otra parte, la confidencialidad de los datos de los sensores es un requisito bastante débil ya que un atacante puede simplemente colocar su propio sensor físicamente cerca de otro y obtener los mismos valores. Por tanto, la necesidad de confidencialidad se centra en la comunicación mientras que la protección de la privacidad de los sensores se dirige principalmente al mundo físico que se está detectando. Mecanismos como poner las caras borrosas en los vídeos, pueden ser utilizados para preservar la privacidad.

La disponibilidad de los sensores depende sobre todo de la infraestructura de comunicaciones que tengan. Las regulaciones o leyes para definir estándares son necesarias para preservar la privacidad de las personas, ya que éstas deben ser conscientes en todo momento de que existen sensores donde se encuentran, por ejemplo mediante carteles que lo anuncien, como los bares lo hacen ya con sus sensores de videos o cámaras de vigilancia.

- **Actuadores**: La integridad, autenticidad y confidencialidad de los datos enviados por los actuadores, sobre todo dependen de la seguridad de las comunicaciones. Por tanto, la baja sensibilidad del propio actuador se hace necesaria. Además se debe garantizar que un atacante no pueda controlar ningún actuador, y la privacidad depende mucho del escenario donde se ubique el actuador. Las leyes y estándares al respecto son muy similares a las referidas a los sensores, ya que se debe asegurar la no perturbación de la intimidad a través de su uso.

- **Almacenamiento**: Los mecanismos de seguridad para los dispositivos de almacenamiento en la Internet de las Cosas están bien establecidos [178], pero su empleo sigue siendo bastante ineficiente. Los estándares deben asegurar la protección de la privacidad del usuario. La disponibilidad de almacenamiento depende de la disponibilidad de la infraestructura de comunicaciones. Existen mecanismos bastante eficaces para controlar la redundancia de los datos almacenados.

- **Dispositivos**: En el ámbito de la integridad de los dispositivos inteligentes es necesario que estén libres de malware, propiedad llamada admisibilidad. Garantizar esta admisibilidad es un tema abierto actualmente en el mundo de la investigación de las plataformas seguras en informática o TPC (Trusted Platform Computing). La confidencialidad de un dispositivo está unida a su integridad, para garantizar que ningún tercero tenga acceso a los datos internos de los dispositivos. La privacidad de los dispositivos depende de la intimidad física y la privacidad de las comunicaciones. La disponibilidad de un dispositivo depende de la integridad y fiabilidad de los dispositivos, y de la disponibilidad de la parte de las comunicaciones que se encargan de conectar el propio dispositivo.

- **Procesamiento**: La integridad del procesamiento se basa en la integridad de los dispositivos y de las comunicaciones. Por otra parte, depende del correcto diseño e implementación de los algoritmos. El procesamiento también depende de las acciones que se lleven a cabo por los actuadores. La autenticidad no es sensible ya que depende en exclusiva de la autenticidad de las comunicaciones y de los dispositivos. La confidencialidad en el procesamiento sólo depende de la integridad

del dispositivo, y en caso de procesamiento distribuido, depende de la integridad de la comunicación. La privacidad protege contra la amenaza de la minería de datos, pero las regulaciones deben ser empleadas para asegurarse de que se aplican correctamente. La disponibilidad de procesamiento depende sólo de la disponibilidad de los dispositivos y de la comunicación.

- **Localización**: La integridad de la localización se basa especialmente en la integridad de la comunicación. Además, es necesario garantizar la integridad de las señales de referencia utilizadas en la localización (GSM, GPS, etc.). Del mismo modo, la autenticidad de la localización depende de la autenticidad y la integridad de la comunicaciones. La confidencialidad y la privacidad de los datos de localización son de gran importancia para garantizar la privacidad del usuario. La confidencialidad en este contexto significa que un atacante no debe ser capaz de revelar los datos de localización y por lo tanto se basa principalmente en la confidencialidad de la comunicación. La privacidad de los datos de localización significa que no hay manera de que un atacante revele la identidad de la persona u objeto de los datos de localización y que la localización y seguimiento no es posible sin el acuerdo explícito o conocimiento. La disponibilidad de localización es importante asegurarla para que los sistemas de localización sean robustos y que no puedan ser manipulados por un atacante. Las regulaciones en la localización y seguimiento son de gran importancia sobre todo en términos de privacidad, como se mencionó anteriormente.

- **Identificación**: Para la identificación se tiene principalmente la misma sensibilidad que para la localización. Sin embargo es más fácil para un atacante manipular el proceso de identificación, que manipular el proceso de localización. Este resultado se debe principalmente a que la tecnología utilizada (por ejemplo, RFID o biometría), es más factible que un atacante pueda manipularla que a las tecnologías de localización (por ejemplo, GSM).

Como conclusión, los puntos flacos o sensibilidades más notables de los tópicos anteriores de la Internet de las Cosas respecto a las características de seguridad, se muestran en la tabla 1.1, que describe cómo está el estado del arte en cada situación y cuáles son los mayores retos a abarcar. En verde se pueden observar los tópicos y propiedades que están siendo ampliamente investigados, en amarillo los que aún necesitan más labor de investigación para ser seguros, y en rojo se observan las características más necesitadas actualmente y que requieren de soluciones inmediatas.

Por otro lado, en [187] discuten acerca de los principales retos de seguridad que deben abordarse para que Internet de las Cosas se convierta en una de las principales corrientes tecnológicas. En particular, identifican tres

Tabla 1.1: Retos en la Seguridad de Internet de las Cosas

	Propiedades de Seguridad				
Características IoT	**Integridad**	**Autenticidad**	**Confidencialidad**	**Privacidad**	**Disponibilidad**
Comunicaciones					
Sensores					
Actuadores					
Almacenamiento					
Terminales					
Procesamiento					
Localización					
Identificación					

cuestiones claves que requieren enfoques innovadores: la confidencialidad, la privacidad y la confianza. En lo referente a la confidencialidad, es necesario definir mecanismos adecuados para el control de acceso a los datos generados por los dispositivos IoT, así como definir un lenguaje de programación adecuado que permita a las aplicaciones consultar de forma eficiente la información generada por los dispositivos IoT y definir un sistema de gestión de identidades para los objetos inteligentes. En cuanto a la privacidad, los mayores retos se centran en la definición de un modelo genérico para establecer un paradigma de privacidad en IoT, en el desarrollo de técnicas innovadoras para que los escenarios de IoT puedan escalar y adaptar su heterogeneidad, y en el desarrollo de soluciones que permitan mantener el anonimato en materia de localización e identificación. Por último, para conseguir que un esquema sea confiable es necesario definir un lenguaje sencillo para gestionar la confianza en base a la interoperabilidad semántica de los objetos, definir un mecanismo de confianza basado en el control de acceso, desarrollar un sistema gestor de identidades y diseñar un marco de confianza genérico y flexible.

De esa manera, los principales retos de investigación [145] en materia de Seguridad para Internet de las Cosas que están abiertos y vigentes a fecha de hoy en cuanto a las comunicaciones [100], pueden resumirse como:

- En lo referente a las Capas de Comunicación a nivel Físico y de MAC: A pesar de la madurez del estándar IEEE 802.15.4, existen diversas limitaciones con respecto a cómo se implementan los servicios de seguridad compatibles con la capa MAC [112]. Se hace necesario el diseño de mecanismos de gestión de claves que soporten los mecanismos de seguridad punto a punto en las capas superiores, eludiendo así las limitaciones de la gestión de las Listas de Control de Acceso en la capa de enlace. La seguridad en entornos de comunicación limitados a través de la capa de enlace, requiere una mejora en cuanto al tiempo empleado por el estándar IEEE 802.15.4e.

- En lo referente a la Capa de Red: Es necesario un mayor número de propuestas para asegurar la confidencialidad, la integridad, la autenti-

cación y el no repudio [100] así como nuevas e innovadoras propuestas contra los ataques de fragmentación de paquetes [125] [111]. La gestión de claves es un área que aún tiene muchos retos abiertos que deben ser solucionados [218].

- En lo referente al Enrutamiento: Existen grandes limitaciones actuales en los controles de seguridad de los protocolos de enrutamiento RPL [243]. Se necesita, por otra parte, identificar los modelos de amenazas para estos protocolos [243]. También se deben desarrollar nuevas propuestas para crear soluciones contra ataques internos a estos protocolos [243].

- En lo referente a la Capa de Aplicación: Existen grandes limitaciones de seguridad en los protocolos CoAP (Constrained Application Protocol). La gestión de claves en esta capa también debe ser revisada [123]. Deben aparecer nuevas propuestas para modificar protocolos como DTLS [106] que permitan delegar las operaciones costosas a otros procesos o protocolos menos críticos. En esta capa es necesario crear soluciones y propuestas innovadoras para el soporte de certificados digitales y esquemas de cifrado asimétrico más eficientes para el ámbito de Internet de las Cosas.

1.2. Redes Vehiculares

Las aplicaciones de Internet de las Cosas que son más mediáticas suelen ser las que tienen un corte más orientado al consumidor, que son al fin y al cabo las que le darán éxitos o fracasos. Sin embargo, en la mayoría de casos dichas aplicaciones resultan poco escalables a nivel industrial. Por tanto, lo complicado en estos momentos es saber si su implementación se extenderá a sectores más amplios y si será capaz de redefinir sus procesos para crear eficiencia y valor perdurable. Entre los sectores que han apostado desde un principio por la Internet de las Cosas, destacan por encima de todos los de la logística y transporte gracias a los campos de innovación que se abren en estos sectores. En esta sección se explica uno de los paradigmas más emergente en estos sectores, las redes vehiculares (ver Figura 1.4). Con las redes vehiculares se pretende crear redes inteligentes entre vehículos para gestionar de forma más eficiente todo lo que rodea al tráfico vial. Además, la aparición de vehículos autónomos está ayudando a que a la población actual les suene cada día más normal hablar de redes vehiculares. Esta sección se centra en introducir diversas cuestiones de seguridad necesarias para este tipo de redes, caracterizadas por su idiosincrasia móvil.

Este tipo de redes llevan investigándose desde hace varias décadas mediante algunas redes precursoras [141] y su mayor reto siempre ha sido el despliegue en entornos reales. El rendimiento del enrutamiento de las comu-

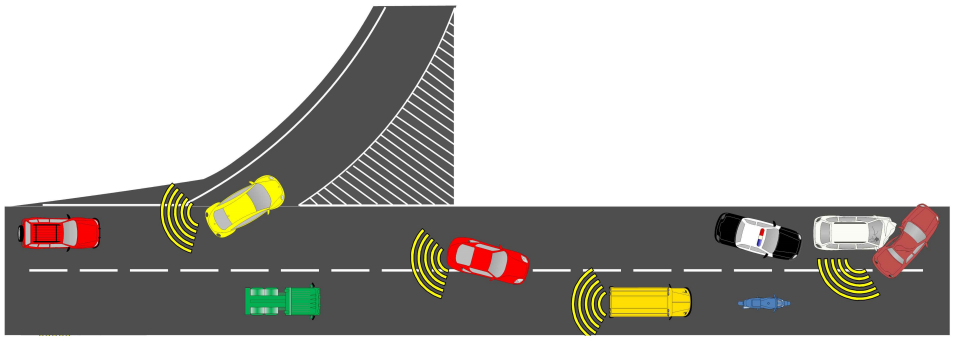

Figura 1.4: Red Vehicular

nicaciones, la adopción de un protocolo único, los frameworks de simulación y la capacidad de configuración han sido otros de los muchos retos asociados históricamente a este tipo de redes.

1.2.1. El Origen: Las Redes Móviles

Antes de entrar en detalle a hablar acerca de las redes vehiculares, se explica brevemente la red matriz donde se engloban las redes vehiculares: las redes móviles.

Las redes móviles ad-hoc o MANETs (Mobile Ad-hoc NETworks) son un tipo de redes ad-hoc inalámbricas cuyos dispositivos poseen propiedades de auto-configuración, además de poseer cierta movilidad, pues se encuentran montados en plataformas móviles [253]. Cada dispositivo en una MANET posee libertad para desplazarse independientemente en cualquier dirección, y eso permite que cambien dinámicamente las condiciones de enlace entre los dispositivos o nodos [161]. Esto significa que los enlaces entre los nodos pueden cambiar con el tiempo, que nuevos nodos pueden unirse a la red y que los nodos de la red pueden abandonarla en cualquier momento [69].

El alcance de una MANET es mayor que el radio de una única antena inalámbrica. Para ello es necesario enrutar el tráfico a través de saltos entre nodos intermedios de forma que dos nodos cualesquiera que quieran comunicarse puedan hacerlo. En las MANETs no existe ningún enrutador en una posición fija, como ocurre por ejemplo en las redes de telefonía móvil, que consisten en un cable central al cual se conecta una estación base, de forma que los nodos móviles solo pueden comunicarse mediante un salto inalámbrico a la estación base, y múltiples saltos inalámbricos no son posibles.

Uno de los principales retos a la hora de construir una MANET es lograr que sea posible equipar cada dispositivo para mantener continuamente la información necesaria para enrutar los mensajes. Este tipo de redes puede operar de forma autónoma o bien estar conectada a Internet. La principal

característica de las MANETs es que todos los dispositivos que forman parte de la red, además de funcionar como terminales finales, realizan también funciones de retransmisión de paquetes típicamente asociadas a routers en redes clásicas. Esta cualidad permite enrutar paquetes a destinos sin cobertura directa, a través de otros nodos intermedios que se encuentren en la red. De este modo es posible incrementar la movilidad y el tamaño de la red, creando de una manera rápida y eficaz una red temporal en lugares carentes de una infraestructura de red.

1.2.2. Definiciones

Tras la aparición de las redes móviles ad-hoc, surgieron varias versiones entre las que destacan las redes vehiculares o VANETs (Vehicular Ad-Hoc NETworks). Estas redes se caracterizan por utilizar vehículos en movimiento como nodos y gestionar comunicaciones inteligentes entre ellos. Su principal finalidad es prevenir las circunstancias adversas de tráfico que acaecen normalmente en las carreteras. Además se logra obtener una gestión más eficiente del tráfico mediante la utilización de este nuevo paradigma vial.

Al ser una extensión de las MANETs puede parecer que sus requerimientos son casi idénticos, pero las VANETs poseen ciertos detalles que las hacen sustancialmente diferentes. Los nodos de las VANETs tienden a moverse de forma organizada y no aleatoriamente, como se presupone en las MANETs, debido a que son vehículos que siguen un trazado marcado por las carreteras. En la definición clásica se supone que para que los vehículos puedan formar parte de la red, deben tener hardware instalado que gestione las comunicaciones. Por tanto este equipamiento debe ser añadido a todos los vehículos. Además, según esa concepción estas redes se caracterizan por tener, en su mayoría, cierto equipamiento adicional que se debe instalar en las carreteras para lograr una comunicación más eficiente con los vehículos e informarles de los eventos que ocurren. Esto implica tener que adecuar las infraestructuras de las carreteras, con el coste económico que ello supondría. Por tanto, las redes vehiculares están conformadas por dos tipos de nodos: las unidades de a bordo y las unidades de carretera. Las unidades de a bordo, también conocidas como OBUs (On-Board Units), son los dispositivos que se colocan en el vehículo para que estos puedan comunicarse e interactuar con el resto de elementos de la red. Las unidades de carretera, también conocidas como RSUs (Road-Side Units) son los dispositivos estáticos que se sitúan a lo largo de la red de carreteras para dotar de inteligencia al trazado vial. Teniendo en cuenta los dos posibles tipos de dispositivos que pueden encontrarse en una red vehicular, existen dos grandes tipos de comunicaciones que caracterizan a una VANET:

- Comunicación Vehículo a Vehículo o V2V (Vehicle to Vehicle): Es la comunicación que se realiza entre las diferentes OBUs que identifican

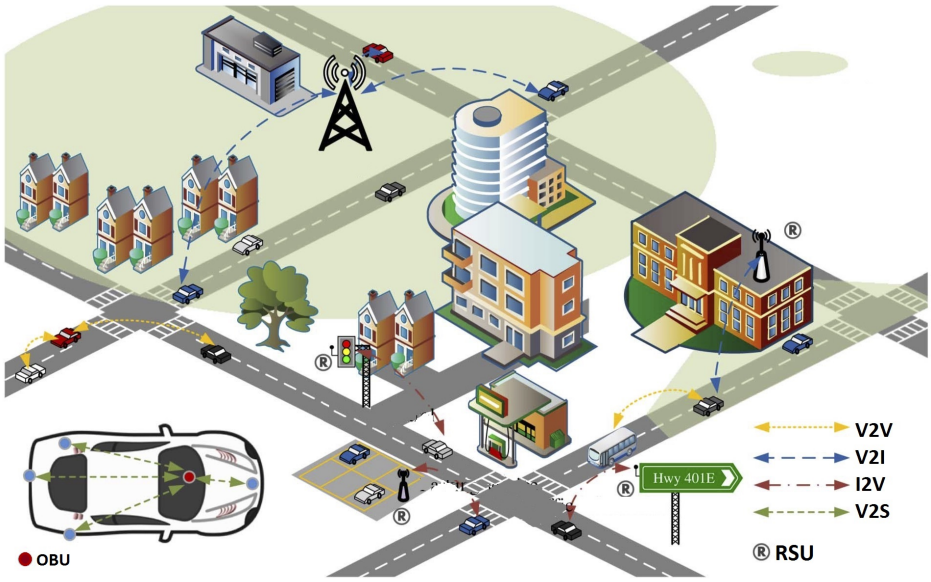

Figura 1.5: Tipos de Conexiones en una Red Vehicular

a los vehículos.

- Comunicaciones Vehículo a Infraestructura o V2I (Vehicle to Infrastructure) e Infraestructura a Vehículo o I2V (Infrastructure to Vehicle): Es la comunicación que se realiza entre las OBUs en los vehículos y las RSUs en las infraestructuras de carretera.

Además puede haber diferentes plataformas de sensores dentro de un vehículo, propiciando así las comunicaciones Vehículo a Sensor o V2S (Vehicle to Sensor) que son las que conectan todas las plataformas de sensores dentro de un vehículo. En la Figura 1.5 se pueden visualizar todos estos conceptos.

Por tanto y para resumir, una red vehicular se caracteriza por:

- Una topología dinámica en la que los vehículos alcanzan velocidades que rondan en media los 50 km/h en entornos urbanos y hasta 120 km/h en autopistas. Además los movimientos de los vehículos se pueden realizar en distintas direcciones conllevando que puedan conectarse y desconectarse a la red vehicular en periodos de tiempo muy cortos. Todo esto provoca frecuentes y rápidos cambios en la topología de red.

- Una frecuente desconexión en las comunicaciones provocadas por la topología de la red. Dicha circunstancia se materializa en que una transmisión de información entre dos nodos pueda quedar incompleta.

- Un tipo de comunicación basado en proximidad. Normalmente para conectarse con un nodo en una red sólo se debe conocer su identificador. En una red vehicular, los nodos sólo son alcanzables si están geográficamente en posiciones cercanas, ya que las comunicaciones son punto a punto de forma inalámbrica y descentralizada.

- Una movilidad restringida a pesar de poseer una topología dinámica. Las carreteras están perfectamente delimitadas y definen el grado de movilidad de los vehículos. Por ello, se pueden predecir las situaciones futuras de los vehículos en base al trazado delimitado.

- Un modelo de propagación complejo. Una red vehicular puede operar en tres grandes escenarios: Una autopista, un entorno rural o dentro de la ciudad. En una autopista, el modelo de propagación de las comunicaciones inalámbricas se establece al aire libre, pero la potencia de la señal inalámbrica puede sufrir interferencias por los elementos viales que existen a los lados de la autopista. En una ciudad, la propagación de las señales inalámbricas tiene mucho menos recorrido debido a la densidad de tráfico, a la presencia de edificios, árboles y otros objetos que actúan como obstáculos en las comunicaciones. En un entorno rural, con complejas arquitecturas topológicas en forma de campos, bosques densos, montañas, etc., es importante tener en cuenta la reflexión y atenuación de la señal. Además en estos tres escenarios hay que sumar el efecto de las interferencias de otro tipo de comunicaciones inalámbricas como pueden ser redes de telefonía móvil, redes de conexión a Internet inalámbricas, entre otras muchas.

Todas estas características traen nuevos desafíos para el diseño de protocolos de comunicación en VANETs. Las limitaciones espacio-temporales de este tipo de redes y la heterogeneidad de los vehículos en términos de velocidad y movilidad, son los principales factores de diseño que deben considerarse al desarrollar nuevos algoritmos y protocolos para las redes vehiculares [62].

Las comunicaciones en las redes vehiculares vienen definidas por el estándar WAVE (Wireless Access in Vehicular Environment) y permiten usar tecnologías tan variopintas como RFID, Bluetooth, Wi-Fi, 3G, LTE o el llamado DSRC (Dedicated Short-Range Communications) específicamente diseñado para las comunicaciones de corto alcance en automóviles, y que define el método por el cual los vehículos intercambian regularmente información con otros vehículos cercanos mediante mensajes cortos que contienen su ubicación y velocidad.

1.2.3. Aplicaciones

La finalidad del despliegue de una VANET en un entorno real viene incentivada por la posibilidad de generar un sistema inteligente que supervise

diferentes tipos de datos, tales como las condiciones de los vehículos, el estado de las carreteras, el flujo real del tráfico o las condiciones atmosféricas del trazado (ver Figura 1.6). De esta manera se lograría que la infraestructura vial fuese mucho más segura y eficiente y se permitiría que los vehículos se comunicasen entre ellos para crear un ecosistema único.

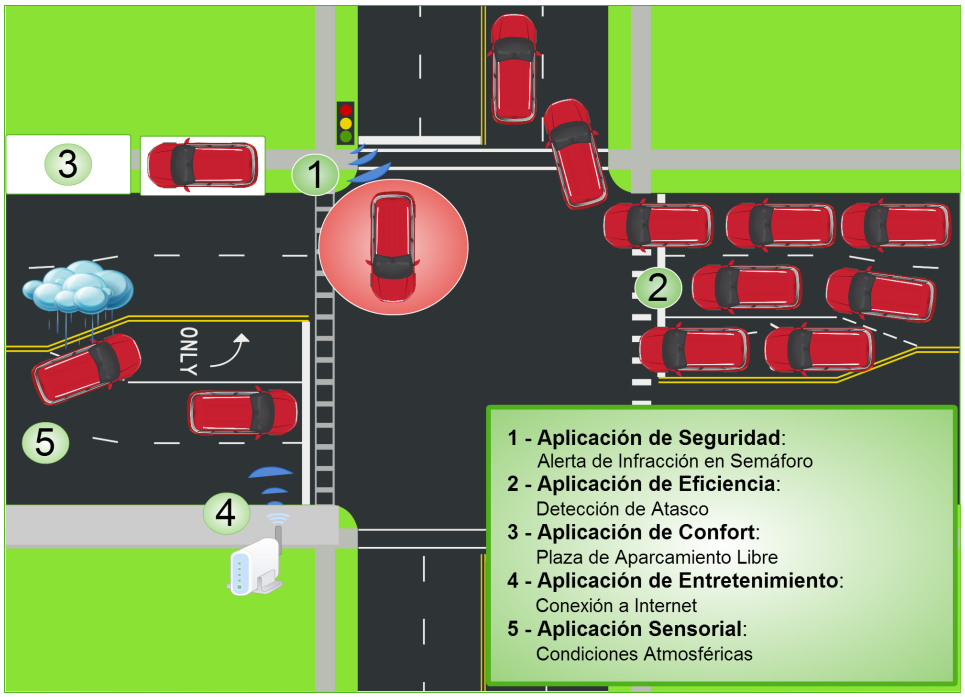

Figura 1.6: Ejemplos de Aplicaciones de las Redes Vehiculares

Así, las principales aplicaciones de una red vehicular se muestran en la Tabla 1.2 y se detallan a continuación:

- Aplicaciones de seguridad. La razón principal de este tipo de aplicaciones es la reducción del número de accidentes de tráfico. Estas aplicaciones no deben tener retraso en las comunicaciones por lo que las tecnologías se utilizan para las conexiones directas vehículo a vehículo, mayoritariamente. Otro aspecto importante es la fiabilidad, por lo que todos los vehículos cercanos al peligro deben de alertar o corroborar el evento en cuestión. Una aplicación de este tipo puede ser para avisar de una colisión cercana y alertar a los vehículos que vayan en dirección al lugar del accidente. Cuando la colisión ocurre, los vehículos afectados generan una notificación que llega al centro de emergencias para comenzar las gestiones oportunas. De hecho ya existen aplicaciones que implementan este sistema de notificación, como el sistema OnStar de General Motors [191]. Las funcionalidades de este tipo de aplicaciones pueden variar y pueden incluir desde la grabación del vídeo para ser

Tabla 1.2: Categorización de las Aplicaciones de las Redes Vehiculares

Categoría	Arquitectura	Tecnología de Comunicación	Ejemplos
Seguridad	V2V - V2I	DSRC/RFID/ Bluetooth/Wi-Fi	Alerta de colisión, Intersección cercana, Alerta de peatones, Alerta de ciclistas o motoristas...
Eficiencia	V2V - V2I	DSRC	Flujo de tráfico, Condiciones de las carreteras, Peligros en las carreteras...
Confort	V2I	WiMAX/Wi-Fi/ 3G/LTE	Plazas de aparcamiento libres, Descarga de música, Visualización de videos...
Entretenimiento	V2V - V2I	WiMAX/Wi-Fi/ 3G/LTE	Videojuegos, Actividades lúdicas, Conexión a Internet...
Sensorial	V2V - V2I	WiMAX/Wi-Fi/ 3G/LTE	Monitorización medioambiental, Condiciones atmosféricas, Condiciones del vehículo...

de utilidad en los procesos posteriores, hasta la alerta y propuesta de rutas alternativas para los vehículos que vayan en la dirección hacia donde se ha producido el accidente.

- Aplicaciones de eficiencia. Se trata de aplicaciones que monitorizan la ubicación de cada vehículo con el objetivo de optimizar el flujo de tráfico [216]. En esta categoría, la mayoría de las aplicaciones requieren una alta disponibilidad, debido a que los conductores necesitan conocer la información de primera mano para tomar decisiones durante el trayecto. En general, las comunicaciones se realizan tanto entre los propios vehículos como con las infraestructuras de carreteras. A su vez, este tipo de aplicaciones incluye dos grupos:

  - Aplicaciones para controlar las intersecciones. Se refiere a las aplicaciones que permiten alertar y coordinar a los vehículos en puntos críticos como son las intersecciones donde convergen varios flujos de tráfico provenientes de diferentes direcciones. Dichas aplicaciones requieren de un estricto procesado de los datos en tiempo real para poder predecir y avisar antes de que ocurra un desastre, y no avisar cuando ya haya ocurrido.
  - Aplicaciones para gestionar una congestión de tráfico. Una aplicación de este tipo puede ofrecer desde las mejores rutas para los conductores, hasta determinar los mejores horarios para activar los semáforos de un núcleo urbano. El objetivo es reducir la congestión en las carreteras y mantener un tráfico fluido que permita aumentar la capacidad de las carreteras y evitar los atascos de tráfico.

- Aplicaciones de confort. Son las aplicaciones donde los conductores

pueden recibir la información precisa para que el trayecto sea más cómodo y agradable [235]. Los requisitos habituales de este tipo de aplicaciones son la fiabilidad y disponibilidad de la información en el momento preciso y necesario, como por ejemplo: información meteorológica, ubicación de gasolineras y/o restaurantes, información turística, información sobre la disponibilidad de aparcamiento, información sobre los peajes y aduanas, tiempo estimado del viaje, etc. No se requiere un gran ancho de banda para las comunicaciones, que son V2V o V2I, debido a que los datos que se manejan son simples.

- Aplicaciones de entretenimiento. Con el objetivo de distribuir la información a los conductores y pasajeros, esta categoría de aplicaciones tiene como características principales la conectividad y la disponibilidad [227]. Por tanto, las comunicaciones pueden ser directas entre vehículos de forma cooperativa o haciendo uso de la infraestructura situada en las carreteras. El mayor reto que surge en esta categoría es la consecución de un canal estable de comunicación que permita disfrutar del contenido en tiempo real. Como ejemplos de este tipo de aplicaciones destacan los videojuegos online, microblogs, chats, descargas de música, navegación web, intercambio de archivos, la domótica, etc.

- Aplicaciones sensoriales. Una de las aplicaciones más interesantes de una red vehicular es la posibilidad de monitorizar el entorno e intercambiar los datos entre usuarios. Están sobre todo orientadas a zonas urbanas, donde es posible conseguir una masa crítica de vehículos equipados con sensores a bordo. De esa manera, se pueden gestionar las condiciones medioambientales como proponen en [61] y en [142]. Además este tipo de aplicaciones pueden ser potenciadas con la información recogida por los sensores que poseen los teléfonos móviles inteligentes, como proponen en [132] y en [138]. En este contexto, el diseño de una red vehicular de sensores o VSN (Vehicular Sensor Networks) introduce nuevos desafíos y problemas totalmente diferentes a los encontrados en las tradicionales redes de sensores inalámbricas o WSN (Wireless Sensor Networks), por lo que se requieren soluciones innovadoras. Este es un área de investigación prometedora donde se resuelve el gran problema de las WSN, que son las restricciones de energía y consumo, que un vehículo podría subsanar con su sistema de alimentación. Además, los vehículos pueden estar equipados con poderosos equipos de procesamiento, así como con diferentes dispositivos de comunicación inalámbricos, sistemas de navegación, y un sinfín de dispositivos de detección como detectores químicos, sensores acústicos, sensores de vibración, cámaras fotográficas y de video, etc. La combinación de redes vehiculares y plataformas de sensores presenta una gran oportunidad que será explotada a corto y medio plazo.

Muchos proyectos nacionales e internacionales de investigación abogan por hacer real el despliegue de aplicaciones de las redes vehiculares. Estos proyectos de investigación están dando los primeros resultados después de varios años en laboratorios, consiguiendo una mejora sustancial en la seguridad vial en las primeras pruebas de campo, así como una gestión más eficiente del sector del transporte. Actualmente muchos proyectos de investigación tanto académica como industrial en el mundo de los sistemas de transporte inteligentes o ITS (Intelligent Transport System) están activos en todo el mundo. También existen proyectos de investigación que afectan a diversas organizaciones internacionales tales como IEEE, IETF, ETSI, ISO, SAE o ASTM. Por ejemplo, la IETF está trabajando en una extensión del protocolo de Internet o IP (Internet Protocol) conocida como IPv6 o IP móvil, y en auto-configuración para VANETs. La organización ISO también está desarrollando el estándar CALM (Communication Access for Land Mobile) para las redes de vehículos. El consorcio Car-to-Car o C2C-CC [32] se encuentra inmerso en el desarrollo de protocolos de prueba para VANETs. En Europa, el ETSI está trabajando en la adaptación de la norma ISO, que describe los estándares IETF. La interoperabilidad y la integración de estos proyectos son objeto de intensos debates y estudios, así como motivo de decenas de reuniones científicas a lo largo de la geografía mundial.

En lo referente a los proyectos industriales [211] [72] [226] cabe destacar: en Estados Unidos los proyectos VII, CICAS y IVBSS; en Europa los proyectos CVIS, SafeSPOT, CARAVAN, COOPERS, PReVENT, GST, DRiVE, HIGHWAY, FleetNet, SeVeCom, GeoNet, PREDIT, NoWe; en Japón los proyectos SmartWay y VIC; y en India el proyecto ITSIndia.

La mayoría de los proyectos mencionados incluyen la integración de las comunicaciones V2V y V2I [72]. Por ejemplo, PReVENT ayuda a los conductores a evitar los accidentes o a mitigar su impacto en el peor de los casos. GST (Global System for Telematics) se centra en la creación de un estándar abierto para los servicios telemáticos intra-vehículos. El proyecto CVIS se centra en la seguridad vial y en la integración de las comunicaciones V2V y V2I. El proyecto DRiVE se centra en el uso exclusivo de la infraestructura existente para la implementación del sistema de IVC, utilizando diferentes tecnologías sobre redes heterogéneas.

En España existen varios proyectos vigentes que implementan aplicaciones para desplegar entornos vehiculares, como los proyectos del grupo en el que se ha desarrollado esta Tesis: DEPHISIT (Desarrollo Experimental de una Plataforma Híbrida Inalámbrica para Sistemas Inteligentes de Transporte [114]), ATLAS (Aplicaciones de la Tecnología Lte para Aumentar la Seguridad [222]) o CASUS (Cooperación móvil segura Aplicada a Situaciones de emergencia e infraestrUcturas críticas de tranSporte [242]), además de los ya finalizados TUERI (Tecnologías segUras y Eficientes para las Redes inalámbricas en la Internet de las cosas con aplicaciones en transporte y logís-

tica [245]) y MUOVE (Mejora de la segUridad vial mediante la planificación, diseño e integración de servicios criptOgráficos en VanEts [244]).

1.2.4. Seguridad

La seguridad de una red vehicular es uno de los mayores retos a los que se debe hacer frente para conseguir desplegar una red de este tipo. Gracias al auge de los vehículos inteligentes, se puede conseguir que los trayectos en carretera sean más seguros y rápidos. Sin embargo, a la vez, dichos vehículos se vuelven más susceptibles a ataques de personas malintencionadas con altos conocimientos en informática que pueden aprovecharse para crear el caos y el pánico, provocando accidentes o inyectando la red de información falsa [22]. Esta es ahora la principal cuestión a resolver por los investigadores y los fabricantes de automóviles, para dar confiabilidad a las VANETs, ya que en este tipo de redes no sólo se juega con datos, sino con algo mucho más crítico como son las vidas humanas. Dicho esto, se hace esencial que cualquier tipo de información que se maneje en esta red no pueda ser modificada, alterada o eliminada por terceras personas. A la par que el sistema debe ser lo suficientemente inteligente como para determinar la responsabilidad de un acto en concreto, sin desvelar la identidad del responsable [255]. Además, cualquier tipo de información que se genere y distribuya en la red, así como información intrínseca de los propios vehículos y sus conductores, debe ser cifrada y protegida para garantizar la seguridad y el buen funcionamiento de los sistemas de transporte inteligentes.

Por el contrario, aunar un grado alto de seguridad con un sistema rápido y dinámico se vuelve casi una utopía, de ahí los múltiples proyectos e investigaciones que se están realizando en los últimos años. Los vehículos se mueven de forma constante a velocidades elevadas y en multitud de sentidos, provocando conexiones cortas y con interferencias. Por todo ello, la seguridad requerida de este tipo de redes es muy compleja de establecer.

En el lado de las comunicaciones, los estándares que definen las comunicaciones realizadas en las VANETs son el DSRC y el WAVE. Respecto a estos estándares, ha habido un gran inversión por parte de gobiernos, centros de investigación e industrias bajo el paraguas de los sistemas de transporte inteligente. La Organización Internacional de Normalización o ISO (International Organization for Standardization) está desarrollando una familia de estándares y arquitecturas internacionales para el acceso a las comunicaciones móviles terrestres CALM. Se espera que el futuro sistema CALM haga uso de una amplia gama de tecnologías, incluyendo satélites, comunicación de tipo móvil (GSM, 3G y 4G/WiMAX), redes inalámbricas Wi-Fi de área local (WLAN), redes inalámbricas Wi-Fi para entornos vehiculares (WAVE) evoluciones de los protocolos IEEE P1609 y IEEE802.11P, redes inalámbricas Bluetooth de área personal (WPAN), mm-Wave, identificación por radiofrecuencia (RFID), etc. El estándar WAVE ha sido propuesto recientemente

en la banda de frecuencias de comunicaciones dedicadas de corto alcance o DSRC. DSRC/WAVE es la única tecnología inalámbrica que puede cumplir con el requisito necesario de las VANETs de tener una latencia extremadamente baja en el intercambio de mensajes, tan necesario para el control de la seguridad vial en tiempo real.

Un primer paso para ver la complejidad real a la que se debe hacer frente en este tipo de redes, es categorizar las mayores amenazas a las que se debe hacer frente. La idiosincrasia de una VANET hace que haya ataques sobre redes ad-hoc convencionales que no le afecten, pero por otro lado, existen ataques críticos que repercuten de una forma muy grave sobre las redes vehiculares y que no afectan a las redes ad-hoc clásicas [255]. Múltiples han sido los autores que han categorizado los principales ataques sobre redes vehiculares [66] [75]. No obstante, a continuación se aporta una clasificación basada en el trabajo de [181] donde se priorizan las amenazas en base a su carácter criptográfico.

- Ataques contra la disponibilidad. Garantizar que la red sea funcional y que la información sea accesible en cualquier momento, es una de las premisas más importantes que debe cumplir una red vehicular. Este tipo de ataques van en contra de esta premisa, intentando inyectar en la red información errónea para hacerla lenta y no utilizable. Los ataques más comunes en esta categoría aplicados a VANETs son: los ataques de denegación de servicio DoS [255] [66] [221] [31], los ataques de jamming [186] [105], los ataques greedy [104], los ataques por broadcast tampering [255], los ataques de malware [66] [1], los ataques de spamming [66] [255] o los conocidos ataques blackhole [255] [1] [66], grayhole [196], simkhole [29], wormhole [204] [223], entre otros.

- Ataques contra la autenticidad y contra la identificación. Este tipo de ataques se basan en la clásica suplantación de identidad y en la infiltración de usuarios malintencionados. Es crucial poder autenticar de forma rápida y segura a los vehículos, sin dejar que ningún usuario no legítimo o fraudulento pueda formar parte de la red vehicular. La importancia del proceso de autenticación en una VANET está caracterizada por la frecuencia de acceso de los vehículos a la red o a sus servicios, ya que se unirán y desconectarán constantemente debido a su carácter móvil. Hay varios tipos de ataques en esta categoría, entre los que destacan los ataques Sybil [71], los ataques de ubicación falsa o GPS spoofing [66] [1], los ataques de suplantación [1], los ataques de tunnelling [206] [255], los ataques de masquerading, los ataques de manipulación de mensajes, los ataques de supresión de mensajes, los ataques de alteración o los ataques de replicación de certificados [181].

- Ataques contra la confidencialidad. La confidencialidad en una VANET debe asegurar que los datos sólo sean accesibles y comprensibles por

los vehículos autorizados. Si una red vehicular es vulnerable a este tipo de ataques, un atacante puede obtener información crítica como la ubicación del vehículo o los trayectos que ha realizado, así como acceder a datos privados de los usuarios. Autores como Raya y Hubaux [209] remarcan que existen ciertos datos que no son sensibles, por lo que la confidencialidad no es necesaria en ciertos casos. Los ataques más destacados de esta categoría son los ataques de espionaje [66], los ataques de recopilación de información o los ataques sobre el análisis del tráfico.

- Ataques contra la integridad y la veracidad de los datos. La integridad de la información intercambiada en un sistema vehicular debe asegurar que estos datos no se han alterado durante la transmisión. Los mecanismos de protección de la integridad, por lo tanto, ayudan a proteger la información contra los ataques de alteración, supresión o adición. Esta categoría se focaliza principalmente en las comunicaciones V2V a causa de su fragilidad. Una de las posibles técnicas que facilitan este tipo de ataques es la manipulación de los sensores dentro del vehículo [87]. Los ataques más relevantes son los típicos ataques de masquerading, los ataques de repetición [201], los ataques de manipulación de los datos [206] o los ataques de ilusión [176].
- Ataques contra el no repudio. El no repudio en el contexto de seguridad informática, se aplica a la capacidad de verificar que el emisor y el receptor son las entidades que afirman haber enviado y recibido los mensajes, respectivamente [228]. En el contexto de las redes vehiculares se refiere a la capacidad de verificar todos los cambios que se realicen tanto en el hardware como en el software. Los ataques más reconocidos de esta categoría son los ataques de trazabilidad y pérdida de eventos [232].
- Otros tipos de ataques. Existen otros muchos tipos de ataques, como los ataques sobre la privacidad que pueden detectar dónde está un vehículo en cada momento, si está en movimiento o en circulación, etc. [122] [66]; los ataques de timing que se realizan con el fin de ralentizar las trasmisiones de los mensajes [237] [1]; los ataques de fuerza bruta para descifrar los mensajes intercambiados; o los ataques por antonomasia, los ataques MitM (Man in the Middle) donde un vehículo malintencionado intercepta y manipula la comunicación entre dos vehículos legítimos de la red [1].

1.3. Desarrollo Móvil

Como se ha remarcado en secciones anteriores, el despliegue real de una VANET no parece algo abarcable a corto plazo. Sin embargo se puede utilizar

la tecnología existente para acelerar el proceso y ver los primeros servicios reales de una VANET desplegada en un periodo corto de tiempo. Una de las tecnologías que más pueden aportar por su carácter móvil y su alto nivel de penetración en la sociedad son los teléfonos móviles inteligentes. Con ellos, se puede simular el funcionamiento de una OBU o de una RSU, poniendo al alcance de cualquier usuario una potencia de cálculo que sea suficiente como para poder interactuar en una red vehicular. Propuestas innovadoras, como VAiPho (VANET in Phones) [36], proponen desplegar la primera red vehicular haciendo uso únicamente de la tecnología de los teléfonos móviles para suplir la carencia del hardware adicional que se debe instalar tanto en vehículos como en carreteras [37] [175]. Esta sección se adentra en el paradigma de la programación para dispositivos móviles, como eje central de los desarrollos e implementaciones que se han realizado en el presente trabajo de Tesis [156]. Siguiendo la estela marcada por VAiPho [144] [170] [169], se ha considerado implementar todas las aplicaciones y servicios para las redes vehiculares, que se han generado como resultado de las investigaciones realizadas durante los últimos años y que se proponen en capítulos posteriores, sobre dispositivos móviles, debido al nivel de penetración que tienen en la sociedad actual y a la capacidad de cómputo de los smartphones de hoy en día.

1.3.1. Evolución Histórica

La popularidad del desarrollo en plataformas móviles está creciendo exponencialmente. Estas aplicaciones móviles son intrínsecamente diferentes de las aplicaciones de escritorio tradicionales. En 1974, Lehman y Belady propusieron un conjunto de leyes relativas a la evolución del software [133]. Las leyes de Lehman se basaron en observaciones empíricas de software comercial. Mucho tiempo ha pasado desde entonces, pero la mayoría de las leyes de Lehman siguen siendo válidas en nuestros días a pesar de la evolución de la tecnología en general [254].

Con la llegada de los dispositivos móviles inteligentes ha aparecido un nuevo nicho de mercado para las aplicaciones software. Estas aplicaciones móviles son diferentes de las aplicaciones de escritorio tradicionales [185], por lo que su desarrollo también es diferente. Aún así, el desarrollo móvil se sigue rigiendo por las leyes de Lehman de 1974 [256], que son:

- Cambio continuo. Para que un software sea útil, debe estar evolucionando y cambiando constantemente. Esto se aplica a la perfección al desarrollo móvil.
- Aumento de la complejidad. A medida que el software cambia y evoluciona, su complejidad aumenta. Es totalmente aplicable al desarrollo móvil ya que nuevos sensores y tecnologías aparecen con la evolución natural.

- Disminución de la calidad. A medida que el software cambia, su calidad va siendo peor. Cuando un desarrollo evoluciona, mantener el mismo nivel de calidad se complica de forma exponencial porque el código a mantener es mucho mayor y los bugs y errores aparecen con más frecuencia.

Además, esta evolución en el desarrollo móvil ha venido marcada por la evolución en la tecnología de las comunicaciones móviles [202]. La primera generación de tecnología móvil se conoció como 1G y fueron los estándares de telecomunicaciones analógicas introducidos en la década de 1980. Las redes 1G se basan en los sistemas analógicos, mientras que las redes 2G se basan en los sistemas digitales. Los sistemas de transmisión móvil de primera generación se basaron exclusivamente en FDMA/FDD y en FM analógica. La segunda generación o 2G es el estándar de comunicaciones más popular y extendido por la geografía mundial. Esta segunda generación utiliza formatos de modulación digital y técnica de acceso múltiple como TDMA/FDD y CDMA/FDD. El 2G se compone de tres estándares TDMA y un estándar CDMA, como son el GSM (Global System for Mobile), el IS-136 (Interim Standard 136), el PDC (Pacific Digital Cellular) o el IS-95 (Interim Standard 95). La tercera generación o 3G destaca por centrarse en los servicios multimedia y la mejora de las velocidades de datos a Internet. Se trata de un conjunto de estándares utilizados para los dispositivos móviles y sus servicios de telecomunicaciones en las redes que cumplan con las especificaciones IMT-2000. La última generación, conocida como 4G, es la cuarta generación de estándares de comunicaciones móviles de telefonía. Es el sucesor natural de los estándares de tercera generación. El sistema 4G proporciona un acceso mucho más rápido a Internet gracias a un mayor ancho de banda en las comunicaciones. Esto provoca que las aplicaciones para plataformas móviles no tengan que preocuparse del ancho de banda a la hora de poder visualizar streaming de videos, o cualquier otra tarea que requiera de una conexión a la red de banda ancha. En el futuro, el 5G que es un proyecto que está ya en fase de realización, sustituirá al 4G. La primera gran diferencia entre el 4G y la idea de 5G que hay actualmente es la frecuencia que se usa. Mientras en 4G lo más habitual es usar frecuencias bajas, entre los 800 MHz y 2.6 GHz, en el caso de las pruebas de 5G que se han llevado a cabo hasta ahora se han utilizado bandas situadas entre los 26 y 38 GHz. Además de la velocidad, la latencia es uno de los puntos importantes en las futuras redes 5G. Se está hablando de que esta nueva tecnología sería capaz de reducirla hasta valores cercanos al milisegundo. Los retos de implantar una nueva tecnología van más allá de la velocidad y la latencia. Lo primero de todo será que todos los países se pongan de acuerdo en cuáles serán concretamente las bandas que se destinarán al 5G, a fin de que dicho espacio del espectro se deje disponible. Corea del Sur y Europa ya han anunciado que el objetivo pasa por el comienzo de los despliegues en 2020.

Por tanto, actualmente está teniendo lugar un momento dulce para el desarrollo de plataformas móviles, debido a la evolución de la tecnologías de transmisión de datos y a las mejoras constantes tanto en el hardware de los dispositivos como en el software, en sus sistemas operativos.

1.3.2. Sistemas Operativos Móviles Nativos

En el ámbito de la programación para dispositivos móviles, la amplia variedad de dispositivos y sistemas operativos con los que siempre han contado los desarrolladores, ha supuesto un dilema a la hora de seleccionar la plataforma en la que desarrollar. El número de usuarios de cada plataforma es grande y razonablemente balanceado a nivel mundial [30], por lo que seleccionar un SO (Sistema Operativo) para trabajar se vuelve una ardua tarea. Otro problema surge cuando se desea desarrollar para más de una plataforma, siendo necesario repetir algunos procesos tales como el desarrollo nativo en cada una de las plataformas elegidas, las pruebas realizadas, así como el despliegue específico para lanzar la aplicación en cada tienda en concreto. En la primera década del siglo XXI, las plataformas como J2ME (Java 2 Micro Edition) empezaron a abrir caminos hacia el desarrollo multiplataforma, lo que permitía poder crear un único software que pudiera ser ejecutado en cualquier sistema operativo móvil gracias a la máquina virtual de Java. Sin embargo, poco a poco, cada plataforma fue personalizando más su arquitectura interna lo que hacía que el desarrollo multiplataforma se convirtiese en un dolor de cabeza para los usuarios debido a su pobre rendimiento. Eso conllevó que el uso de J2ME disminuyera y se viera abocado a la desaparición, mientras los principales kits de desarrollo de software o SDKs (Software Development Kit) específicos de cada la plataforma, creciesen y ofreciesen mucha más potencia y versatilidad de cara a los desarrolladores. Sin embargo, recientemente, han vuelto a surgir diversas herramientas de desarrollo multiplataformas basadas en programación web, como por ejemplo, Appcelerator Titanium, PhoneGap o Sencha Touch, desafiando los paradigmas actuales del desarrollo nativo. Estas herramientas son cada vez más potentes, gracias en gran medida al avance de las tecnologías web como HTML5 o CSS3. Ya no es tan grande la barrera entre un desarrollo nativo y un desarrollo multiplataforma, y cada vez será menor, debido a los nuevos paradigmas de programación móvil donde se desarrolla usando tecnologías web como JavaScript (React Native) o .NET (Xamarin), para a la postre generar código nativo. Aún así, cuando se requiere realizar desarrollos donde aprovechar todas las tecnologías que ofrece un dispositivo móvil (Bluetooth Low Energy, WiFi Direct, acelerómetros, giroscopios, etc.), el desarrollo nativo sigue imponiéndose con clara ventaja al desarrollo multiplataforma, ya que aprovecha mejor todas las librerías o APIs (Application Programming Interfaces) que ofrecen los sistemas operativos móviles para acceder a sus sensores y demás componentes electrónicos.

Los principales sistemas operativos móviles de la actualidad (ver Figura 1.7), se nombran y detallan a continuación, haciendo énfasis en sus ventajas e inconvenientes:

- Android: El sistema operativo número uno en cuanto a popularidad. Con una cuota de mercado cercana al 85 %, el sistema operativo de Google se caracteriza por ser abierto y disponible para cualquier fabricante interesado en utilizarlo para sus dispositivos móviles. Esta disponibilidad ha generado sin embargo una gran fragmentación, pudiéndose encontrar innumerables dispositivos de miles de formas y funcionalidades con todas las versiones de Android existentes. Además, la posibilidad de que cada fabricante incluya su propia capa sobre el original, propicia que la experiencia de usuario no sea siempre la deseada por Google y las actualizaciones tarden en llegar. Una penetración de mercado tan grande ha propiciado por otro lado, que aunque en un primer momento el sistema operativo de Apple, iOS, fuera el más popular de los sistemas operativos para los desarrolladores, cada vez más, estos dedican grandes esfuerzos a diseñar sus apps para los usuarios de Android.

- iOS: El primer sistema operativo móvil que llegó a una masa crítica de usuarios en el año 2007. Su compañía creadora, Apple, se supo posicionar en el apogeo primitivo de las tecnologías móviles. Lo que caracteriza a iOS frente a otros es que es un sistema operativo cerrado. Apple no permite que se modifiquen características internas del sistema más allá de las limitadas opciones que da en los ajustes. Un sistema cerrado permite, sin embargo, ofrecer siempre una experiencia más estable y segura, tal y como diseñó el fabricante en un principio. Sin embargo a muchos usuarios, que buscan una mayor personalización, se les pueden quedar cortas las opciones que le da Apple. Por otro lado, como también suele ser habitual en los productos de la empresa, no se licencia a terceros por lo que tan solo los iPhone (teléfonos creados por Apple) disponen de este sistema operativo.

- Windows Phone: El sistema operativo móvil por antonomasia de Microsoft, que está realizando un gran esfuerzo financiero para posicionar Windows Phone como una tercera opción interesante para los consumidores después de que llegara tarde a la fiesta de los smartphones. Su alianza con Nokia y su posterior compra le ha ayudado a darse a conocer mejor e ir arañando cuota de mercado a los dos líderes. Los últimos datos hablan de un 2,5 % a nivel mundial. Con un diseño radicalmente distinto a las dos opciones ya comentadas, Windows Phone destaca por su pantalla de inicio personalizable que ofrece las notificaciones de las apps de una manera sencilla y limpia. Además ofrece una experiencia

de usuario muy buena independientemente del tipo y gama de terminal en que se esté usando. Aunque con menos apps disponibles que en Android y iOS, Windows Phone, cuenta ya con más de 500.000 apps en su tienda, además de ofrecer aplicaciones propias de la compañía como Skype, OneDrive o Xbox Live.

- BlackBerry OS: Anteriormente conocida como RIM, Blackberry no está pasando por sus mejores momentos. Al igual que le pasó a Nokia, el cambio de paradigma en los smartphones le pilló con el pie cambiado. Acostumbrado a ofrecer terminales con teclado físico, el paso a las pantallas táctiles se le atragantó. Sin embargo, los esfuerzos realizados por la compañía canadiense para recuperar el terreno perdido han sido grandes y en el año 2012 lanzaron su órdago con un renovado sistema operativo el Blackberry 10. Aun así, los últimos estudios sobre cuota de mercado lo dejan en tan solo un 0,5 % mundial. Blackberry 10 tiene una interfaz más fluida, un teclado inteligente y táctil más depurado y otra serie de opciones que lo acercan a las de la competencia. Al igual que iOS, el sistema operativo es software propietario y solamente los teléfonos de la compañía llevan su sistema instalado.

Figura 1.7: Sistemas Operativos Móviles

- Firefox OS: Sistema operativo basado en HTML5 con núcleo Linux, de código abierto. Desarrollado por Mozilla Corporation con apoyo de empresas como Telefónica. El sistema operativo está basado en Linux

y usa la tecnología de Mozilla, Gecko. Se basa en estándares abiertos como por ejemplo HML5, CSS3 y JavaScript. Pensado para ser un sistema operativo realmente abierto, a diferencia de Android, donde Google controla ciertos aspectos del sistema. Esta característica, permite a Firefox OS llegar a cubrir el nicho de mercado de la gama baja con mayor facilidad que Android. El anuncio hecho en febrero del año 2014 de lanzar un smartphone por 25 euros va completamente en esa línea. Movistar ya lanzó hace más de dos años los primeros smartphone con este sistema operativo en España y Latinoamérica. Entre las interesantes características de este sistema operativo abierto están las aplicaciones web, que pueden ser de dos tipos diferentes: aplicaciones de servidor o empaquetadas. A diferencia de los sistemas operativos ya comentados, en este caso, las apps de servidor corren vía web, es decir son páginas webs con la apariencia de aplicaciones y sin conexión a Internet no es posible acceder a estas. Las aplicaciones empaquetadas necesitan la descarga de un paquete comprimido y se cargan desde la fuente local cada vez que se accede a la aplicación.

- Ubuntu Touch: Otro sistema operativo basado en Linux pero en esta ocasión bajo la famosa firma Ubuntu. Presentado en 2013, se trata de un proyecto de la empresa Canonical. En la actualidad varias empresas están desarrollando terminales para este sistema operativo, entre ellas la española BQ. Ubuntu Touch utiliza las mismas tecnologías de la versión de escritorio, por lo que ambas comparten apps sin problemas de compatibilidad. Dispone también de algunas de las aplicaciones más populares como Facebook y YouTube.

- Tizen: Sistema operativo móvil, también basado en Linux, patrocinado por Linux Foundation y Fundación LiMo. Se ha desarrollado a partir de la plataforma Linux de Samsung. Aunque en un principio fue presentado como un SO de código abierto, Tizen 2 funciona con un sistema de licencias no abiertas. El SDK completo fue publicado bajo licencia de Samsung de código no abierto. Aunque pueda parecer que Tizen forma parte de la estrategia de Samsung a largo plazo, su apuesta es arriesgada debido a que no tiene casi cuota de mercado. De momento sólo algunos de los dispositivos de Samsung lo incorporan como el caso del famoso smartwatch Samsung Gear S.

- WebOS: Este interesante sistema operativo, fue a Palm lo que Blackberry 10 a RIM. Sin embargo, pese a las buenas críticas que cosechó, no consiguió salvar la compañía. Tras la compra por parte de HP de la compañía Palm Inc., en la actualidad webOS es propiedad de LG, que lo utiliza como sistema operativo para sus televisores inteligentes.

1.3.3. Desarrollo Multiplataforma

El enfoque actual del desarrollo para dispositivos móviles hace entrever que en general es necesario desarrollar cada aplicación en más de un sistema operativo móvil para poder llegar a los máximos usuarios posibles con objeto de que la aplicación pueda obtener un aporte de valor tanto para el resto de usuarios como para la rentabilidad de la propia aplicación. Limitar el alcance a una sola plataforma impide que una aplicación pueda impactar a un mayor número de usuarios y clientes potenciales que puedan estar interesados. El desarrollo multiplataforma abre la posibilidad de ampliar el ámbito de una aplicación sin incrementar en demasía el coste del propio desarrollo.

Hoy en día es mucho más fácil poder contar con diferentes herramientas que permiten la creación de aplicaciones web para dispositivos móviles con una apariencia nativa y con un rendimiento similar al desarrollo nativo mediante el SDK de cada plataforma en cuestión. Por ello el desarrollo multiplataforma ofrece varias ventajas frente al desarrollo nativo:

- No existe limitación a la hora de desarrollar sobre un lenguaje y un framework específico para cada plataforma. Existe un único lenguaje que permite desarrollar para múltiples plataformas.
- Se puede aprovechar del conocimiento y la experiencia de los desarrolladores web para crear aplicaciones móviles usando paradigmas y lenguajes web.
- Reduce la necesidad de repetir el desarrollo nativo en múltiples plataformas para generar múltiples aplicaciones compatibles.

Sin embargo, aún existen varias desventajas que hacen meditar mucho a un desarrollador la elección de un desarrollo multiplataforma:

- Aún es necesario una mayor implicación de los estándares web para adaptarse mejor a los dispositivos móviles. Tecnologías como HTML5 y WebGL deberán progresar a la hora de ayudar en la superación de las limitaciones existentes en las aplicaciones multiplataformas. Por ejemplo, HTML5 debe proporcionar el almacenamiento en caché y las capacidades de almacenamiento en local que permitan a las aplicaciones multiplataformas operar en modo offline con la solvencia que lo pueden hacer las aplicaciones nativas. WebGL, por su parte, debe incentivar la facilidad de representación de gráficos [239]. Esto permitirá que las aplicaciones multiplataforma se lleguen a comportar casi como una aplicación nativa.
- Las herramientas de desarrollo multiplataformas actuales no permiten el acceso completo a algunas funciones del dispositivo móvil (por

ejemplo, el acceso completo a las comunicaciones inalámbricas, la integración con otras aplicaciones del dispositivo, etc.). Las APIs existentes deben ser mejoradas y evolucionadas para que un desarrollador multiplataforma pueda contar con las mismas librerías internas que lo hace un desarrollador nativo.

- El desarrollo multiplataforma debe ser un proceso sencillo y no arduo, en el que desplegar en múltiples plataformas sea una acción simple. Las tareas largas y pesadas desalientan a los desarrolladores a usar herramientas multiplataforma, prefiriendo programar en nativo en cada plataforma específica. La personalización que puede llegar a realizar un desarrollador debe ser lo más cercana posible al desarrollo nativo, sin que un usuario pueda darse cuenta de que una determinada aplicación ha sido desarrollada en nativo o en multiplataforma.

- El desarrollo multiplataforma debe mejorar mucho el rendimiento actual de las aplicaciones que genera para que sea posible su ejecución en entornos limitados. Es necesario aclarar que las aplicaciones móviles se ejecutan en una CPU lenta, con relativamente poca memoria RAM, y alimentándose de una batería limitada [27]. Este escollo puede verse solventado con el paso del tiempo gracias a la mejora constante de las capacidades de cómputo de los teléfonos inteligente, aunque el tema de la batería sigue siendo un problema global, por lo que el rendimiento y eficacia debe ser clave en una aplicación móvil.

Por todo lo anterior, es importante remarcar que actualmente el desarrollo multiplataforma no es una opción si se requiere exprimir al máximo los dispositivos móviles haciendo uso de todas las tecnologías que ofrecen, sobre todo las tecnologías inalámbricas para generar redes ad-hoc como se ha realizado en las aplicaciones resultado de las investigaciones del presente trabajo de Tesis.

1.3.4. La Plataforma Android

La elección del sistema operativo móvil utilizado para el desarrollo de los prototipos ha sido la del sistema operativo de Google: Android, por su gran capilaridad en el mercado actual, como se ha destacado antes, y por contar con múltiples ecosistemas paralelos que se basan en este sistema operativo para dar soporte a múltiples dispositivos electrónicos como son smartphones, tablets, televisores, relojes, vehículos y hasta objetos inteligentes más propios de la Internet de las Cosas.

Android es una plataforma para dispositivos móviles de la compañía norteamericana Google. Está basado en Linux, por lo que es de código abierto y cualquiera puede editarlo. Esa es la razón por la que empresas como Samsung, LG o Sony ofrecen sus propias versiones personalizadas de Android, con

objeto de diferenciarse en un mundo donde más del 85 % de los smartphones usan este sistema operativo. La idea detrás de Android se remonta hasta octubre de 2003, cuando Andy Rubin, Rich Miner, Chris White y Nick Sears fundaron Android, Inc. Su idea era desarrollar un sistema operativo para móviles que estuviera basado en Linux. Además de eso, querían que fuera fácil de utilizar y de programar. Las empresas dominantes en la época eran Nokia y BlackBerry, por lo que cuando Google anunció en 2005 que compraba Android, el mercado de tecnologías móviles empezó a cambiar de forma drástica.

Existen 14 versiones de Android que han sido distribuidas desde que apareció Apple Pie en 2007. Todas reciben el nombre de un postre en inglés, y en riguroso orden alfabético: Android 1.0 Apple Pie, Android 1.1 Banana Bread, Android 1.5 Cupcake, Android 1.6 Donut, Android 2.0 Eclair, Android 2.2 Froyo, Android 2.3 Gingerbread, Android 3.0 Honeycomb, Android 4.0 Ice Cream Sandwich, Android 4.1 Jelly Bean, Android 4.4 KitKat, Android 5.0 Lollipop, Android 6.0 Marshmallow o el reciente Android N.

Android debe considerarse una plataforma porque aúna a varios sistemas operativos móviles, como Android Wear para los wearables, Android TV para las televisiones o Android Auto para los vehículos. La apuesta de Android por los dispositivos wearables, como los relojes inteligentes, fue lanzada el 18 de marzo de 2014. Pretende ser un ecosistema fácil de usar y especialmente diseñado para este tipo de aparatos, y empresas como Asus, LG y Motorola ya han lanzado relojes con esta plataforma. En el camino de Android Wear hay otros competidores, como Tizen, un sistema operativo creado por Samsung y que ha sido implementado en sus smartwatches, o watchOS, basado en iOS y utilizado por el Apple Watch. Por su parte Android TV es el heredero del fallido Google TV. Es un sistema operativo diseñado para las televisiones inteligentes. Anunciado en 2014, tiene una interfaz similar a la de la tienda de aplicaciones Google Play, pero orientada hacia la venta de contenidos multimedia como programas de televisión y películas. En ese mismo año, en 2014, Google anunció Android Auto, el sistema operativo de Google para conseguir con los vehículos lo que en su momento consiguió con los teléfonos. Ya conformó la Open Automotive Alliance, de la que ya forman parte 28 fabricantes de automóviles. Android Auto busca convertir en inteligentes los coches, y en ofrecer para estos vehículos aplicaciones de música, mapas, GPS, telefonía y búsqueda web, todo a través de comandos de voz para una conducción segura.

Los componentes principales de cualquier sistema operativo Android son: las aplicaciones, el framework de interoperabilidad, las librerías, el entorno de ejecución y el núcleo Linux. Dentro de la plataforma Android está el eje central de todos los sistemas operativos que se crean para los diferentes dispositivos: AOSP o Android Open Source Project. Android Open Source Project es como su nombre lo dice, el proyecto de código abierto de Android

liderado por Google, con la tarea de mantener y continuar el desarrollo futuro de la plataforma Android. En el sentido más puro, AOSP se refiere a código de Google que no ha sido modificado. Es el código fuente que se utiliza para generar los diferentes sistemas operativos de los smartphones, tablets, televisores, relojes, vehículos, objetos inteligentes, etc.

Google desarrolla Android a puerta cerrada, y cuando está lista una nueva versión, la empresa libera el código fuente. Es entonces cuando cualquier fabricante puede tomar el sistema operativo en su estado vainilla y añadir sus capas de personalización. Android en cualquier dispositivo está personalizado en cierto grado, ya que necesita tener drivers específicos para que el aparato pueda funcionar, muchos de los cuales son propietarios, y es por esto que Android siempre termina siendo una combinación de partes abiertas con partes cerradas.

Por esto se pueden encontrar diferentes sabores de Android, que se utilizan para personalizar a medida cada nuevo gadget que aparece. Además hay una versión con alto nivel de interoperabilidad con otros sistemas operativos, para objetos inteligentes, llamada Brillo. Brillo es el sistema operativo para Internet de las Cosas que presentó en el año 2015 la propia Google. Está pensado para ser utilizado con dispositivos inteligentes de bajo consumo de energía y memoria limitada. Junto con Brillo, Google también introdujo el protocolo Weave, que los dispositivos de Brillo usarán para comunicarse con otros dispositivos y que se espera que sea adoptado por otros sistemas operativos de IoT.

Todo esto hace pensar que la mejor plataforma para desarrollar aplicaciones y servicios para redes vehiculares que utilicen dispositivos inteligentes actuales como sistemas de conexión, sea Android. De ahí la elección de la plataforma Android como sistema operativo para desarrollar los prototipos, aplicaciones y servicios resultado de los trabajos de investigación de esta Tesis.

Capítulo 2

Autenticación en VANETs

El proceso de autenticación es uno de los procesos más críticos para cualquier tipo de red. Mediante la autenticación se verifica la veracidad del origen de los datos. Es el primer método de seguridad que deben implementar las redes, y al que deben enfrentarse todos los usuarios al intentar acceder a ellas. Existen múltiples métodos de autenticación de usuarios para escenarios dispares, como por ejemplo los métodos biométricos basados en huellas dactilares, la retina ocular, etc., los métodos basados en tarjetas inteligentes que almacenan la información de los certificados del usuario, o los métodos clásicos basados en contraseñas. La elección del método más seguro y eficiente de autenticación se basa en las características del sistema donde se va a desplegar. De esa manera, un sistema de autenticación biométrica no resulta muy útil en un sistema basado en nodos en movimiento como pueden ser las VANETs. Por ello, es esencial explotar las propiedades de los sistemas para que el método de autenticación utilizado sea el más acorde a cada escenario. En lo referente a las redes móviles descentralizadas, es necesario algún tipo de autenticación rápida, así que la utilización de algoritmos criptográficos pesados no es recomendable.

En este capítulo se propone un esquema de intercambio de información confidencial en entornos no seguros sobre redes móviles descentralizadas, basado en el concepto de demostración de conocimiento nulo no interactiva. De esa manera se consigue que en una única comunicación se puedan inferir datos relevantes para la verificación de la legitimidad de los nodos de la red. El esquema ha sido diseñado con el fin de ser aplicado a los procesos de autenticación y control de accesos en escenarios de transporte, permitiendo el establecimiento de claves para uso de algoritmos de cifrado ligeros. Es destacable que los resultados expuestos en este capítulo han sido publicados en una revista de impacto indexada en el primer cuartil [167], así como en un congreso indexado en la base CORE [160].

2.1. Estado del Arte

Cuando la seguridad de las comunicaciones que se establecen sobre canales inseguros se protege a través de criptografía simétrica, la distribución previa de las claves secretas es una de las tareas más delicadas. Este tema ha atraído mucho la atención de los investigadores en los últimos años [179] [68]. Uno de los sistemas más conocidos para distribuir de forma segura estas claves secretas a través de canales inseguros continúa siendo el protocolo que Diffie y Hellman propusieron en la década de los 70 [67]. Este esquema permite que dos usuarios puedan calcular una clave secreta compartida a través de dos números secretos y un intercambio público de información, gracias al problema del logaritmo discreto. Sin embargo, este algoritmo no incluye la autenticación del usuario, lo que provoca la posibilidad de que un ataque Man in the Middle sea lanzado. Por ello, con el fin de subsanar esta deficiencia, se pueden utilizar certificados.

Respecto a las MANETs, se han publicado muchas propuestas para proteger las comunicaciones basadas tanto en criptografía de clave secreta [143], como en criptografía de clave pública [246]. El nivel de seguridad de muchos de los esquemas de clave simétrica es alto, pero su principal inconveniente viene a colación de la dificultad de poder distribuir con antelación al inicio de las comunicaciones seguras, las claves secretas compartidas entre los usuarios legítimos de la red. En un entorno como el de las MANETs aplicado a la Internet de las Cosas [9], la hipótesis acerca de la existencia de un canal completamente seguro para transmitir las claves simétricas requeridas, es muy complejo de cumplir. Además, si la red es grande y está basada únicamente en criptografía simétrica, el número de claves secretas que necesitarían sería demasiado elevado. Por tanto, para resolver el problema de la distribución segura de claves secretas, la criptografía asimétrica o de clave pública es la mejor solución.

Por otro lado, la criptografía de clave pública ofrece la posibilidad de establecer un esquema de firma digital [95], siempre y cuando el emisor utilice su clave privada para firmar el mensaje y el receptor verifique la firma a través de la clave pública del emisor. A través de la firma digital, la autenticación tanto del emisor (identificación) como del propio mensaje (integridad) está garantizada. Curiosamente, la capacidad de utilizar los esquemas de firma digital que la criptografía asimétrica provee, es exactamente la manera de resolver el principal problema de certificación planteado antes, que no es otro que la necesidad de establecer un entorno de confianza para utilizar claves públicas y prevenir los ataques Man in the Middle. En un ataque de este tipo, el atacante realiza conexiones independientes de forma sincronizada con dos nodos de la red, para transmitir mensajes entre ellos, haciéndoles creer que se están comunicando directamente entre sí a través de una conexión privada, cuando en realidad toda la conversación está siendo controlada por

el atacante. Una posible solución al problema consiste en que cada clave pública esté certificada para que la firma digital que contiene el certificado, garantice la identificación de cada usuario legítimo de la clave público en cuestión.

Existen varios modelos para lograr la certificación de las claves públicas [177]. El más común se basa en una infraestructura de clave pública o PKI (Public Key Infrastructure) donde la confianza se delega a las autoridades certificadoras. Otro esquema interesante se basa en el modelo de confianza descentralizada conocido como "web of trust", donde cada usuarios tiene un conjunto de agrupaciones de claves públicas que son seguras. El modelo alternativo para no usar certificados es la criptografía basada en identidad, donde la clave pública de cada usuario es algún dato que identifique la identidad pública de ese usuario, por lo que los certificados de clave pública son innecesarios en este caso.

En [12] se presentan nuevos esquemas de autenticación punto a punto para redes inalámbricas descentralizadas. Concretamente se explica cómo utilizar una identificación demostrativa para llevar a cabo la pre-autenticación de nodos a través de canales de localización limitados. La identificación demostrativa proporciona una forma intuitiva para identificar y autenticar a los usuarios en una comunicación. Los esquemas descritos no requieren una infraestructura de clave pública, y permiten acabar con algunos de los problemas que afecta a los sistemas tradicionales de autenticación.

Entrando en el entorno de las VANETs, en [234] se presenta un esquema, denominado TACKs (Temporary Anonymous Certified Keys), como un medio eficaz para cumplir con las propiedades de seguridad y privacidad necesarias para la gestión de claves en este tipo de redes. En este esquema las OBUs utilizan claves de corta duración para firmar los mensajes enviados. Estas claves de corta duración están certificadas por las autoridades regionales o RA (Regional Authorities). Durante la actualización de las claves, las RAs verifican que el solicitante es una OBU legítima que no ha sido revocada, sin conocer su identidad para preservar su privacidad. Los certificados emitidos por las RAs sólo son válidos en su región local, por lo que las OBUs deben actualizar las claves al entrar en una nueva región. Cuando un conjunto de OBUs entran en una región, todas actualizan sus claves al mismo tiempo, evitando que posibles intrusos puedan realizar un seguimiento de los vehículos a través de estas actualizaciones.

La principal desventaja de la criptografía asimétrica es la alta complejidad computacional requerida lo que hace que la mayoría de criptosistemas de clave pública sean demasiados pesados como para ser utilizados en escenarios tan dinámicos como las redes móviles o las redes vehiculares dentro de la Internet de las Cosas. Para resolver estos problemas, en este capítulo se propone una combinación de criptografía simétrica y asimétrica para permitir el uso de claves secretas de sesión [49] en estos escenarios. En particular,

el resultado de la investigación llevada a cabo ofrece tanto la autenticación fuerte de nodos legítimos de la red mediante comunicaciones en abierto, como el intercambio de claves secretas compartidas entre pares de nodos, que pueden ser utilizadas como claves de sesión [162]. En la siguiente sección se detalla el esquema propuesto basado en demostraciones de conocimiento nulo en su modo no interactivo para evitar el sucesivo paso de mensajes a la hora de autenticar a los nodos.

2.2. Sistema no Interactivo

El sistema descrito en esta sección permite autenticar nodos de una red móvil inalámbrica descentralizada usando demostraciones de conocimiento nulo. Concretamente el esquema está basado en demostraciones de conocimiento nulo no interactivas, adecuadas para estos entornos donde los nodos se desplazan a grandes velocidades y no se puede presuponer un intercambio interactivo de mensajes [40]. De este modo, se aplica el protocolo de Diffie-Hellman para generar una clave de sesión compartida entre dos nodos con la mínima interacción posible.

2.2.1. Demostración de Conocimiento Nulo

Una demostración o prueba de conocimiento nulo es un proceso interactivo donde un probador convence a un verificador, hasta un nivel aceptable, que conoce, o tiene algún secreto, sin que el verificador pueda extraer ninguna información de la prueba que no pudiera haber extraído por cualquier otro procedimiento, con o sin la participación del probador o, incluso, si este miente en la prueba.

Para establecer un método de autenticación entre los nodos de la red, la propuesta realizada y explicada en este capítulo hace uso de un enfoque basado en la idea de las demostraciones de conocimiento nulo o ZKP (Zero Knowledge Proof) [93] que define un método para probar el conocimiento de un secreto sin revelar ninguna pista sobre él. Las pruebas de conocimiento nulo básicas están basadas en una serie de retos y respuestas que deben ser realizados en orden, mediante la interacción de los dos nodos de la red que se quieren autenticar, siguiendo el siguiente protocolo:

- $A \rightarrow B$: El usuario A quiere probar algo al verificador B y le envía algún elemento para su identificación.
- $B \rightarrow A$: El verificador B presenta un desafío a A.
- $A \rightarrow B$: El usuario A tiene que efectuar unos cálculos privadamente y enviar al verificador B una respuesta al desafío planteado.

Si alguna de las respuestas es incorrecta, B deduce que A no dispone del secreto y rechaza su identidad. Por el contrario, si en todas las etapas la respuesta es correcta, entonces B acepta que A conoce (o tiene) el secreto.

En el ámbito de las MANETs usadas en la Internet de las Cosas, una demostración de conocimiento nulo típica basada en sucesivos retos y respuestas implicaría un intercambio de sucesivos mensajes, lo que conllevaría tener que presuponer una conexión estable y continua entre los nodos [79]. En entornos tan volátiles como la Internet de las Cosas, donde existen dispositivos que se pueden mover a gran velocidad (como por ejemplo, los vehículos que conforman las redes vehiculares), un intercambio masivo de mensajes para realizar una demostración de conocimiento nulo puede ser inviable debido a los posibles fallos de conexión durante el protocolo. Para subsanar el problema de la multitud de mensajes bidireccionales que producen las pruebas de conocimiento nulo tradicionales han surgido en la bibliografía las demostraciones de conocimiento nulo no interactivas o Non-Interactive Knowledge Proof (NIZKP) [23], que condensan todos los retos en un único paquete enviado en un único mensaje. De esta forma, el tiempo que conllevaría el intercambio de mensajes para llevar a cabo el protocolo interactivo se minimiza.

La definición formal de una demostración de conocimiento nulo es como sigue: Sea V el verificador y P el probador. Si $\{0,1\}^*$ denota el conjunto de todas las cadenas, para un lenguaje $L \subseteq \{0,1\}^*$, un par de máquinas probabilistas de Turing (P,V), en donde P posee cierta potencia probabilista en tiempo polinomial y V posee cierta potencia determinista en tiempo polinomial, se considera como una demostración de conocimiento nula no interactiva de un lenguaje L si verifica la exactitud y seguridad frente a probadores y verificadores malintencionados mediante las siguientes condiciones:

- Totalidad: Para cualquier polinomio $p(\cdot)$ y entrada $x \in L$:

$$Pr[V(x, R, P(x, R)) = 1] \geq 1 - \frac{1}{p(|x|)} \tag{2.1}$$

- Solvencia: Para cualquier maquina interactiva de Turing P', polinomio $p(\cdot)$ y entrada $x \in L$:

$$Pr[V(x, R, P'(x, R) = 1] < \frac{1}{P(|x|)} \tag{2.2}$$

- Conocimiento Nulo: Para cualquier $x \in L$, existe un algoritmo probabilista en tiempo polinómico M que verifica:

$$V(x) = x, (R \in 0, 1^{c(|x|)}, P(x, R)) \approx_c M(x)_{x \in L} \tag{2.3}$$

Por tanto, la propiedad de Totalidad indica que si la afirmación a demostrar por un probador legítimo es verdadera, cualquier verificador legítimo

que siga el protocolo correctamente se convence de ello. La Solvencia significa que si la declaración es falsa, no existe probador malicioso que pueda convencer a un verificador legítimo de que es verdad, excepto con alguna probabilidad pequeña. El Conocimiento Nulo significa que si la afirmación es verdadera, ningún verificador malicioso aprende otra cosa más que este hecho. Esto se formaliza mostrando que todo lo que recibe de la interacción con el probador cualquier verificador malicioso lo puede calcular usando algún simulador que, tenga en cuenta sólo la declaración a ser demostrada y sin acceso al probador, pues dicho simulador puede producir una transcripción que parece una interacción entre un probador legítimo y un verificador malicioso.

Uno de los factores más importantes de cualquier demostración de conocimiento nulo es la elección del problema matemático que formalice las bases del esquema a utilizar. El trabajo propuesto en [92] muestra que bajo ciertas hipótesis de complejidad, por un lado, cualquier problema de tipo NP puede ser usado para definir una demostración de conocimiento nulo, y por otro lado que solo los problemas en tiempo polinómico BPP (Bounded-error Probabilistic Polynomial) pueden ser usados para describir una demostración de conocimiento nulo no interactiva.

En este trabajo de Tesis, el problema elegido para la base del diseño del esquema propuesto es el problema del isomorfismo de grafos. Un isomorfismo entre dos grafos es una biyección que preserva la relación de adyacencia, de manera que cualquier par de vértices en un grafo son adyacentes si y solo si tienen imagen en el otro grafo. El problema del isomorfismo de grafos consiste en determinar si dos grafos son isomorfos o no. Este problema ha sido usado en criptografía [92] [94] debido a que no se ha encontrado un algoritmo eficiente para resolverlo. En particular, para determinar si dos grafos con el mismo número v de vértices y el mismo número e de aristas son isomorfos, involucra un ataque de fuerza bruta porque requiere comprobar si alguna de las $v!$ biyecciones posibles preservan la adyacencia. En general, el problema del isomorfismo de grafos es uno de los pocos problemas en complejidad computacional que pertenecen a los problemas de tipo NP, pero no se conoce si pertenecen a los de subtipo P o los de subtipo NP-Completos [89]. Por lo tanto, este problema en función del tamaño y del tipo de grafos utilizados, puede ser muy difícil de resolver. En particular, se ha demostrado que el problema del isomorfismo de grafos pertenece a los problemas BPP, lo que permite que se puedan diseñar demostraciones de conocimiento nulas no interactivas basados en dicho problema.

En el trabajo de investigación propuesto en [82] se explica un método que transforma un protocolo interactivo en un protocolo no interactivo, que puede ser aplicado para convertir una prueba de conocimiento nulo interactiva en una prueba de conocimiento nulo no interactiva, gracias al uso de una función hash. Además, por una parte, como resultado teórico general, en el

trabajo de investigación [80] se presenta la primera demostración de conocimiento nula no interactiva para problemas NP cuya construcción se basa en permutaciones en un solo sentido y permutaciones trampas certificadas [140]. Por otra parte, como resultado práctico específico, el trabajo propuesto en [131] muestra una demostración de conocimiento nulo no interactiva basada en el problema del circuito hamiltoniano, que puede servir de base para justificar la elección del problema matemático utilizado en el esquema propuesto en la investigación detallada en el presente capítulo.

La propuesta que se detalla a continuación propone enviar el mensaje de autenticación como beacon en modo broadcast a la red en la que se utilice el esquema, para acelerar los procesos de autenticación [158].

2.2.2. Esquema de Autenticación

Para solventar el problema de la autenticación segura en un entorno móvil descentralizado, se ha propuesto un esquema basado en las pruebas de conocimiento nulo no interactivas, en el que ha mejorado la eficiencia de las comunicaciones mediante el envío de un único mensaje que sirva para verificar el conocimiento requerido [157]. De este modo, el algoritmo propuesto puede aumentar y disminuir el grado de seguridad a través de la eliminación o agregación de nuevos retos al mensaje de autenticación. Esto conlleva que la seguridad sea adaptable en función de las necesidades requeridas y en detrimento de la velocidad de verificación de la validación de esos retos. En particular, los parámetros usados en el esquema que se ha diseñado e implementado se muestran en la Tabla 2.1.

El grafo inicial que define a la red G y la clave de secreta del esquema Sol_G, que es la solución al problema computacional complejo en dicho grafo, deben ser conocidos por todos los usuarios legítimos de la red. Como problema complejo se ha elegido el circuito hamiltoniano porque dicho problema sobre grafos arbitrarios es considerado como un problema matemático difícil

Notación	Definición
G	Grafo conocido por todos los nodos legítimos de la red
Sol_G	Solución del problema complejo en G
Cha_i	$i-th$ reto propuesto por el verificador
G_i	Grafo isomorfo i del Grafo conocido G usado como compromiso
Iso_i	Isomorfismo entre G y G_i
Res_i	$i-th$ Respuesta correspondiente al reto Cha_i sobre el Grafo Isomorfo G_i
$h(\cdot)$	Función hash del esquema
$LSB(\cdot)$	Bit menos significativo de una cadena de entrada
$E_{k_i}(\cdot)$	Cifrado simétrico mediante clave k_i
$Subkey$	Contribución del nodo verificador a la clave de sesión

Tabla 2.1: Parámetros del Esquema

de resolver. El problema del circuito o ciclo hamiltoniano consiste en determinar si existe un camino en el grafo que visite cada vértice exactamente una vez. Este problema es a menudo considerado de tipo NP-completo, pero hay algunos grafos particulares para los que el problema es polinomial en tiempo de resolución o incluso lineal. Debido a esto, en el esquema diseñado se propone la utilización de grafos no planos. Un grafo plano es un grafo que puede ser dibujado en el plano sin que ninguna arista se cruce. Para comprobar si un grafo es planar se puede utilizar el Teorema de Kuratowski.

Teorema de Kuratowski. Un grafo es plano si y solo si no contiene un subgrafo isomorfo a una subdivisión elemental de K_5 (grafo completo de 5 vértices) o $K_{3,3}$ (grafo bipartito completo de 6 vértices).

Sin embargo, con el fin de comprobar en tiempo lineal si un grafo dado es o no plano, el Teorema 1 es el que se utiliza en el esquema:

Teorema 1. Para cualquier grafo plano con v vértices y e aristas: Si $v \geq 3$ entonces $e \leq 3v - 6$.

Por otra parte, se ha demostrado que la validez de las demostraciones de conocimiento nulo se basan en la suposición de una función criptográfica de hashing ideal [23]. Por lo tanto, en el esquema que se ha propuesto en esta memoria de Tesis se usa una función hash h que cumple con dicho requisito. La función hash elegida para el cálculo de los retos y las claves de cifrado de cada segmento del mensaje es el nuevo estándar de función hash SHA-3 [17] [19].

Por último, el sistema de cifrado simétrico seleccionado para cifrar los segmentos del mensaje del esquema es el cifrado en flujo usado en la cuarta generación de comunicaciones móviles LTE [73] [76], conocido como SNOW 3G [74]. Esta elección está basada en la complejidad computacional del sistema criptográfico elegido, que garantiza la eficiencia y rapidez de los procesos de cifrado y descifrado.

En el esquema diseñado, cada nodo emite en modo broadcast un mensaje o beacon para identificarse como nodo legítimo de la red. El mensaje está conformado por una serie de compromisos definidos por grafos isomorfos generados a partir de un grafo inicial conocido por todos los nodos legítimos de la red. Este grafo conocido G, representa a la red mediante la representación de los usuarios o grupos de usuarios a través de nodos. Cada compromiso que contiene el mensaje sólo puede ser accesible después de verificar el reto del compromiso anterior.

En particular, el mensaje se divide en *n+1* segmentos cifrados simétricamente con claves diferentes, excepto el primero que no está cifrado (ver Figura 2.1). Cada segmento contiene una tupla compuesta por una posible respuesta al reto de ese segmento y un grafo isomorfo G_i del grafo conocido G. De esa manera, un usuario legítimo de la red puede demostrar que es un usuario fidedigno y crear una sesión de comunicación con otro usuario si este último es capaz de descifrar todos los segmentos (retos) del mensaje enviado

por el primer usuario en modo broadcast, para acceder al último segmento que contiene su aportación a la clave de sesión. La clave de cifrado de cada segmento depende del segmento anterior, por lo que, aunque alguien quiera descifrar sólo el último segmento, no lo podrá conseguir sin descifrar todos los segmentos anteriores. Como se ha comentado antes, el nivel de seguridad del esquema depende del número de segmentos del mensaje, que representan diferentes retos. A mayor número de segmentos, más complejo será llegar al último segmento y obtener la información necesaria para el establecimiento de la clave de sesión compartida. Se puede utilizar el esquema de forma bidireccional para autenticar a los usuarios que quieran comunicarse, basándose en la idea del protocolo de Diffie-Hellman, donde ambos nodos compartirán una clave de sesión formada por subclaves intercambiadas a través del método descrito.

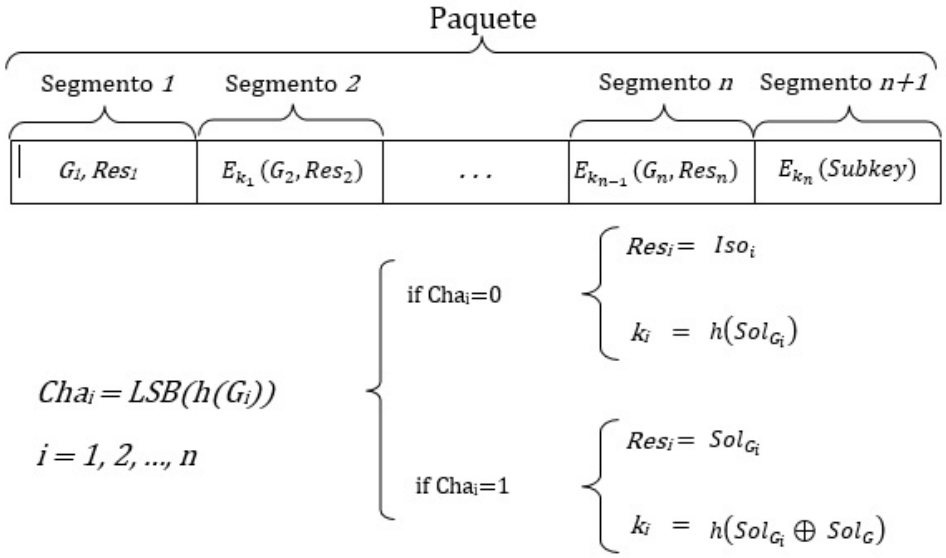

Figura 2.1: Mensaje Enviado Desglosado

Cada segmento contiene una copia de un grafo isomorfo del grafo original. Además, todos los usuarios legítimos deben conocer la función hash que utiliza el esquema, ya que se utiliza para calcular el reto a resolver sobre cada grafo isomorfo. Es más, la función hash se usa para definir la clave de cifrado de cada segmento del que se compone el mensaje.

Teniendo en cuenta los requisitos necesarios para la elección de la función hash, del problema matemático del esquema y del grafo inicial conocido por todos los usuarios legítimos, la probabilidad de que $Cha_i = 0$ es $1/2$. Por tanto, la probabilidad de que un usuario ilegítimo logre superar la prueba sin conocer la clave k_1 es $1/2$, la probabilidad de que logre superarla sin conocer las dos claves k_1 y k_2 es $1/2^2$, y la probabilidad de que la supere y no conozca

ninguna de las n claves k_1, k_2, ..., k_n es $1/2^n$.

Los retos que se proponen en el esquema están basados en el problema del isomorfismo de grafos, al igual que varias de las demostraciones de conocimiento nulo convencionales. Sin embargo, en la propuesta no interactiva realizada, los retos están definidos a partir del resultado booleano (bit menos significativo) de una función hash aplicada a cada grafo isomorfo generado como compromiso de cada segmento. Por ello, para cada reto, la respuesta se define como:

- Si $Reto = 0$, la respuesta es el isomorfismo entre el grafo isomorfo y el grafo inicial (Iso_i).
- Si $Reto = 1$, la respuesta es la solución al problema que define el esquema sobre el grafo isomorfo (Sol_{G_i}).

Las operaciones que un verificador debe realizar sobre el mensaje recibido son las siguientes:

1. Procesar el primer segmento del mensaje, que está en claro, sin cifrar.
2. Calcular, utilizando la función de hash, el reto que coincida con la información incluida en el segmento.
3. Comprobar si la respuesta se corresponde con el reto y el grafo isomorfo del segmento.
4. A partir del reto, calcular la clave que se utiliza para descifrar el siguiente segmento.
5. Repetir los pasos del 2 al 4 hasta llegar al último segmento, que una vez descifrado contiene la información necesaria para establecer el secreto compartido.

El diagrama de flujo que define el algoritmo diseñado y que debe ser ejecutado por el receptor o probador se muestra en la Figura 2.2.

2.3. Análisis

El esquema descrito está destinado para su uso en escenarios relacionados con Internet de las Cosas. Por lo tanto, este sistema ha sido implementado en Android y para la plataforma Android Wear, que son dos ejemplos de la proliferación de dispositivos inteligentes en esta nueva dimensión de Internet. Android es el sistema operativo para teléfonos inteligentes más popular, como se mostró en los preliminares de esta memoria de Tesis, con una cuota de mercado de más del 80 % en todo el mundo. Android Wear es el sistema

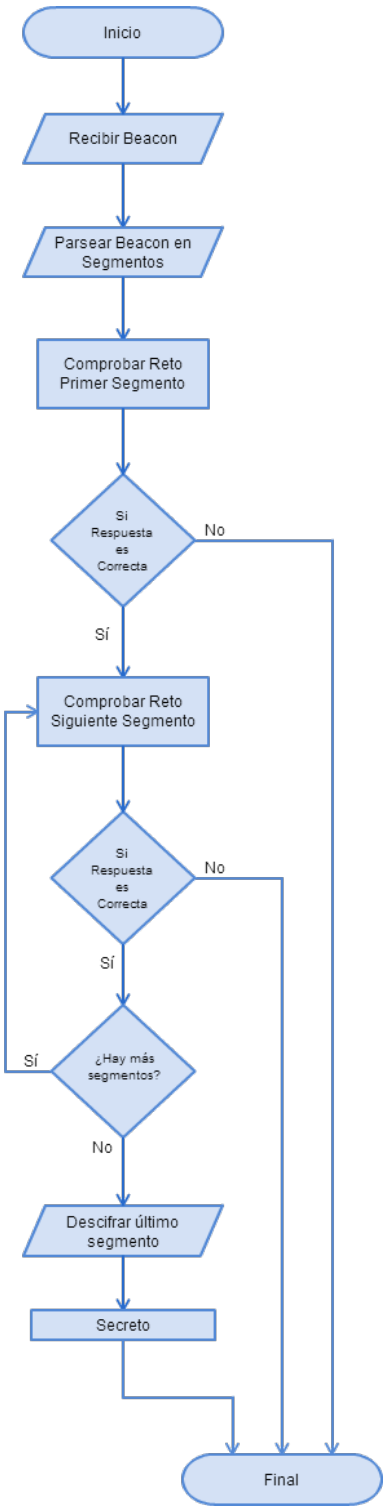

Figura 2.2: Diagrama de Flujo del Algoritmo Propuesto

operativo basado en Android para relojes inteligentes de la misma compañía. Android Wear tiene más de un 90 % de la cuota de mercado en los dispositivos de este tipo. Por lo tanto, todos los resultados que aquí se presentan son el resultado de la aplicación del sistema en estas dos plataformas pertenecientes a Android Open Source Project. El código fuente es de código abierto bajo un repositorio Git en la plataforma GitHub [150]. La Figura 2.3 muestra una captura de pantalla de la aplicación para Android que ha sido creada para analizar el rendimiento del sistema en los teléfonos inteligentes.

2.3.1. Implementación

El esquema ha sido diseñado e implementado siguiendo el paradigma de 'Secure by Default'. Se ha desarrollado dos algoritmos que permiten tanto generar un mensaje de tipo beacon con los retos y respuestas que se configuren, y otro algoritmo que permite descifrar un beacon recibido para obtener el secreto que guarda en el último segmento.

El pseudocódigo del algoritmo de generación de un mensaje tipo beacon se puede ver en el Algoritmo 1:

Algoritmo 1 Generación del Beacon

//Params: secret, The confidential data to share
//Params: nCha, Numbers of Challenges
//Params: graphNetwork, The graph network representation
//Params: hamiltonianCycle, The secret of the network representation
//Return: Beacon, The Authentication Beacon Message
function getBeacon (char[] *secret*, int *nCha*, int[][] *graphNetwork*, int[] *hamiltonianCycle*)

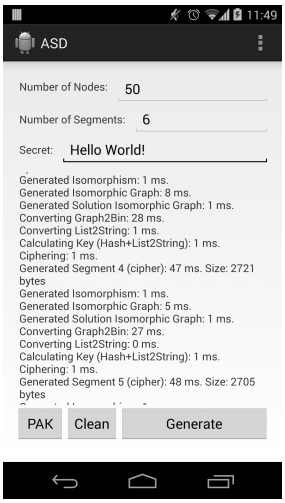

Figura 2.3: Captura de Pantalla de la Aplicación Android

```
01: var segs[]; // Stores the message segments
02: // First segment is not encrypted
03: // Isomorphic graph and the isomorphism
04: var gi, iso = genearteGi(graphNetwork);
05: var cha = LSB.hash(gi.getBytes());
06: var res; // Response to the challenge
07: var key[]; // Key to cypher next segment
08: // Hamiltonian Cycle Problem Solution over Isomorphic graph
09: var solgi = getHCPOverGi(graphNetwork, gi, iso, hamiltonianCycle);
10: if (cha == 0) // Challenge Type '0'
11:   // The isomorphism between both graphs
12:   res = iso;
13:   key[1] = hash(solgi);
14: else // Challenge Type '1'
15: // The hamiltonian cycle over Isomorphic graph
16:   res = solgi;
17:   key[1] = hash(solgi, hamiltonianCycle);
18: endif
19: segs[0] = createSegment(gi, res);
20: // The other segments are encrypted
21: for (int i = 1; i < nCha; i++){
22:   gi, iso = genearteGi(graphNetwork);
23:   cha = LSB.hash(gi.getBytes());
24:   solgi = getHCPOverGi(graphNetwork, gi, iso, hamiltonianCycle);
25:   if (cha == 0) // Challenge Type '0'
26:     res = iso;
27:     key[i+1] = hash(solgi);
28:   else // Challenge Type '1'
29:     res = solgi;
30:     key[i+1] = hash(solgi, hamiltonianCycle);
31:   endif
32:   segs[i] = Crypto.decrypt(createSegment(gi, res), key[i]);
33: }
34: return segs;
endfunction
```

Por su parte, el pseudocódigo de procesado del beacon una vez recibido se puede ver en el Algoritmo 2:

Algorimo 2 Procesado del Beacon

```
//Params: beacon, encrypted message segments
//Params: tseg, dimension of beacon segments
```

```
//Params: solg, solution in the original graph
//Return: subkey, contribution to the key
function getSecret (char[] beacon, int tseg, char[] solg)
01: var segs[]; // Stores the message segments
02: // Message is divided into tseg − size segments
03: segs = beacon.splitByTam(tseg);
04: // Isomorphic graph and response
05: // First segment is not encrypted
06: var gi = getGi(segs[0]);
07: var res = getRes(segs[0]);
08: // The challenge is computed
09: var cha = LSB.hash(gi.getBytes());
10: // Check whether the response is correct
11: if (res != response(gi, cha))
12:   return; // If not correct, abort
13: endif
14: // The solution is obtained in gi
15: var sol = solve(gi);
16: // ki is the encryption key of the next segment
17: var ki = cha ∗ hash(sol) ⊕ cha ∗ hash(sol ⊕ solg)
18: var decryption;
19: // The following steps are repeated
20: for (int i = 1; i < segs.size() − 1; i + +){
21:   // The segment is decrypted with the key ki
22:   decryption = Crypto.decrypt(segs[i], ki);
23:   gi = getGi(decryption);
24:   res = getRes(decryption);
25:   cha = LSB.hash(gi.getBytes());
26:   if (res != response(gi, cha))
27:     return;
28:   endif
29:   sol = solve(gi);
30:   ki = cha ∗ hash(sol) ⊕ cha ∗ hash(sol ⊕ solg)
31: }
32: // Decryption of the last segment provides the
33: // contribution to the shared key.
34: return Crypto.decrypt(segs[segs.size()], ki);
endfunction
```

2.3.2. Eficiencia

Una de las premisas que debe cumplir el esquema es que debe ser ligero en términos de tamaño y velocidad de computación, debido a las peculiaridades de los dispositivos que se van a utilizar. Por lo tanto, el tamaño del mensaje o paquete debe ser tan pequeño como sea posible para utilizar el menor espacio posible en la memoria y para tener una comunicación rápida entre dispositivos. Debido a esto, el formato de almacenamiento de cada uno de los elementos que componen el mensaje ha sido optimizado. De esa manera, se han diseñado una serie de estructuras para serializar y empaquetar los distintos datos del beacon.

El caso más crítico es el envío de los grafos de forma reducida, ya que son los datos que ocupan mayor tamaño. Los grafos son serializados a través de sus matrices de adyacencia representadas por números enteros, debido a que esta es su conversión más rápida y óptima. Se ha generado una versión adicional de este empaquetado mediante el uso de códigos hexadecimales en lugar de números enteros puros. Sin embargo, tras analizar los resultados, y aunque el tamaño de almacenamiento requerido es más pequeño, la velocidad de serialización es bastante más lenta. Por lo tanto, finalmente se ha optado por una serialización más rápida en detrimento de un tamaño más reducido.

Dicha serialización realizada se muestra a continuación mediante un ejemplo.

Dada la matriz de adyacencia para un grafo dado de dimensión 5:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

El sistema lo representa como un array de una sola dimensión:

$$0110110100110010000110110$$

Esta representación unidimensional es convertida a números enteros separados por el carácter ','. De manera que para el ejemplo dado, se obtendría como serialización:

$$13, 20, 25, 1, 22$$

En lo referente a las respuestas de los retos que se envían junto a los grafos, estas también han sido optimizadas y comprimidas en un objeto más pequeño que la representación natural que tendría. Por tanto, las respuestas

o soluciones se empaquetan mediante el uso de listas de números enteros separados por el carácter ','. Esta elección se ha realizado en base a que los dos posibles tipos de soluciones: el circuito hamiltoniano y el isomorfismo entre los grafos, se pueden representar como listas. Por ejemplo, si tenemos un isomorfismo definido por:

$$1 \to 2$$
$$2 \to 5$$
$$3 \to 1$$
$$4 \to 3$$
$$5 \to 4$$

se pueden utilizar los índices o la posición de cada elemento de la lista para denotar a los nodos de partida. En este caso, los valores almacenados en cada posición representan los nuevos nodos producto del isomorfismo generado. De esa manera, finalmente, se obtendría el dato siguiente:

$$2, 5, 1, 3, 4$$

Respecto a los segmentos que conforman cada beacon, se serializan usando códigos hexadecimales. Destacar y recordar que todos los segmentos, exceptuando el primero, están cifrados como se ha explicado en las secciones anteriores. Por ejemplo, para el caso de un segmento que contenga el grafo isomorfo anterior y su reto y respuesta correspondiente, su representación hexadecimal sería:

$$A324D0E3F19$$

Finalmente, el mensaje o beacon a enviar de forma constante, se resume como una concatenación de cada segmento separados mediante el carácter '|'. Siguiendo con los ejemplos anteriores, si iteramos el algoritmo 2 veces más, para obtener un total de 3 grafos isomorfos distintos, obtendríamos un mensaje similar al siguiente:

$$A324D0E3F19|F9223B3EE34|DC34F212ACB$$

Una vez determinada la estructura de los datos a enviar en forma de beacons, se detalla y evalúa a continuación los tamaños finales que ocupa cada mensaje formateado. Teniendo en cuenta el formato que se ha utilizado para representar los elementos que forman parte del sistema propuesto, se han alcanzado tamaños de paquete muy satisfactorios.

El tamaño del mensaje está dado por la dimensión del grafo secreto utilizado para representar la red. El espacio requerido va en proporción al número

de nodos que posea dicho grafo. Esto implica que cada segmento generado es mayor para un mayor número de nodos, lo que provoca un aumento de la dimensión total del mensaje.

Por lo tanto, hemos analizado el tamaño óptimo por segmento en función del número de nodos del grafo. Los resultados se pueden ver en la Figura 2.4.

Dicha figura revela una tendencia polinómica de orden 2 que representa la relación entre el tamaño de segmentos y el número de nodos del grafo de red. Extrapolando los datos, hemos obtenido una función polinómica (ver ecuación 2.4), que define el comportamiento del tamaño de los segmentos. Con dicha función se puede estimar el tamaño de segmento para grafos de cualquier dimensión.

$$y = 0,9765x^2 + 5,8046x - 1,1513 \tag{2.4}$$

De esa manera, por ejemplo, para un esquema en el que el grafo de red se estipula de dimensión 50, y el número de retos que define el beacon a enviar se establece en 6, se requiere un tamaño de paquete de 16 Kilobytes, sin incluir el tamaño del secreto a compartir. Por lo tanto, con el fin de enviar de forma segura un mensaje en un único paquete mediante el uso de este sistema, se ha concluido que la elección anterior de parámetros es la más idónea, en cuanto a relación seguridad, dimensión del paquete. Con un overhead de 16 Kilobytes, se obtiene un sistema de autenticación basado en una demostración de conocimiento nulo no interactiva de grado 6, es decir, con 6 iteraciones del protocolo propuesto.

Por otro lado, también se ha analizado el tiempo de cómputo requerido por los dispositivos inteligentes para crear los beacons. Este tiempo depende del número de segmentos totales del mensaje. Por lo tanto, aunque el número de segmentos influye de manera positiva en el nivel de seguridad del esquema, también repercute negativamente en el tiempo de cálculo requerido

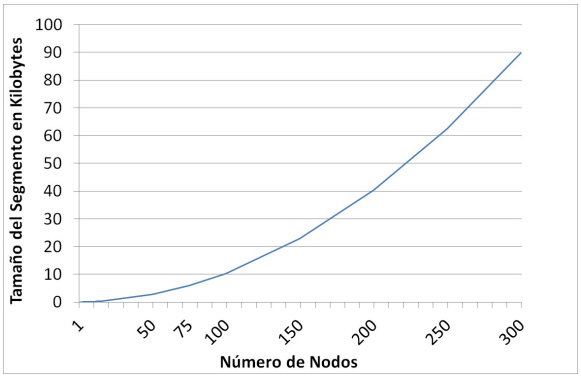

Figura 2.4: Tamaño de Segmento vs. Tamaño del Grafo

por el microprocesador para generar el paquete. Para realizar estas pruebas en entornos reales, se han seleccionado una gama variopinta de dispositivos inteligentes. Esta elección se ha tomado en base a tres posibles rangos en el que se segmenta el mercado: dispositivos de bajo coste, dispositivos de coste medio y dispositivos de gama alta. Tras esta decisión los modelos elegidos han sido: Motorola Moto G, Samsung Galaxy S3, LG Nexus 5. Esta selección se ha tomado tanto para verificar la eficacia del sistema en dispositivos con diversas capacidades de procesamiento, desde capacidades tope de gama hasta limitadas. Además, también se han generado las pruebas usando relojes inteligentes, con Android Wear como sistema operativo. Específicamente se han elegido el LG G Watch y el Samsung Gear, dos de los modelos pioneros en el sector de los smartwatches. Tras realizar las pruebas en estos relojes inteligentes, se ha concluido que como actualmente estos relojes dependen de un teléfono inteligente, al que se conectan por BLE (Bluetooth Low Energy), la generación de los beacons se realiza del lado del smartphone, aunque se lance desde el propio smartwatch.

Teniendo en cuenta todas estas características, el lenguaje de programación que se ha utilizado ha sido Java a través del SDK de Android, ya que es el lenguaje nativo que se recomienda para programar aplicaciones y servicios en dichos sistemas operativos. Después de hacer docenas de experimentos, bajo diversas circunstancias, se ha elaborado un promedio de los resultados. El tiempo requerido para la generación de cada segmento en función del número de nodos del grafo de red se muestra en la Figura 2.5.

Sin embargo, los resultados mostrados en la Figura 2.5 están fuertemente condicionados por el proceso de serialización de los grafos. De hecho, este proceso requiere mucho más tiempo de cálculo que la suma de todos los pa-

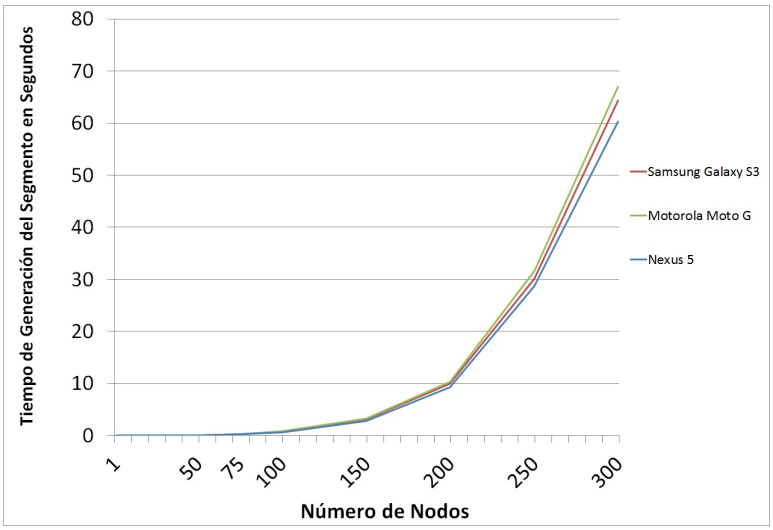

Figura 2.5: Tiempo de Generación de Segmento vs. Tamaño del Grafo

sos restantes del esquema. No es sorprendente que el proceso de serialización para grandes grafos requiera más de 98 % del tiempo necesario para construir el paquete. Los primeros pasos del proceso de serialización eran casi 15 veces más lento. Convirtiendo la matriz de adyacencia en caracteres hexadecimales, se mejoró y optimizó este rendimiento. Después de una nueva optimización del proceso, se concluyó que es más eficiente usar números enteros mediante las características de las clases de Java, y no utilizar códigos hexadecimales para serializar los grafos. El problema es que ahora el espacio ocupado es mayor que en el proceso anterior que usa códigos hexadecimales, pero la velocidad de cálculo ha mejorado casi 20 veces más. Aún así, el proceso sigue siendo muy lento en comparación con otras operaciones del sistema, debido a las limitaciones de la máquina virtual de Java. Por ejemplo, para un grafo definido por 300 nodos, la serialización de dicho grafo tarda 59,138 ms de media, frente a la generación completa del segmento que utiliza este grafo, que tarda alrededor de 60,345 ms de media. Estos resultados pueden mejorarse sustancialmente utilizando el NDK de Android, que permite programar a bajo nivel, evitando la máquina virtual de Java. Para desarrollar en dicho stack tecnológico, se debe usar C, un lenguaje compilado que mejora el rendimiento a gran escala frente a los lenguajes interpretados como Java.

A partir de los resultados de la Figura 2.5, se ha interpolado la tendencia polinómica, de orden 4, asociada (véase la Ecuación 2.5). Esta ecuación representa el tiempo de generación de los segmentos en función del número de nodos totales del grafo. De esta manera, esta ecuación estima el tiempo total requerido para crear los beacons.

$$y = (5\text{E-}6)x^4 + (23\text{E-}4)x^3 - 0,537x^2 + 30,548x - 227,48 \tag{2.5}$$

Por ejemplo, para un esquema con un grafo de red de dimensión 50, y un número de retos establecido en 6, el tiempo requerido para construir el beacon es de 300 ms. Por lo tanto, un dispositivo móvil con capacidades limitadas podría generar fácilmente paquetes para el esquema propuesto, como se demuestra en estos resultados. Se mejoraría la eficiencia si se usaran técnicas de caché en la generación de estos paquetes, de manera que se podría lanzar en segundo plano un hilo de generación de paquetes que fuesen almacenándose en caché para ser utilizados de forma inmediata cuando fuera necesario.

Finalmente, se ha analizado el tiempo que conllevaría a los dispositivos móviles descifrar el paquete que reciben de otros dispositivos móviles mediante broadcast. Estos tiempos de procesamiento de paquetes son realmente cortos, como demuestra los resultados de la Figura 2.6.

Después de analizar los resultados anteriores, se ha concluido que la tendencia de los datos sigue una función polinómica de orden 4 (véase la Ecuación 2.6). Esta ecuación se puede usar para estimar el tiempo de procesa-

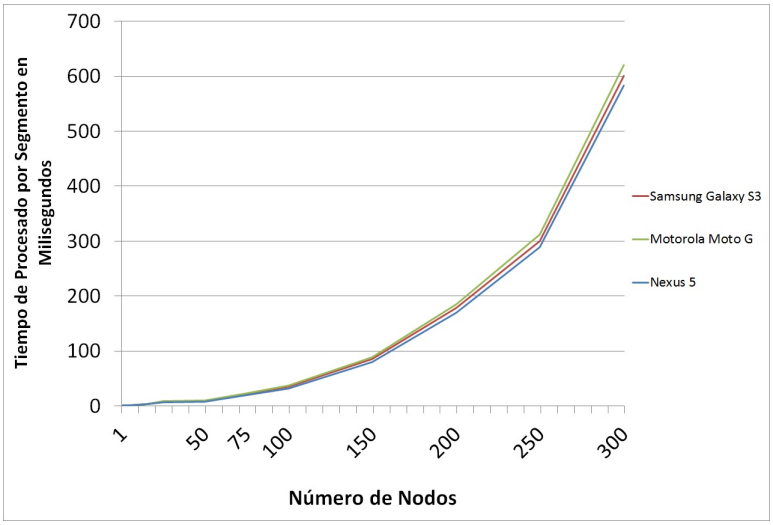

Figura 2.6: Tiempo de Procesado de Segmento vs. Tamaño del Grafo

miento por segmento en función de los nodos del grafo que representa a la red.

$$y = (2\text{E-}9)x^5(-1\text{E-}6)x^4 + (2\text{E-}4)x^3 - 0,0184x^2 + 0,673x - 1,7392 \quad (2.6)$$

Concretamente, estos datos obtenidos reflejan que, por ejemplo, un esquema en el que el grafo de red tiene dimensión 50, y el número de retos a utilizar en el esquema es de 6, el tiempo que un dispositivo móvil necesita para procesar un paquete recibido y obtener el secreto es sólo de 48 ms.

2.3.3. Comparación con Esquemas ZKP

En la literatura reciente, se hace ardua la tarea de encontrar esquemas propuestos para la autenticación de dispositivos móviles específicamente en el ámbito de Internet de las Cosas.

Muchos trabajos de investigación existentes proponen esquemas de autenticación ligeros basados en demostraciones de conocimiento nulo, pero ninguno de estos esquemas propuestos se ha llegado a implementar por completo. Muchos autores omiten una de las piezas más importantes de información que pueden caracterizar un esquema de este tipo: el tiempo requerido para la autenticación. Sólo unos pocos autores han analizado la velocidad de transmisión, el porcentaje de paquetes que se pierden en los esquemas de tipo interactivo y el tiempo total que se requiere en entornos reales. En el trabajo aquí descrito, se han analizado los parámetros de configuración del esquema propuesto, sin evaluar el porcentaje de paquetes perdidos debido a que el

		Conf1	Conf2	Conf3	Conf4	Nuestro Esquema
10 Retos	Time	469	1302	484	1522	454
	Size	4045	4045	4045	4045	17,826
100 Retos	Time	3422	8070	3703	9824	5665
	Size	39,595	39,595	39,595	39,595	187,132

Tabla 2.2: Datos de la Comparativa: Tiempo (ms) y Tamaño (bytes).

tipo de esquema diseñado no es interactivo, enviándose un sólo mensaje sin que existan pérdidas de paquetes intermedios.

Se han analizado varios esquemas de autenticación ligera que se puedan aplicar a escenarios de Internet de las Cosas. Por ejemplo, uno de estos esquemas es un trabajo de investigación que propone, analiza y evalúa un algoritmo de autenticación basado en una demostración de conocimiento nulo mediante grafos isomorfos [101]. En dicho trabajo, los autores han diseñado un mecanismo que permite realizar la autenticación configurando distintos niveles de confianza y seguridad. La implementación la han realizado sobre equipos informáticos convencionales, con distinto procesamiento hardware, de manera que las condiciones de partida son más potentes que las condiciones del esquema propuesto en este capítulo, donde se utilizan meros dispositivos móviles. Otro de los aspectos a destacar del trabajo descrito en [101], es que utilizan grafos aleatorios de 41 nodos de dimensión. Debido a esto, para la comparación, se ha configurado nuestro esquema con grafos de 41 nodos. Los resultados de la comparación se muestran en la Tabla 2.2, donde se muestran las cuatro configuraciones hardware utilizadas por el esquema comparado [101]. Por último, hacer notar que dicho esquema utiliza pruebas de conocimiento nulo de tipo interactivas que involucra un número de intercambios de mensajes por cada nodo, como retos tenga el protocolo configurado. Nuestra propuesta, al diseñarse como un esquema no interactivo, sólo requiere un único intercambio de mensaje, sea cual sea el número de retos del esquema.

De la Tabla 2.2 se puede concluir que el esquema que se ha diseñado aquí es computacionalmente más rápido teniendo en cuenta los resultados medios y las características técnicas de ambas implementaciones. Por otra parte, el esquema con el que se ha comparado utiliza menos bytes de memoria, al no agrupar todos los retos y respuestas en un sólo mensaje. Sin embargo, analizando la relación entre la memoria utilizada y el tiempo requerido, se puede entrever que nuestro esquema tiene un mejor rendimiento. Esto se debe a que el espacio de memoria requerido de más por nuestro esquema, es tan sólo de unos pocos kilobytes, cantidad que es aceptable para los dispositivos móviles de hoy en día, tanto a la hora de enviar estos datos mediante tecnologías inalámbricas como a la hora de ser almacenados y procesados.

2.3.4. Comparación con Esquemas Diffie-Hellman

También se han realizado comparaciones con otros esquemas cuyo objetivo era similar. Uno de estos esquemas elegidos para la comparación, fue seleccionado debido a su popularidad, su aplicabilidad a entornos similares y el uso del protocolo de Diffie-Hellman [67] para el intercambio de claves. Dicho método permite autenticar a dos nodos, utilizando claves de sesión compartida de manera autenticada, evitando la mayor vulnerabilidad del protocolo de Diffie-Hellman: los ataques Man in the Middle.

Después de una extensa revisión de la bibliografía, el esquema elegido para la comparación que cumplía los requisitos anteriores fue el famoso intercambio Diffie-Hellman PAK (Password Authenticated Key). El esquema propone añadir autenticación mutua basada en contraseñas memorizadas utilizando como base el esquema de intercambio de claves no autenticadas de Diffie-Hellman. PAK permite que dos nodos se autentiquen mediante un protocolo seguro de intercambio de claves de forma autenticada, lo que garantiza la confidencialidad y la robustez contra ataques de tipo diccionario sobre las claves utilizadas.

Para la comparación, el esquema PAK se ha implementado como parte de este trabajo de Tesis para dispositivos móviles con sistema operativo Android Wear o Android. Hasta ahora, el esquema PAK no había sido implementado en plataformas móviles, por lo que este trabajo proporciona la primera implementación del esquema PAK en la plataforma más popular de dispositivos móviles. El código fuente de la implementación en estas plataformas para el esquema propuesto en este documento y para el esquema PAK, ha sido liberado como software libre y está disponibles en un repositorio público alojado en la famosa plataforma GitHub [150]. En las pruebas realizadas se ha utilizado como secreto a compartir en el protocolo de autenticación entre dos nodos, el típico Hello World! ampliamente utilizado en entornos de programación.

En particular, a fin de comparar el esquema propuesto con el esquema PAK, se ha usado la siguiente configuración:

- Grafos de dimensión 41, que son suficientemente grandes como para considerar seguro al esquema. Por lo tanto, un ataque sobre un grafo isomorfo de esta dimensión involucra 41! iteraciones, un número inviable de iteraciones necesarias para hackear un esquema de este tipo.
- Paquetes de mensajes con 3, 4, 5, 6, 7 y 8 retos.
- El tiempo evaluado en estas comparaciones incluye tanto la generación del mensaje a enviar por el nodo emisor, como el procesado de dicho mensaje por el nodo receptor.

Los resultados de la comparación se muestran en la Tabla 2.3. De esta manera, se puede concluir que, en general, el esquema propuesto en esta

Esquema PAK	Nuestro Esquema	
Time (ms)	Retos	Tiempo (ms)
197	3	86
	4	112
	5	153
	6	176
	7	195
	8	221

Tabla 2.3: Esquema PAK *vs.* Esquema Propuesto.

memoria de Tesis tiene un rendimiento similar al famoso esquema PAK. Además, cabe destacar que incluso en algunos casos, el esquema propuesto mejora ligeramente los resultados del esquema PAK.

2.3.5. Aplicaciones

Todas las aplicaciones que han sido implementadas y analizadas en esta subsección para probar la eficacia del sistema propuesto son para un entorno descentralizado, donde se utilizan Wi-Fi Direct y/o Bluetooth Low Energy para las comunicaciones inalámbricas. Esto se debe a que muchos de los objetos interconectados de hoy en día vienen integrados con estas tecnologías y están disponibles para Android. En el futuro, se podrá hacer uso de otros estándares como el LTE-Direct, que está en sus primeras fases de gestación.

En particular, el esquema es apropiado para aquellos casos donde se haga necesario un grado de confidencialidad basado en comunicaciones cifradas con una clave de sesión compartida.

La distribución y gestión de credenciales para la comunicación segura puede ser una tarea relativamente simple si se considera la existencia de un sólo grupo restringido de proveedores de aplicaciones centralizadas. En cambio, en una arquitectura distribuida, como el de la Internet de las cosas, surgen muchos más problemas, como se explica en [219], ya que cualquier dispositivo puede estar conectado a cualquier otro en cualquier momento, y sin haber tenido un contacto previo. Por lo tanto, en este escenario, la gestión de claves se convierte en un problema importante. Una posible solución se basa en el uso de un esquema como el propuesto en este capítulo, ya que es bastante flexible y adaptable a las necesidades de los dispositivos que interactúan en la Internet de las Cosas.

En lo referente a las aplicaciones para entornos móviles aplicados a MANETs, es posible aplicar el esquema para su uso en múltiples escenarios que se definen a continuación.

Un uso interesante de la propuesta diseñada es su utilización en transacciones comerciales para MANETs, ya que en ese escenario, un nodo de red

legítimo puede querer compartir sus recursos con otros nodos legítimos de manera comercial. Debido a su naturaleza móvil, los nodos de una MANET no tienen acceso a Internet en muchos lugares. Por tanto, aquellos nodos autorizados que tienen una conexión a Internet pueden ofertar su conexión a otros nodos de la red. Para esta tarea, ambos nodos pueden establecer una clave de sesión secreta y compartida, mediante el esquema propuesto en este documento.

De este modo existen diversos escenarios diferentes aplicados a MANETs para el uso de las variantes del esquema propuesto. Por un lado se puede utilizar para reportar información autenticada de forma unidireccional ya sea a través de la utilización de claves secretas o del uso de un esquema de clave pública. Esto conlleva que la única posibilidad para transmitir información sea de forma autenticada (véase la figura 2.7) y así el resto de nodos pueden capturar estos mensajes emitidos, y descifrarlos gracias a su conocimiento sobre la clave de red secreta utilizada para generar los retos y las respuestas. Desarrollando esta idea de manera más específica, el sistema se puede utilizar para la notificación o difusión de publicidad de los negocios comerciales y resto de tiendas. La transmisión de publicidad utilizando el esquema propuesto implica que sólo los nodos autorizados pueden enviar cierta publicidad, lo que evita el spam masivo de nodos que no pertenecen a la red. Por otro lado, otro de los casos de uso atañe a la posibilidad de que un nodo legítimo difunda su clave pública al resto de nodos pertenecientes a la red (véase la figura 2.7). Este caso de uso requiere la aplicación de otra variante del esquema propuesto, donde los mensajes enviados contienen en su último segmento la clave pública del nodo emisor. Gracias a esto, sólo los usuarios de la red pueden acceder a la clave pública del remitente, ya que los retos y las respuestas se basan en una clave de red secreta que sólo es conocida por los usuarios legítimos. Este enfoque puede ser utilizado cada vez que la difusión de un evento a través de una MANET requiera el uso de un esquema de firma digital, para compartir claves públicas de forma autenticada y restringida.

En el entorno de las redes vehiculares, una VANET puede ser vista como un tipo especial de MANET donde los nodos móviles son vehículos y el objetivo principal es evitar las circunstancias adversas en las carreteras y lograr una gestión más eficaz del tráfico. Uno de los objetivos más importantes en el diseño de este tipo de redes es conseguir la resistencia frente a ataques de seguridad [137]. Por tanto, en el área de las VANETs, el esquema propuesto puede ser utilizado para autenticar los vehículos en zonas aisladas (zonas de montaña, túneles, etc.), donde la conexión a Internet no está disponible. Cada vehículo puede enviar un mensaje de autenticación para negociar una clave compartida, siguiendo el esquema propuesto acorde al protocolo de Diffie y Hellman, y realizar así futuras comunicaciones seguras en estos entornos. Además, la propuesta se puede aplicar para solucionar otros pro-

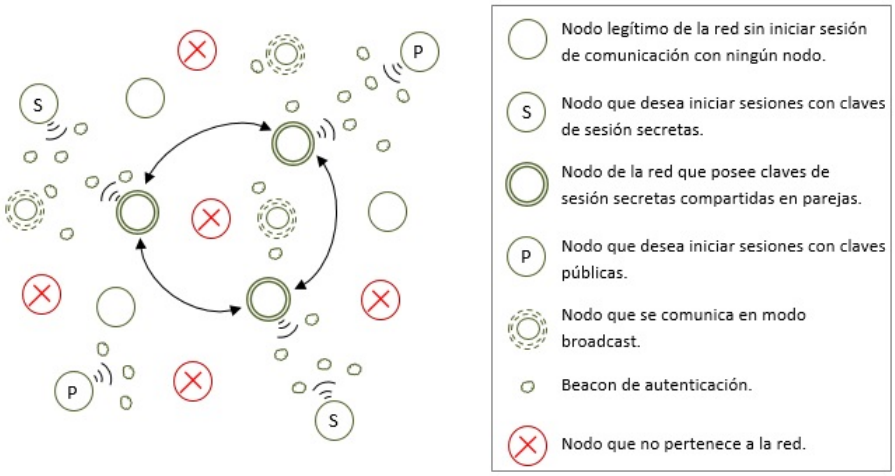

Figura 2.7: Tipos de Nodos de una MANET

blemas en redes vehiculares. Por ejemplo, se puede utilizar para autenticar la información enviada desde los semáforos inteligentes hacia los vehículos. Algunos entornos críticos exigen que esta información se envíe autenticada, de modo que sólo los usuarios legítimos de la red pueden procesarla. No en vano, existen propuestas propias que también se han diseñado a lo largo de la Tesis, como [159] que describe una solución de bajo costo para semáforos inteligentes caracterizados por sensores de luz que proporcionan información en tiempo real sobre el color del semáforo. Dicha solución se describe en el último capítulo de la Tesis, donde la notificación del semáforo se envía de forma autenticada a través de la propuesta explicada en este capítulo.

Otro de los entornos en el que el esquema es sumamente útil, es en las redes de sensores. Especialmente para su aplicación en redes de sensores inalámbricas o WSNs (Wireless Sensor Networks). Las WSNs han evolucionado significativamente en los últimos años, generando un área de investigación prometedora sobre una tecnología fructífera y útil. Esta tecnología consiste en dos tipos de entidades: los nodos o sensores y las estaciones base. En general, las estaciones base son más potentes que los sensores, pero esta tendencia está cambiando gracias a los avances de la tecnología de baja potencia.

Hoy en día, es posible crear una red sensorial usando plataformas como Arduino, Raspberry Pi, Odroid, Intel Edison, etc., que pueden tener varios tipos distintos de sensores. Dado que algunas aplicaciones para las redes de sensores inalámbricas sólo requieren la información de un sensor específico, el sistema propuesto puede ser utilizado en estas plataformas para enviar la información de los sensores de forma independiente y autenticada mediante el uso de un mensaje de difusión que envíe los datos de cada sensor. Dichos datos enviados sólo serán accesibles por usuarios legítimos de la propia red.

La aplicabilidad de la solución propuesta está pensada más hacia plataformas de sensores, como las que se pueden crear mediante Arduino o cualquier otra placa de estas características, que a sensores individuales. Esto es debido a que el sistema operativo para el que se ha decidido programar los resultados de las investigaciones de la Tesis, Android, se puede instalar en estas placas de forma trivial. De esta manera, se pueden crear frameworks o APIs bajo estas placas, para realizar comunicaciones seguras basadas en el esquema propuesto.

Además, el esquema de autenticación puede ser utilizado para complementar otras soluciones y añadir una capa de seguridad extra. Por ejemplo, los modelos de redes de sensores actuales basados en parámetros precargados, como la transmisión de las claves de los sensores, se pueden mejorar mediante el esquema que se presenta en este capítulo para generar claves compartidas y usarlas como claves de sesión de los sensores.

2.3.6. Seguridad

Varios son los ataques más frecuentes a entornos inalámbricos y descentralizados. A continuación se analizan varios ataques conocidos que pueden afectar a la propuesta descrita.

- Ataques a operaciones criptográficas. La seguridad del esquema depende de la función hash elegida. Un ataque de colisión a una función hash criptográfica trata de encontrar dos entradas que producen el mismo valor hash. Otra posible debilidad puede ser un ataque de imagen inversa, que trata de encontrar un mensaje que tiene un valor hash específico. Por ello, la función hash criptográfica debe ser resistente frente a ataques tanto de imagen inversa como de la colisión. En este trabajo, se utiliza la función hash SHA-3 para realizar varias operaciones criptográficas. El algoritmo base de SHA-3 es la función Keccak. Con Keccak es posible configurar el nivel de seguridad en función de la capacidad del atacante. Este enfoque es similar al sugerido por el Instituto Nacional de Estándares y Tecnología o NIST (National Institute of Standards and Technology) [13]. Por otro lado, la seguridad de la propuesta depende del algoritmo de cifrado simétrico utilizado para cifrar cada segmento. El cifrado utilizado en el sistema es la base de la seguridad de las comunicaciones LTE, llamado Snow3G. Snow3G está publicado bajo una versión portable de Verilog, con una versión disponible para hardware denominada VHDL (Verilog Hardware Description Language), que permite a los clientes llevar a cabo una revisión del código interno para garantizar su seguridad. Por tanto, la seguridad de las operaciones criptográficas está garantizada gracias a los estándares utilizados: SHA-3 y Snow3G. La construcción esponja ha demostrado ser inmutable en ciertas condiciones [16]. Esto también es aplicable a la

función Keccak que utiliza la propuesta, dado que es la base del estándar SHA-3. Además, existen valores óptimos para Keccak que aseguran la resistencia a colisiones, a imágenes inversas y a segundas imágenes inversas, en el modelo de permutación ideal [6]. En cuanto a Snow3G, su complejidad computacional es de tiempo lineal, lo que garantiza la eficiencia durante el proceso de cifrado y de descifrado. Las pruebas de seguridad de Snow3G se basan en el supuesto de que este sistema de cifrado se comporta como una función aleatoria perfecta de la clave [126].

- Ataques al problema de grafos. El problema utilizado en una ZKP es fundamental para la seguridad de todo esquema de este tipo. El esquema implementado utiliza dos de los problemas más difíciles que existen sobre grafos aleatorios: el isomorfismo de grafos y el circuito hamiltoniano. En cuanto a posibles ataques al problema del isomorfismo de grafos, se han propuesto algunos algoritmos eficientes para algunos grafos específicos. Por ejemplo, en [10], explican varios algoritmos probabilísticos; y en [60] se describen algoritmos más específicos para determinar si dos grafos son isomorfos o no. Sin embargo, en el sistema propuesto se pueden usar grafos de diferentes tamaños, lo que permite aumentar el nivel de seguridad en función del tamaño de estos grafos. Si los grafos generados son completamente aleatorios, ninguno de los algoritmos propuestos es útil para encontrar la solución a los retos que se utilizan. Esto conlleva que bajo la hipótesis de la selección de grafos idóneos, el problema del isomorfismo de grafos puede considerarse NP-completo para la propuesta. Con respecto al problema del circuito hamiltoniano, bajo las condiciones del esquema definido, el problema puede ser también considerado NP-completo. Esto se logra con el uso de grafos no planares. A través del teorema explicado en secciones anteriores, podemos asegurar que ninguno de los grafos que se generan en el esquema son planares. Una vez más, bajo la hipótesis de la selección idónea de grafos, el problema del circuito hamiltoniano puede considerarse NP-completo para la propuesta. Por tanto, no es posible realizar un ataque sobre la base de los problemas utilizados, debido a que según su complejidad, dichos problemas elegidos se pueden considerar NP-completos.

- Ataques Man in the Middle. Un ataque Man in the Middle (MitM) se produce cuando un atacante de forma secreta altera la comunicación entre dos partes que creen que están comunicándose directamente entre sí. Este es uno de los esquemas de ataque más utilizados en redes inalámbricas. En el caso de la propuesta, un ataque MitM no puede llevarse a cabo debido a que no hay manera de obtener información durante la transacción. En particular, el esquema utiliza un solo mensaje

para enviar información desde el emisor al receptor. Por tanto, no existe comunicación interactiva entre dos usuarios, impidiendo que usuarios malintencionados puedan interceptar la comunicación y suplantar las identidades en las comunicaciones. Si se utiliza el esquema para establecer una clave secreta a través del algoritmo de Diffie-Hellman, la propuesta es robusta frente a ataques de este tipo, gracias al hecho de que el protocolo utiliza autenticación mutua con claves secretas. En consecuencia, un ataque MitM no tendría éxito. El atacante podría interceptar los mensajes, pero sin acceso a los parámetros secretos del esquema no puede obtener ninguna información confidencial. Los parámetros de configuración del sistema propuesto sólo son accesibles para los usuarios legítimos de la red y son proporcionados por una tercera parte de confianza autorizada durante la inicialización del protocolo.

- Otros tipo de Ataques. En MANETs otros de los ataques que se pueden llevar a cabo debido a la falta de una estructura centralizada son los ataques de denegación de servicio (DoS). El esquema propuesto no es demasiado propenso a este tipo de ataques ya que aunque las comunicaciones se realizan a través de un canal no seguro, sólo los nodos legítimos de las redes son capaces de enviar y descifrar los mensajes válidos, obviando desde el primer momento aquellos mensajes caducados o mal formados. Otro ataque peligroso en MANETs es el ataque sibling, que se produce cuando un nodo utiliza múltiples identidades de forma fraudulenta. Este problema se evita en el esquema propuesto gracias a la naturaleza distribuida de la NIZKP utilizada. Por último, la propuesta también es resistente a posibles robos de identidades ya que el acceso de los usuarios está controlado por la NIZKP. En definitiva, los ataques más habituales no tienen un efecto nocivo sobre la propuesta, debido a que su seguridad está basada en problemas matemáticos complejos y estándares actuales de funciones hash y de cifrado.

Todo esto conlleva concluir que el esquema propuesto supone una mejora significativa con respecto a las propuestas existentes, y además permite definir un nuevo sistema de autenticación basado en demostraciones de conocimiento nulo no interactivas, donde se condensan todos los retos y las respuestas en un único paquete, con una estructura innovadora para ser utilizada en entornos dinámicos de Internet de las Cosas como son las redes móviles descentralizadas y las redes vehiculares.

Capítulo 3

Revocación en VANETs

Junto a un buen método de autenticación de usuarios, una red vehicular necesita un método que permita de forma eficiente comprobar la honradez de los usuarios. Si un usuario empieza a generar eventos fraudulentos haciendo que la VANET se convierta en una red poco fiable, el resto de usuarios dejarán de creer en este tipo de soluciones. Ese usuario debería ser reconocido como usuario malintencionado y quedar almacenado en algún tipo de estructura que permita su consulta de forma rápida y veraz. Para ello están las denominadas Listas de Revocación de Certificados o CRL (Certificate Revocation List) que permiten almacenar los certificados de aquellos vehículos que han tenido un comportamiento malicioso o simplemente han caducado. Los trabajos de investigación actuales sugieren que las RSUs, que son las encargadas de la gestión de cualquier tipo de certificado, sean las responsables de monitorizar el mal comportamiento de los vehículos para proponer la revocación de sus certificados y generar las listas de revocación de certificados. Para ello deben tener comunicación directa con la Autoridad de Certificación o CA (Certificate Authority), que es la entidad de confianza responsable de emitir y revocar los certificados. Las RSUs son dispositivos electrónicos de pequeño y medio tamaño que pueden estar situados en cualquier punto de la carretera, generando una red de RSUs. De acuerdo con [224] y debido al estándar DSRC 5.9 GHz, debe haber una RSU cada kilómetro, por lo que en ciertas zonas aisladas se hará inviable poder contar con RSUs desde el primer momento, lo que conllevaría no poder comprobar la legitimidad de los vehículos de las zonas donde no haya conexión directa con las RSUs. Las listas de revocación de certificados clásicas pueden ser vistas como listas negras donde se almacenan de forma consecutiva todos los certificados no válidos en un determinado momento. Otro aspecto a destacar es que con el fin de proteger la privacidad de los usuarios de una VANET, una solución sería que cada vehículo utilizara múltiples seudónimos para evitar que otros usuarios fraudulentos puedan seguir su trayectoria [33]. Esto provocaría que si en una red vehicular se adoptase el esquema clásico de CRLs para resol-

ver el problema de la revocación de vehículos, las listas de revocación serían completamente inmanejables porque su tamaño crecería rápidamente debido al aumento de vehículos y el uso de múltiples seudónimos. Además, este problema puede verse magnificado por culpa del efecto denominado Implosion Request, que se refiere a la situación donde varios vehículos al mismo tiempo quieren sincronizar y actualizar su lista de revocación, produciendo congestiones en las comunicaciones y una sobrecarga en la red que conllevaría generar una latencia mayor en el proceso de validación de un certificado. En esa situación, la RSU sufriría una sobrecarga al asumir la responsabilidad de la revocación de usuarios de toda una VANET, ya que las CRLs necesitan ser retransmitidas cada 0,3 segundos [207], provocando que la comprobación de la honradez de un usuario sea por cada mensaje que recibe un vehículo. Además, en una red vehicular existen tipos de mensajes periódicos cuya frecuencia de envío es de apenas 0,1 segundos [207], haciendo inviable que las RSUs se encarguen del proceso de revocación usando la forma tradicional.

Este capítulo propone un nuevo sistema de gestión de certificados y/o seudónimos revocados usando árboles hash como estructura de datos autenticada. Los resultados expuestos acerca de esta línea de investigación han sido publicados en varios congresos indexados de alto impacto, entre los cuales se encuentra un CORE A* [164] y un CORE A [165].

3.1. Estado del Arte

Existen muchos investigadores trabajando en el problema de la revocación de certificados para entornos móviles como las redes vehiculares. Así, en [224] se proponen mecanismos de seguridad para lograr la revocación de certificados de forma segura, y para superar los problemas que causan las CRLs clásicas. El uso de la criptografía de clave pública es esencial para la seguridad de la información [21]. Por ello, otros trabajos sobre seguridad en VANETs [208] [201] proponen el uso de una infraestructura de clave pública o PKI (Public Key Infrastructure) y el uso de firmas digitales, pero no proporcionan ningún mecanismo de revocación de certificados, a pesar de que es un componente necesario en cualquier solución basada en PKIs. Esto se debe, en gran medida, a que el problema de la revocación es uno de los más difíciles de resolver. En [136] se proponen algunos protocolos de revocación de certificados usando la arquitectura PKI tradicional. Los procedimientos de revocación habituales se basan en una CA que administra los certificados de clave pública revocados añadiendo sus números de serie a una CRL y distribuyendo esta CRL a lo largo de la red con el fin de que los usuarios sepan qué nodos ya no son dignos de confianza [194]. En estas circunstancias, es muy importante que la distribución de la CRL se realice de manera eficiente con el fin de permitir que el conocimiento acerca de los nodos fraudulentos pueda propagarse rápidamente. La familia de estándares IEEE 1609 descri-

be el uso de PKI en VANETs. En particular, el trabajo [109] define una propuesta para el uso de una PKI para proteger los mensajes a través de autenticación mutua de las entidades en VANETs. Como continuación de ese trabajo, la propuesta [209] presenta un protocolo de seguridad basado en PKI donde cada vehículo carga con anterioridad sus claves pública/privada, y una Tercera Parte de Confianza o TTP (Third Trust Party) almacena todos los certificados anónimos de todos los vehículos. Por ello, dicho sistema no puede considerarse eficaz en el proceso de gestión de certificados. También existen soluciones basadas en PKI que utilizan la autenticación fuerte en VANETs para la firma de los mensajes [113]. Sin embargo, el uso de un enfoque tradicional basado en PKI puede no satisfacer los requisitos necesarios en las comunicaciones vehiculares de acuerdo a la necesidad de poder contar con comunicaciones en tiempo real a través del protocolo DSRC. Esto se debe a que cada OBU transmitirá periódicamente en intervalos muy cortos de tiempo mensajes de tipo beacons, lo que genera un tráfico masivo de mensajes que no pueden corroborarse usando PKI tradicionales.

En [210] se presenta el problema de la revocación de certificados y su importancia. Dicha investigación discute los métodos actuales de revocación y sus principales debilidades, y proponen nuevos protocolos para la revocación de certificados a través de un conjunto de diversos esquemas que incluyen: las listas CRL de revocación de certificados convencionales, las listas de revocación de certificados comprimidas o RC2RL (Revocation using Compressed Certificate Revocation Lists), la revocación de dispositivos a prueba de manipulación o RTPD (Revocation of the Tamper Proof Device) y el protocolo distribuido de revocación o DRP (Distributed Revocation Protocol). Además, en el estudio hacen una comparación de cada método, indicando las diferencias entre ellos. Los autores llegan a realizar una simulación del protocolo DRP llegando a la conclusión de que este protocolo es el más conveniente. La simulación se ejecuta en tres escenarios: en autopista, en ciudad y en un entorno mixto.

En [225] proponen dividir la red en pequeñas agrupaciones adyacentes para reemplazar las CRL clásicas por la CRLs locales que se intercambian de forma interactiva entre OBUs, RSUs y CAs. El tamaño de estas nuevas listas es más pequeño, ya que contiene los certificados de los vehículos de un sólo clúster o agrupación.

En [128] hacen una propuesta para distribuir de forma frecuente las CRLs directamente desde las CAs. Estas CRLs sólo contendrían los identificadores o IDs de los vehículos fraudulentos, para reducir su tamaño. La distribución de las CRLs por parte de las CAs hacia las RSUs y hacia todos los vehículos de una región lleva al problema de que no todos los vehículos recibirían la CRL, ya que existirían zonas rurales aisladas donde no llegarían. Para resolverlo, sugieren realizar comunicaciones entre las OBUs de forma cooperativa para enviar las CRLs y así disminuir el número de RSUs necesarias.

En [207] se introduce el concepto de evasión de vehículos problemáticos, además de introducir otros protocolos de revocación como la Revocación de Componentes de Confianza o RTC (Revocation of Trusted Component) y el Protocolo de Salida o Leave Protocol.

Otros trabajos anteriores asumen que la CRL puede ser enviada directamente desde las RSUs a las OBUs [118], y luego distribuirla entre las OBUs de forma cooperativa [188]. Sin embargo, como ya se ha mencionado, el gran tamaño que pueden adoptar las VANETs, y por consiguiente las CRLs, hace que este enfoque no sea factible debido a la sobrecarga que causaría a las comunicaciones de red. Este problema se incrementa aún más con el uso de múltiples seudónimos como sugieren en [200] para proteger la privacidad y el anonimato del vehículo. Si se usan seudónimos en VANETs, y puesto que existen casi mil millones de automóviles en el mundo [180], una conclusión directa es que el número de certificados revocados podría alcanzar pronto esa misma cantidad. Por otra parte, suponiendo que cada certificado ocupa por lo menos 224 bits, el tamaño total de la CRL sería de unos 224 Gbits, lo que quiere decir que su gestión, siguiendo el enfoque tradicional, no sería eficiente. Aunque se utilizasen CAs regionales, el tamaño que ocuparía una CRL seguiría siendo muy pesado, en torno a 1 Gbit. Esto implica que utilizando el protocolo 802.11a para la comunicación con las RSUs, la velocidad máxima de descarga de una OBU estaría entre los 6 y 54 Mbit/s en función de la velocidad del vehículo y la congestión de la red, por lo que, en promedio, una OBU necesitaría más de 30 segundos para descargar a través de una RSU, una CRL regional. Esto conllevaría no generar nuevas CRLs con la frecuencia adecuada, debido al tamaño que ocuparían, lo que afectaría a la frescura de los datos de revocación. Incluso el uso de técnicas conocidas para la realización de grandes transferencias de datos para la distribución de la CRL no sería una solución aceptable, ya que daría lugar a latencias más altas que afectarían a la validez de los datos. En consecuencia, sería muy útil contar con una solución que no requiera la distribución completa de la CRL desde la RSU a la OBU, como se propone en este capítulo de Tesis.

Existen otros métodos de revocación para VANETs que no requieren el uso de CRLs. El más conocido es el llamado OCSP (Online Certificate Status Protocol) [193], que implica la existencia de agentes de validación que respondan a las consultas de los vehículos con respuestas firmadas que indiquen el estado actual de un certificado en concreto. Sin embargo, este método de revocación divulga demasiada información de los vehículos. Además, como los agentes de validación deben ser globales, ha de existir un esquema de réplica lo suficientemente potente como para manejar la carga de todas las consultas de validación. Esto conlleva que la clave para firmar las respuestas debe ser replicada en todos estos servidores de réplica, convirtiendo este proceso en inseguro o muy costoso. Otra propuesta que no usa CRL, son los árboles de revocación de certificados o CRT (Certificate Revocation Tree),

propuestos por [127] como una mejora del protocolo OCSP consistente en una única entidad de alta seguridad que publica periódicamente una estructura de datos firmada a modo de CRL. Esta estructura de datos llega a todos los agentes de validación inseguros del sistema, para que los usuarios les realicen las consultas. En estos árboles, los nodos hoja representan a los certificados revocados y la CA firma el nodo raíz. Mediante el uso de este tipo de estructuras, el agente de validación inseguro puede demostrar el estado de cualquier certificado mediante la ruta desde el nodo raíz hasta el nodo hoja que representa al certificado revocado, gracias a la firma del nodo raíz por parte de la CA. Por lo tanto, no se hace necesario tener agentes de validación de confianza. La propuesta descrita en este capítulo se basa en la combinación de esta idea con los árboles hash, como estructura de datos autenticada o ADS (Authenticated Data Structure).

En general, un árbol hash es una estructura de árbol cuyos nodos contienen un resumen que puede utilizarse para verificar piezas más grandes de datos [183]. Los nodos hoja en un árbol de este tipo son los hashes de los datos que representan, mientras que los nodos internos del árbol son los hashes de sus respectivos hijos, siendo el nodo raíz del árbol el resumen de toda la estructura. Los árboles hash por lo general requieren el uso de una función hash criptográfica con el fin de evitar colisiones. La mayoría de las implementaciones de los árboles hash son binarias, pero la propuesta descrita en secciones posteriores de este capítulo propone el uso de estructuras más generales como son los árboles k-arios. Varios autores han propuesto ADS para gestionar los certificados revocados. En [127] la ADS propuesta es un árbol hash de Merkle [182] donde los nodos hojas representan certificados revocados según el número de serie. La manera de proceder de estas estructuras es mediante la consulta de un nodo acerca de otro nodo para comprobar si es un usuario revocado o no. El agente o entidad de validación más cercana le responde con una prueba concisa indicando si el certificado de dicho nodo está o no está en el CRT. El trabajo [117] presenta varios métodos para recorrer árboles Merkle permitiendo compensaciones espacio-temporales. Otras ADS se basan en estructuras de árbol multi-dimensionales como son las estudiadas en [184] para optimizar las consultas de búsqueda desde repositorios no confiables. Esto puede ocasionar que en cierto instante los árboles empiecen a degenerar y desbalancearse. El problema del balanceo de los árboles se solventa de forma sencilla mediante alguna de las muchas propuestas de algoritmos eficientes para balancear árboles hash descritas en [59]. Por ejemplo, los árboles AVL (Adelson-Velskii y Landis) se auto-balancean mediante rotaciones, los árboles B se equilibran manipulando los grados de los nodos y los árboles 2-3 restringen el número de hijos de un nodo a un mínimo de 2 y un máximo de 3, para asegurar el balanceo. Otro problema de las estructuras CRT aparece cuando se revoca un nuevo certificado, ya que todo el árbol debe ser recalculado y reestructurado. Para ello existen propuestas

basadas en listas de saltos [96] [97], que permiten tener una estructura natural y eficaz para reducir este problema equilibrando los CRT. Sin embargo, no son buenas soluciones para otros problemas como la inserción de nuevos nodos hoja en el árbol.

Tras analizar todas las propuestas existentes, se puede corroborar que la forma más habitual de gestionar la revocación de certificados en redes vehiculares es a través de CRLs o estructuras similares, utilizando servidores centrales para alojar a las Autoridades Certificadoras. Sin embargo, según nuestro conocimiento, ninguna de las propuestas existentes hasta ahora combinan una estructura de revocación segura que permita generar pruebas de revocación concisas junto a algoritmos óptimos de reestructuración de esas estructuras. Existen propuestas como [192] [88] que utilizan árboles hash autenticados para sustituir a las listas de revocación, pero el tipo de árbol utilizado requiere complejos y pesados algoritmos para realizar las operaciones sobre el árbol. En este capítulo se detallan nuevos métodos para gestionar certificados revocados en VANETs, combinando estructuras de datos autenticadas con algoritmos eficientes para operar sobre las estructuras [39].

3.2. Estructuras Basadas en Árboles

En esta sección se proponen diferentes métodos para gestionar las revocaciones de los nodos en las redes vehiculares. Para ello se detallan diferentes estructuras de datos autenticadas basadas en árboles, así como diferentes métodos para generar pruebas fehacientes de que un determinado vehículo está revocado.

3.2.1. Árboles k-arios

Uno de los mayores problemas de las estructuras en árbol utilizadas por la mayoría de propuestas actuales es la reestructuración del árbol. Por ello, se propone un nuevo tipo de estructura de datos autenticada basada en árboles k-arios (ver Figura 3.1). Los árboles k-arios son un tipo de árbol donde cada nodo tiene como máximo k hijos. El uso de árboles hash k-arios en lugar de otro tipo de árboles permite aumentar la eficiencia de la construcción y actualización de los árboles hash. En este caso, la reestructuración del árbol sólo se produciría cuando el árbol k-ario requiera un nuevo nivel de profundidad. De lo contrario, los nodos simplemente se insertan de izquierda a derecha hasta rellenar el espacio disponible en ese nivel. De esta forma, la propuesta aquí descrita se basa en una estructura dinámica de datos en forma de árbol que varía en función del número de revocaciones [156].

El modelo propuesto está diseñado utilizando la siguiente notación:

- h: Función hash criptográfica usada para generar los valores de cada nodo del árbol.

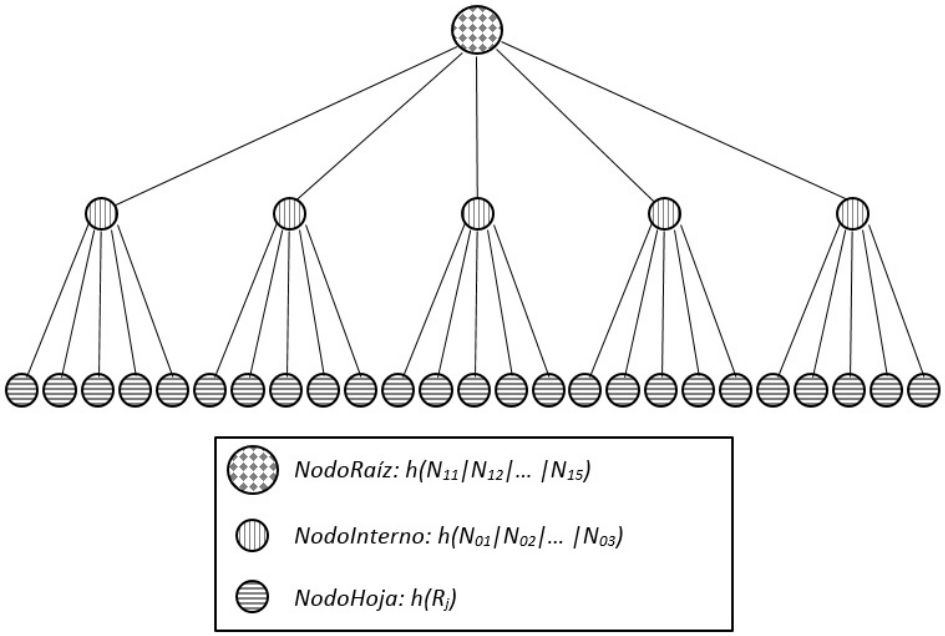

Figura 3.1: Árbol Hash Basado en un Árbol 5-ario

- D (≥ 1): Máxima profundidad posible del árbol hash.
- d ($< D$): Profundidad de un nodo interno del árbol hash.
- t: Número total de certificados o seudónimos revocados.
- R_s ($s = 1, 2, ..., t$): Número de serie del certificado o seudónimo revocado j-ésimo.
- N_{ij} ($i = D-d$ and $j = 1, 2...$): Nodo interno ij-ésimo del árbol obtenido tras hacer el hash de todos sus hijos.
- N_{0j} ($j = 1, 2...$): Nodo hoja del árbol que contiene el hash del certificado o seudónimo revocado ($h(R_s)$).
- k: Número máximo posible de nodos hijos de cada nodo interno del árbol hash.
- f: Función Keccak usada en la función hash SHA-3.
- n: Tamaño en bits del hash producido por h, que se asume en el ejemplo como el menor tamaño posible de SHA-3, 224 bits.
- b: Tamaño en bits de la entrada a la función f, que se asume en el ejemplo como uno de los posibles valores de Keccak, 800 bits.

- r: Tamaño en bits de la entrada con padding a la función h, que se asume en el ejemplo como 352 bits.
- c: Diferencia en bits entre b y r, que se asume en el ejemplo como el estándar SHA-3, $2n$, 448 bits.
- l: Tamaño en bits de los bloques de salida para la construcción de la función hash h, que se asume que no puede ser menor que r.

Por un lado, con el fin de construir el árbol, el primer parámetro a considerar es el número máximo de hijos por nodo. Este parámetro define el valor k del árbol k-ario que se construye. Si k es igual a 2, el árbol resultante es el típico árbol binario, pero la propuesta aquí descrita tiene el potencial de permitir diferentes valores para k, tales como 3, 4, 5, etc. Por ejemplo, se podrían crear árboles 5-arios, como el mostrado en la figura 3.1. Por otra parte, a fin de encontrar un nodo en el árbol con la mayor rapidez posible, la propuesta utiliza una tabla hash para asignar a cada certificado o seudónimo revocado la ruta exacta que define su situación en el árbol. En el esquema descrito, el costo del ancho de banda generado al enviar nuevas versiones del árbol de revocación desde la TTP hacia las OBUs se reduce en comparación con el resto de propuestas existentes, gracias a que sólo se envía las actualizaciones parciales de los nodos en el árbol que han cambiado. Esto implica una mejora significativa con respecto a los esquemas anteriores basados en árboles para gestionar certificados o seudónimos revocados, ya que estos esquemas requieren la actualización de todo el árbol cada vez que se añade o modifica un determinado nodo. La autenticidad de la estructura del árbol hash propuesta está garantizada gracias a la firma del nodo raíz por la TTP.

El procedimiento necesario para verificar que un determinado nodo está revocado es el siguiente. Si la entidad verificadora, en este caso una RSU, encuentra el nodo en cuestión dentro del árbol de revocación, entonces genera una respuesta de revocación que contiene la ruta desde la raíz del árbol hasta el nodo hoja correspondiente y los nodos hermanos de todos los nodos incluidos en esa ruta. Cuando la OBU recibe este paquete de verificación, puede reconstruir el nodo raíz a partir de esta información recibida, y corroborar que dicho nodo raíz coincide con el nodo raíz legítimo del árbol de revocación que debe almacenar como secreto de la red. De esa manera, la OBU puede cerciorarse de que la verificación de que un determinado nodo está revocado es verídica, gracias a la firma del nodo raíz por parte de la TTP.

En cuanto a la función hash criptográfica h utilizada en el árbol, se propone utilizar una función hash basada en el estándar SHA-3. En SHA-3, la función hash criptográfica f, denominado Keccak, contiene 24 iteraciones de transformaciones básicas. La entrada de cada ronda se representa como una matriz de 5×5 con elementos de 64 bits [155] [148]. Nuestra propuesta ha sido diseñada para utilizar elementos de 32 bits, en vez de 64 bits [153].

Otra variación propuesta es el uso de una versión dúplex de la construcción en esponja de SHA-3 [154]. Por un lado, al igual que la construcción esponja de SHA-3, nuestra propuesta también utiliza la función Keccak como una transformación de longitud fija f, así como la misma regla de relleno y la misma tasa de bits r. Por otra parte, a diferencia de una construcción esponja como la utilizada por el estándar SHA-3, nuestra propuesta de usar una construcción dúplex permite que la salida pueda ir concatenándose a medida que se vayan añadiendo nuevas entradas, sin tener que recalcular las salidas acumuladas (véase la Figura 3.2).

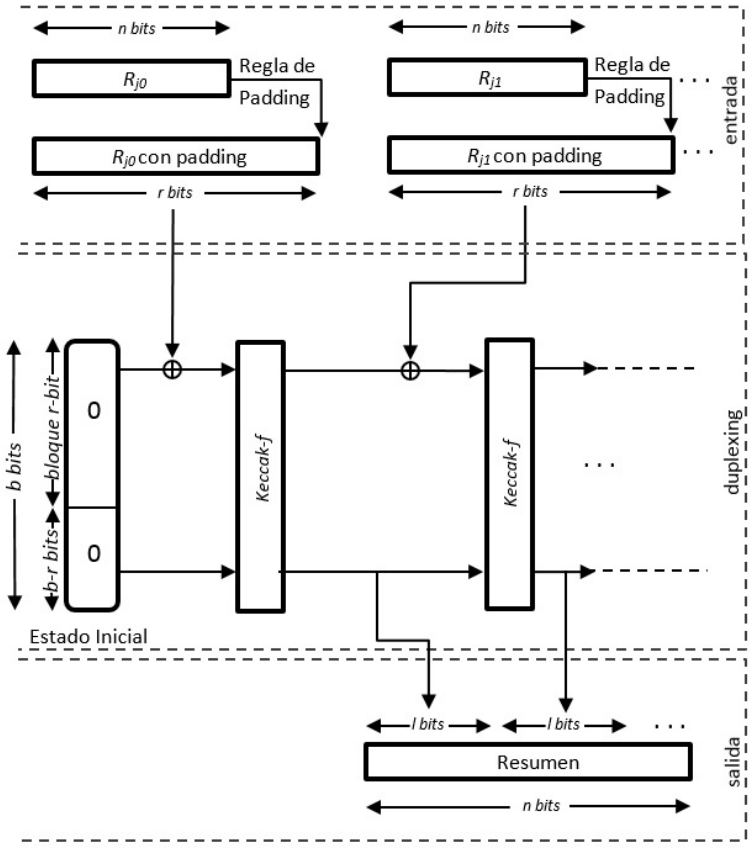

Figura 3.2: Construcción Dúplex Propuesta

Además, el uso de la construcción dúplex propuesta en el árbol hash permite la inserción de un nuevo certificado o seudónimo revocado como un nuevo nodo hoja del árbol mediante la ejecución de una sola iteración de la construcción dúplex. En particular, las RSUs pueden aprovechar todos los hashes correspondientes a los nodos hermanos del nuevo nodo, que se han calculado en iteraciones anteriores, para descartar los últimos bits de cada uno y hacer sitio al hash generado por el nuevo nodo de manera que el tamaño total del hash resultante para almacenar en el nodo padre sigue siendo el

mismo, n. Este procedimiento propuesto hace que la construcción del árbol hash sea más eficiente en el proceso de inserción, que todas las propuestas analizadas con anterioridad. En el caso de que no se haya alcanzado el número máximo de nodos hijos de un nodo interno, las RSUs tienen que almacenar tanto los hashes de los nodos, como el estado resultante de la última iteración de la función hash f para cada nodo interno. De esta manera, cuando un nodo interno añade un nuevo hijo, este estado se utiliza como entrada de la función f para generar el nuevo hash, utilizando únicamente una sola iteración del algoritmo de hashing.

La eliminación de un nodo del árbol debido a que la revocación ha caducado, requiere la reconstrucción de la parte del árbol dónde esté situado el nodo a borrar. Por lo tanto, con el fin de maximizar el proceso de borrado, esa reconstrucción necesaria del árbol sólo se realizará en el momento en el que todos los nodos hermanos también hayan expirado, de manera que se evita tener que reconstruir el árbol de manera frecuente. La TTP es la encargada de actualizar periódicamente el árbol, eliminando los nodos que representen a certificados o seudónimos expirados, y reconstruyendo el árbol cuando sea necesario. Después de cada actualización, la TTP envía las modificaciones del árbol a todas las RSUs. De manera que las RSUs se encargan de atender las solicitudes de las OBUs para notificarles si un determinado nodo está o no revocado. Las RSUs responderán estas solicitudes de comprobación de revocación de las OBUs, mediante la prueba verificable de que un determinado certificado o seudónimo está revocado o mediante un mensaje firmado que indica que dicho certificado o seudónimo que se ha consultado no está revocado. En el primer caso, mediante la prueba verificable, la OBU puede reconstruir el nodo raíz del árbol de revocación y comprobar que la firma de dicho nodo se corresponde con la firma de la TTP. En el segundo caso, cuando una OBU recibe un mensaje firmado por una RSU acerca de que un determinado certificado o seudónimo no está revocado, la confianza que establece es momentánea. Esto se debe a que cuando entra en contacto con otra RSU, le preguntará de nuevo sobre el mismo certificado o seudónimo que le habían dicho que no estaba revocado. Si la nueva RSU proporciona a la OBU una prueba de revocación cuya marca de tiempo contradice la respuesta anterior firmada por la otra RSU, la OBU envía a esta última RSU una notificación de que existe otra RSU que está comportándose de forma fraudulenta, por lo que la RSU honesta enviará una notificación a la TTP, que procederá a revocar la clave pública de la RSU deshonesta. De lo contrario, si la segunda RSU también envía un mensaje firmado corroborando la no revocación, la OBU continúa preguntando por el mismo certificado o seudónimo hasta que consigue la misma respuesta positiva de un determinado número de RSUs. Cada OBU puede almacenar localmente dos estructuras distintas y complementarias, una que contiene los seudónimos de las OBUs deshonestas que ha comprobado previamente, y otra con los seudónimos de

aquellas que han sido catalogadas como fiables hasta entonces. Por lo tanto, en el futuro, si se vuelve a conectar con cualquiera de estos vehículos, puede utilizar dicha información para decidir cómo proceder. Si no hay RSU cerca, puede utilizar estos datos para decidir si establecer la comunicación o no. Para un certificado o seudónimo revocado ya comprobado que esté en la primera estructura, incluso si hay una RSU cerca, no hay necesidad de volver a preguntar.

La elección de los valores adecuados para los distintos parámetros de la propuesta debe hacerse con cuidado, teniendo en cuenta las relaciones entre ellos. En particular, dado que el tamaño máximo del árbol está definido por:

$$n(1 + k + k^2 + k^3 + \cdots + k^D) = \frac{n(k^{(D+1)}-1)}{k-1}$$

Se puede deducir que el tamaño está acotado superiormente por la cantidad de memoria disponible en la RSU, y por el número máximo de nodos hoja del árbol k-ario k^D. Además está acotado inferiormente por el número total de certificados o seudónimos revocados t. Ambas condiciones se pueden utilizar para deducir el valor óptimo de k.

3.2.2. Árboles de Huffman

Otra alternativa que hemos desarrollado a partir de la propuesta anterior basada en el esquema de árbol k-ario añade un nuevo elemento al esquema, que es el uso de códigos de Huffman [41] [44]. En concreto, la nueva propuesta hace uso de la teoría de los códigos de Huffman para representar a los certificados o seudónimos revocados como nodos hoja a diferentes profundidades en el árbol, en función de la frecuencia con la que los vehículos transitan las carreteras. La teoría de los códigos de Huffman [110] define un algoritmo que se utiliza mayoritariamente para la compresión de datos. El término se refiere a la utilización de una tabla de códigos de longitud variable para la codificación de ciertos símbolos, donde la tabla se completa de una manera específica en base a la mejor probabilidad estimada de ocurrencia de cada valor posible del símbolo. Nuestra propuesta se basa en llevar esa idea a las estructuras autenticadas de datos para la gestión de certificados revocados, y asignar rutas más cortas en el árbol a los seudónimos revocados que son más consultados (ver Figura 3.3). Por lo tanto, el árbol se construye acorde a la frecuencia de consulta de cada vehículo en la VANET. De este modo, un vehículo que transite de forma frecuente, en caso de ser revocado, tendrá una prueba verificable de revocación mucho más eficiente.

La notación usada se basa en la mostrada en la sección anterior. En este caso, la estructura de datos autenticada contiene la misma función hash denotada por $h(...)$. Esta función hash se encarga de generar el valor de cada nodo N_{ij} interno como el hash de la concatenación de sus nodos hijos. La profundidad del árbol hash, que en este caso indica el número de los

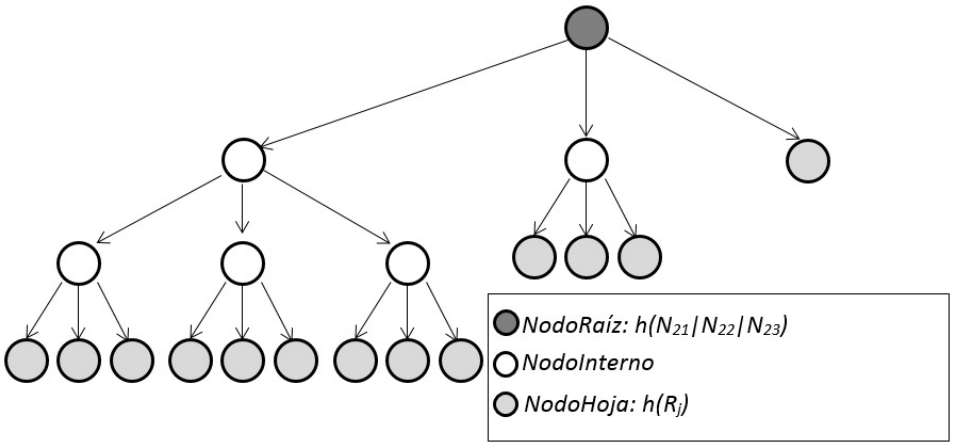

Figura 3.3: Ejemplo de Árbol de Huffman para Código Ternario

diferentes tipos de vehículos usados en el esquema, se designa como D, y su valor mínimo posible es 1. La profundidad de un nodo interno x del árbol se representa como d_x y su valor es menor que D. El número total de seudónimos o certificados revocados es denotado por t. Un seudónimo revocado se denota como R_j, donde $j = 1, 2, ..., t$. Cada nodo en el árbol hash se representa como N_{ij}, donde $i = D - d_{N_{ij}}$ y $j = 1, 2, ...$ El número máximo de hijos por cada nodo interno es k.

Además, esta propuesta se ha combinado con el uso de criptografía basada en identidad o IBC (Identity-Based Cryptography). La idea de la criptografía basada en identidad y, en particular, de la firma basada en identidad o IBS (Identity-Based Signature) es utilizar el identificador de la identidad pública del firmante como clave de verificación de una firma recibida. De esa manera, en el esquema propuesto, un nodo no necesita ningún certificado para demostrar la posesión de una clave pública. En lugar de eso, se utiliza un esquema de autenticación basado en identidad, junto a los árboles de Huffman k-arios para la revocación. Por ello, se considera la siguiente arquitectura de autenticación básica, que incluye tres partes principales:

- TTP: Esta entidad actúa como centro de distribución de claves, ya que es responsable de generar y asignar los parámetros necesarios a los nodos legítimos de la red vehicular. También se encarga de revocar los seudónimos de aquellos vehículos fraudulentos o malintencionados, así como las claves públicas de las RSUs comprometidas.

- RSU: Esta entidad otorga a la OBU toda la información solicitada sobre la consulta de seudónimos revocados.

- OBU: Cada vehículo está equipado con una OBU o equipamiento similar, que periódicamente emite mensajes firmados que son recibidos

por otras OBUs y RSUs cercanas.

En nuestro esquema, la identidad es un seudónimo P_j público que envía el nodo transmisor junto con el mensaje firmado. Por ello, cada nodo debe recibir todas las claves privadas P_rP_j vinculadas a todos sus seudónimos P_j emitidos por una TTP autorizada. En particular, una TTP, a la que se conoce en este esquema como generador de claves privadas o PKG (Private Key Generator), está a cargo de la generación y entrega de las claves privadas de cada nodo a través de un canal seguro y confidencial. Por otro lado, la PKG publica una clave pública principal o MP_u y conserva la correspondiente clave privada maestra o MP_r. Por lo tanto, teniendo en cuenta la clave MP_u, cualquiera de las partes podrá calcular la clave pública P_uP_j correspondiente para cualquier seudónimo P_j combinándola con la clave maestra MP_u. Para utilizar la clave privada correspondiente, el nodo autorizado de un seudónimo debe haber recibido sus claves de la PKG. Por lo tanto, se han diseñado e implementado los siguientes algoritmos:

- **Configuración**: El PKG escoge aleatoriamente su clave privada MP_r, con la que calcula su clave pública maestra MP_u para publicarla (ver Algoritmo 1).

Algorithm 1: Algoritmo de Configuración

```
1 MP_r ← Generar la clave aleatoria;
2 MP_u ← Generar la clave pública maestra a partir de MP_r;
3 Publicar MP_u;
```

- **Extracción**: Para cada seudónimo P_j, el PKG utiliza su clave privada maestra MP_u para calcular la clave privada P_rP_j. Todos los pares (P_j, P_rP_j) se envían de forma segura desde el PKG al propietario correspondiente (ver Algoritmo 2).

Algorithm 2: Algoritmo de Extracción

```
1 for j ← 1 to Total_Pseudonyms do
2     P_rP_j ← Generar claves privadas a partir de MP_r, P_j;
3     Enviar de forma segura (P_j, P_rP_j) desde el PKG al nodo
        correspondiente;
```

- **Firma**: Un nodo firmante utiliza su clave privada P_rP_j para calcular la firma de un mensaje M, y envía en abierto tanto P_rP_j (M) como su seudónimo P_j (ver Algoritmo 3).
- **Verificación**: Un nodo que recibe un mensaje firmado con un seudónimo (P_rP_j (M), P_j) utiliza la clave maestra MP_u y el seudónimo P_j para calcular P_uP_j y verificar la firma P_rP_j (M) (ver Algoritmo 4).

Algorithm 3: Algoritmo de Firma

1 $(P_j, P_rP_j) \leftarrow$ Seudónimo del nodo firmante y clave privada de dicho nodo;
2 $P_rP_j(M) \leftarrow$ Firma del Mensaje M;
3 Enviar $(P_rP_j(M), P_j)$;

Algorithm 4: Algoritmo de Verificación

1 Un nodo recibe $(P_rP_j(M), P_j)$;
2 $P_uP_j \leftarrow MP_u(P_j)$;
3 Verifica la firma $P_rP_j(M)$ mediante P_uP_j;

Esta propuesta no describe ningún nuevo método de cifrado basado en identidad, ya que no es su finalidad [160] [42]. El sistema basado en identidad que se ha implementado para las pruebas de concepto del prototipo diseñado, es el esquema de Boneh-Franklin [25], que utiliza un emparejamiento bilineal sobre curvas elípticas y su seguridad está basada en el problema bilineal de Diffie-Hellman [163].

El sistema basado en identidad está construido sobre un mapa bilineal $e : G1 \times G1 \rightarrow G2$ entre dos grupos $G1$ y $G2$ acorde a la bilinealidad de e : $e(aP, bQ) = e(P, Q)ab$ para todo $P, Q \in G1$ y $a, b \in Z$. Específicamente, un sistema basado en identidad puede construirse a partir de un mapa bilineal e si y sólo si existe una variante compleja del problema computacional de Diffie-Hellman en $G1$. Este problema bilineal de Diffie-Hellman en $G1$ se define como: Dado P, aP, bP, cP calcular $e(P, P)abc$, donde $P \in G1$ y $a, b, c \in Z$. En particular, el uso del emparejamiento bilineal e se describe para una curva elíptica E definida sobre algún campo K, por lo que se debe asignar un par de puntos de E a un elemento del grupo multiplicativo de una extensión finita de K.

La primera versión satisfactoria del esquema de Boneh-Franklin se basa en el emparejamiento de Weil [25]. Sin embargo, el esquema propuesto en esta Tesis utiliza el emparejamiento de Tate ya que se considera la función bilineal más conveniente para el esquema de Boneh-Franklin en términos de coste computacional. En particular, la propuesta implementada [43] incluye el uso del algoritmo de Miller para calcular el emparejamiento de Tate [184].

En IBC existen pocos trabajos que se apliquen a la temática de la revocación. Sin embargo, creemos que es necesario crear nuevas propuestas para proporcionar mecanismos eficientes que solucionen el problema de la revocación de certificados o seudónimos. Por ello, la propuesta descrita en esta sección combina los árboles k-arios de Huffman con la criptografía basada en identidad para gestionar seudónimos revocados en redes vehiculares.

En esta propuesta, la CA se encarga de construir un árbol preparado

para contener todos los nodos revocables. Tan pronto como un seudónimo de un vehículo es revocado, todos los seudónimos asociados a ese vehículo son insertados en esta estructura. Durante la inicialización del árbol, para estimar su tamaño, la CA utiliza datos reales otorgados por la autoridad competente propietaria de esos datos, con el fin de estimar el número de vehículos totales que existen de cada tipo. Con estos datos se estima el número de niveles, de acuerdo con la teoría de Huffman y el parámetro k que define al árbol. El número máximo de certificados revocados se ha estimado teniendo en cuenta los resultados mostrados en [249], donde se concluye que hasta el 1 % del número total de vehículos puede ser revocado. Por otra parte, el tamaño de certificado se supone que es 224 bits de acuerdo con el tamaño típico de los certificados de una CRL clásica en una VANET.

Cada nodo del árbol contiene un valor hash. Para cada certificado revocado R_s, la ruta se define como el camino desde el nodo raíz N_{D1}, al nodo hoja $N_{0j} = h(R_s)$ (ver Figura 3.4). El tamaño de esta ruta, que será la prueba verificable de revocación, depende del número de niveles que tenga el árbol. Cuando se consulta con frecuencia un determinado nodo revocado, su posición en el árbol estará en niveles superiores para tener un camino mucho más corto, y por lo tanto una prueba de verificación más eficiente.

Para encontrar un nodo en el árbol, se utiliza una tabla hash auxiliar que asigna a cada seudónimo revocado una ruta en el árbol. Por tanto, para un árbol de Huffman 3-ario como el representado en la Figura 3.4, y dado el nodo N_{02}, la ruta en el árbol viene definida por $[2, 2, 2, 1]$, lo que significa que desde el nodo raíz se debe ir escogiendo las ramas consecutivas 2, 2, 2, 1 para alcanzar el nodo hoja que representa al certificado o seudónimo revocado en cuestión.

Utilizando el método propuesto basado en los códigos de Huffman, se optimiza el tiempo de consulta y de cálculo de los seudónimos revocados más populares. En general, los vehículos que pasan más tiempo en las carreteras son los más propensos a comunicarse con otros vehículos, por mera estadística. Los sistemas clásicos de listas de revocación no toman en cuenta este factor, por lo que el coste medio por búsqueda de cualquier seudónimo revocado es el mismo. Sin embargo, el costo general puede ser optimizado asignando posiciones menos profundas en el árbol hash a los seudónimos más consultados, que se corresponden con los vehículos que permanecen más tiempo en la carretera.

La propuesta presentada en esta sección es óptima si las RSUs llevan la cuenta del número de consultas que se realizan para cada seudónimo revocado, y durante la actualización del árbol, los nodos se reorganizan en base a las nuevas frecuencias de consulta. Por otra parte, teniendo en cuenta el tipo de vehículo, podemos estimar que los vehículos públicos (autobuses, taxis, etc.) y los vehículos comerciales (camiones de reparto, vehículos turísticos, etc.) son más propensos a estar entre los más consultados, ya que pasan

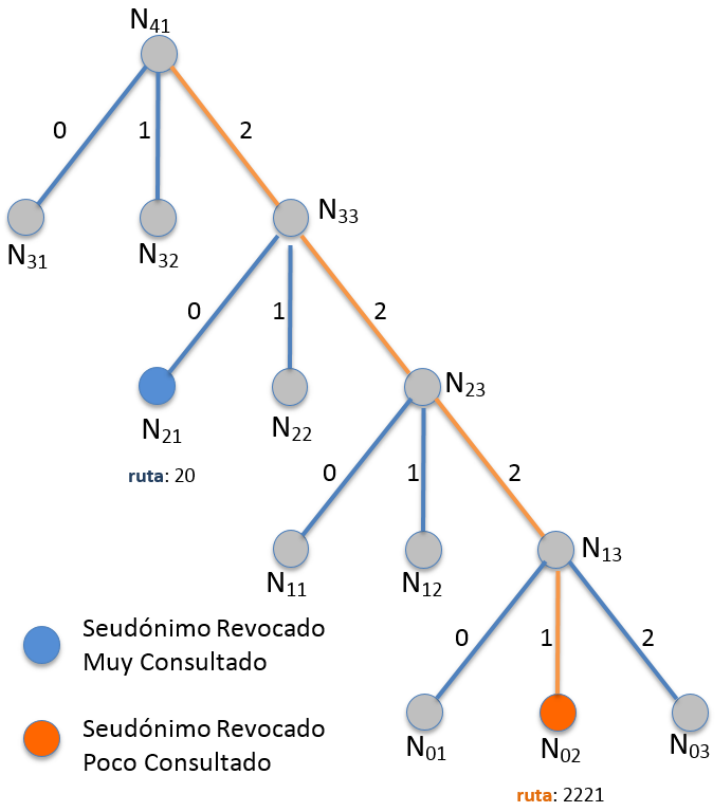

Figura 3.4: Niveles del Árbol según la Popularidad de Consulta

mucho más tiempo en las carreteras.

Además las RSUs son las encargadas de buscar los seudónimos de los vehículos en el árbol de revocación cada vez que un vehículo lo solicite. La RSU debe proporcionar al vehículo una prueba verificable de revocación si el seudónimo consultado está revocado, o un mensaje firmado indicando que el seudónimo solicitado no ha sido revocado. En ambos casos, el sistema de actuación es idéntico al propuesto en la sección anterior para árboles k-arios.

3.3. Implementación y Evaluación

Las propuestas anteriores han sido objeto de diferentes implementaciones y simulaciones con el fin de evaluar su desempeño. Las propuestas pueden ser consideradas computacionalmente eficientes, ya que evitan la necesidad de firmar cada respuesta, sea positiva o negativa, por parte de la RSU. En general, las propuestas no requieren confiar en todas las RSU.

Para lograr una evaluación realista, se han utilizado datos auténticos de entornos vehiculares concretos. Además, dependiendo del número de nodos

de un determinado entorno, el número de certificados revocados se ha estimado usando la propuesta descrita en [14]. El estudio realizado en [249] estima que el 1 % de los certificados podrían ser revocados. Los datos utilizados en las comparaciones han sido elegidos de acuerdo con el estudio presentado en [65]. Tal investigación se centra en la ciudad de Madrid cuya flota de vehículos se estima en 1,7 millones. Dados estos datos y el estudio del NIST sobre la tasa de revocación en VANETs, se ha simulado un escenario real con las características que se muestran en la Tabla 3.1.

Tabla 3.1: Parámetros del Escenario Simulado

Parámetro	**Valor**
Escenario	Madrid (España, 2014)
Tamaño	$25Km^2$
Tiempo de Simulación	1000 s
Número de Vehículos	1349
% de Vehículos con OBU	10, 20, ..., 90, 100
MAC	IEEE 802.11p
Modelo de Propagación	DSDV
Protocolo de Transporte	UDP
Tamaño de Paquete	1 Kb
Capa de Enlace	LL

Las características de los vehículos utilizados en la simulación están definidas en la Tabla 3.2.

Tabla 3.2: Características del Vehículo y la OBU

Parámetro	**Valor**
Velocidad	[0-33] m/s
Distancia de Transmisión	55 m
Antena	OmniAntenna
txPower	1.4 mW
rxPower	0.9 mW
sensePower	0.00000175 mW
idlePower	0 mW
Energía inicial	75 J

Las simulaciones se realizaron utilizando múltiples paquetes software con el fin de dar más credibilidad y presentar una simulación realista. Por lo tanto, el escenario que se ha utilizado para las simulaciones comprende una situación real del tráfico en la ciudad de Madrid (España) en el año 2014.

Para la generación de tráfico se ha utilizado la herramienta SUMO [236]. SUMO es un paquete de software de código abierto para la simulación de tráfico que permite el modelado de sistemas intermodales de tráfico, incluidos

los vehículos, el transporte público y los peatones sobre una área dada.

Con el fin de simular la arquitectura y las comunicaciones de una VANET, se ha utilizado una herramienta llamada NS-2 [197]. NS-2 es un simulador de eventos discretos destinado a la creación de redes que proporciona un apoyo sustancial para la simulación sobre TCP, enrutamiento y protocolos de multidifusión a través de redes ethernet e inalámbricas (locales y vía satélite). Los datos generados con NS-2 se transfirieron posteriormente a SUMO para su visualización en el mapa. El envío de mensajes y las interacciones entre los nodos en NS-2 se pueden ver en la Figura 3.5.

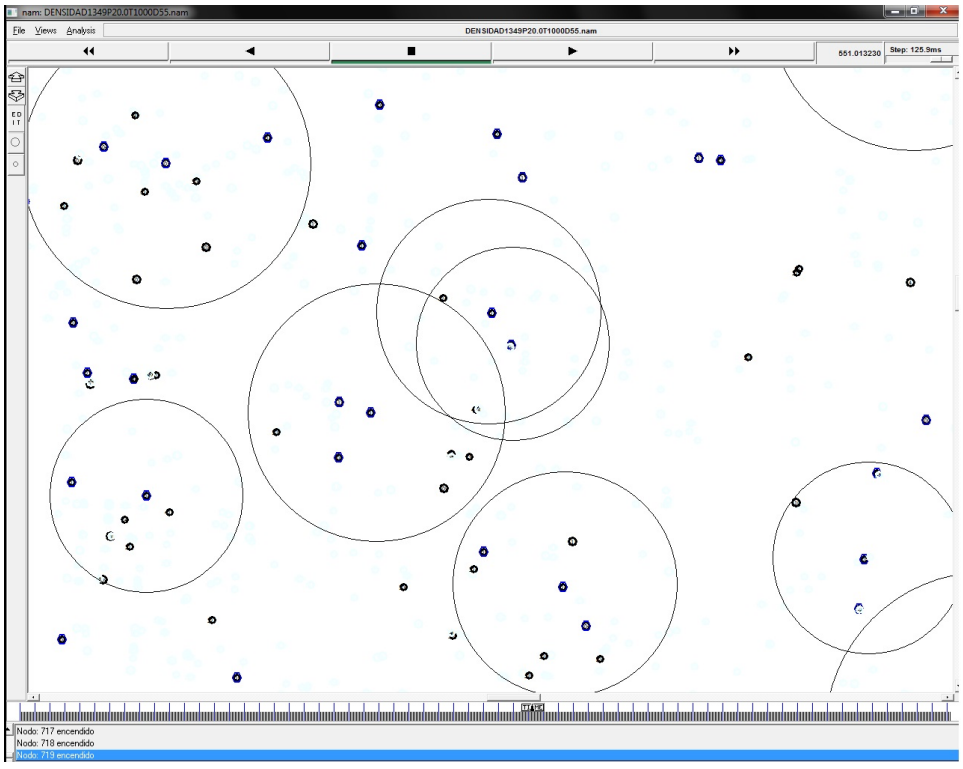

Figura 3.5: Simulación en NS-2

La interacción entre el tráfico generado con SUMO (Figura 3.6) y la simulación de la red con NS-2 se ha construido mediante el paquete de software MOVE.

MOVE permite generar rápidamente modelos de movilidad para las simulaciones sobre VANETs. MOVE está construido sobre el simulador SUMO. Su salida es un modelo de movilidad realista que puede ser usado de forma directa con simuladores de red como NS-2.

Con los datos y las configuraciones de la simulación creada, el aspecto visual del software de SUMO correspondiente al escenario seleccionado se muestra en la Figura 3.6.

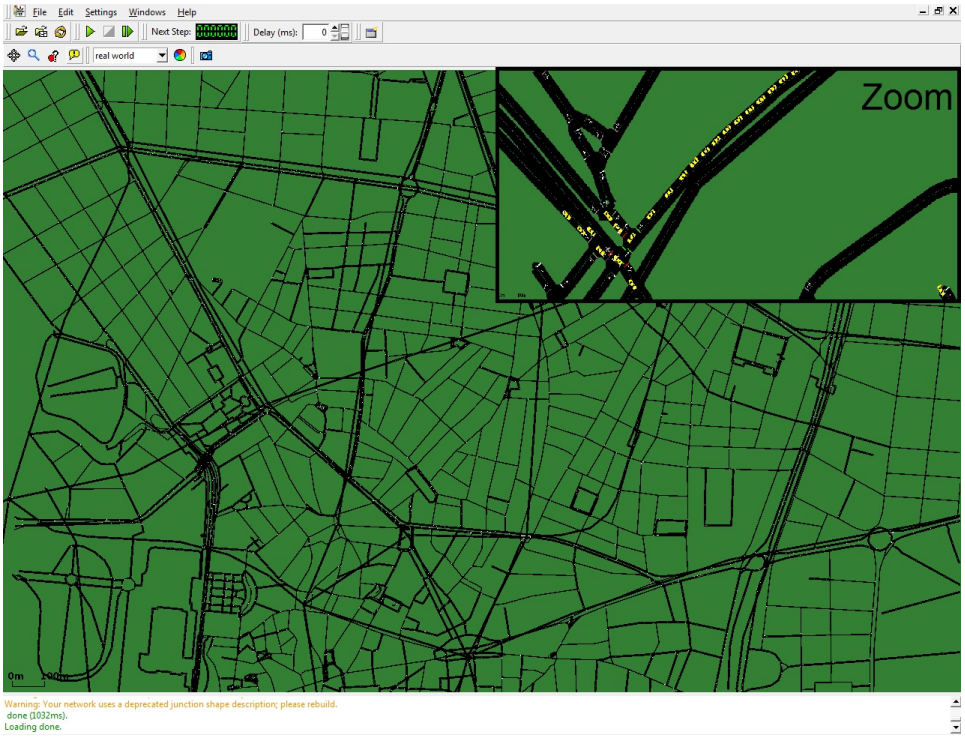

Figura 3.6: Simulación del Tráfico con SUMO

3.3.1. Árboles k-arios

Con el fin de optimizar las operaciones en el árbol, hemos desarrollado algoritmos específicos que trabajan en árboles hash k-arios. En primer lugar, se ha diseñado un algoritmo para encontrar un seudónimo revocado en el árbol (ver Algoritmo 5). La idea de este algoritmo es devolver la ruta completa desde el nodo raíz hasta el nodo buscado, incluyendo todos los nodos hermanos que se encuentran en la ruta. Este conjunto de nodos es la prueba de verificación de que un nodo se encuentra revocado. Así, un vehículo puede recalcular el nodo raíz para comprobar si la firma de la autoridad certificadora es válida.

Además, se ha definido un algoritmo específico para insertar un nuevo seudónimo revocado en el árbol. En este caso sólo se requiere una única iteración para calcular la función hash, al insertar el nuevo nodo (ver Algoritmo 6). Este algoritmo mejora la eficiencia de los algoritmos de inserción tradicionales que utilizan estructuras hash. Esto se debe a que se aprovecha y explotan las características de la construcción dúplex propuesta.

También se propone un algoritmo para el borrado de un seudónimo revocado del árbol (ver Algoritmo 7), donde los nodos hoja del árbol sólo se eliminan cuando expiran. De vez en cuando el sistema ejecuta un proceso

Algorithm 5: Algoritmo de Búsqueda

Input: *rpSearch*, Identificador a buscar
Output: *retPath*, Ruta desde el nodo raíz al nodo hoja encontrado

```
1 Añadir el Nodo Raíz a retPath;
2 for i ← (D − 1) to 0 do
3 |   Calcula la rama de rpSearch a la altura i;
4 |   Añade los nodos superiores de la rama calculada a retPath;
5 Calcula el nodo hoja rpSearch;
6 Añade el nodo hoja rpSearch y sus nodos hermanos a retPath;
7 return retPath;
```

Algorithm 6: Algoritmo de Inserción

Input: *rpNew*, Identificador a insertar

```
1 Ir al nivel más profundo donde se deba insertar rpNew;
2 if Si es Necesario Crear un Nuevo Nivel then
3 |   Generar un nuevo nivel;
4 |   Insertar el nodo rpNew;
5 |   Reconstruir el árbol;
6 else
7 |   Insertar el nodo rpNew;
```

que es responsable de la eliminación de certificados caducado en el árbol. Un certificado caducado es inválido para cualquier tipo de comunicación, por lo que los vehículos no preguntan sobre dichos certificados porque no confían de antemano.

Algorithm 7: Algoritmo de Borrado

Input: *rpDelete*, Identificador a eliminar

```
1 Definir nodeToDelete;
2 Asignar rpDelete a nodeToDelete;
3 while nodeToDelete no tenga Nodos Hijos do
4 |   Calcular el Nodo Padre de nodeToDelete;
5 |   Eliminar el Nodo nodeToDelete;
6 |   Asignar el Nodo Padre de nodeToDelete a nodeToDelete;
7 Reconstruir el Árbol;
```

Por último, se ha diseñado un algoritmo rápido y eficiente para la reestructuración del árbol (ver Algoritmo 8). Un árbol debe ser reestructurado sólo cuando la eliminación o inserción de un nodo provoca la eliminación o la creación de un nivel de profundidad.

Algorithm 8: Algoritmo de Reconstrucción

Input: *Tree*, Árbol Hash
Output: *Tree*, Árbol Hash Reestructurado

```
for i ← 1 to D do
    Calcular la rama i en Tree;
    Trasladar los nodos de la rama en Tree;
for i ← (D − 1) to 0 do
    for j ← 0 to Numero_de_Nodos_en_Nivel_i do
        Recalcular el Hash del Nodo j desde sus Hijos en el Nivel i+1;
return Tree;
```

De esa manera, una vez lanzada la simulación para el esquema propuesto basado en árboles k-arios, se han obtenido los siguientes resultados.

Los resultados obtenidos de comparar el tamaño de las listas de revocación clásicas y el tamaño del árbol de revocación diseñado, se pueden ver en la Figura 3.7.

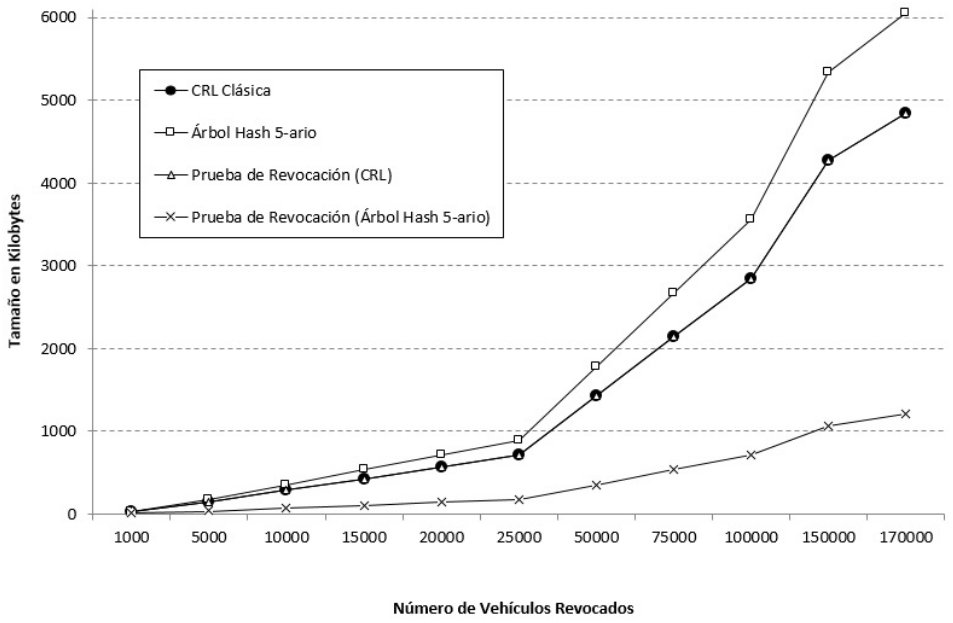

Figura 3.7: Comparativa entre CRL Clásicas y Nuestra Propuesta

Estos resultados demuestran que aunque la estructura de revocación propuesta utiliza más espacio que las listas de revocación tradicionales, la prueba de revocación propuesta requiere mucho menos espacio. Esto se debe a que las listas de revocación clásicas requieren que cada vehículo obtenga la lista

completa cada vez que requiera verificar el estado de un certificado o seudónimo, mientras que en nuestro esquema sólo se requiere conocer la prueba verificable sobre el árbol hash. Como se asume que las RSUs tienen suficiente memoria, la necesidad de almacenar el árbol hash de forma completa no produce ningún inconveniente. La cuestión clave es optimizar el envío de la prueba de revocación porque los vehículos se mueven a altas velocidades y las listas de revocación tradicionales son demasiado pesadas para ser enviadas en estos entornos.

Con respecto al proceso de verificación de revocaciones, el número de consultas realizadas en el escenario utilizando el árbol de revocación diseñado e implementado, se muestra en la Figura 3.8, dependiendo del porcentaje de vehículos que poseen una OBU.

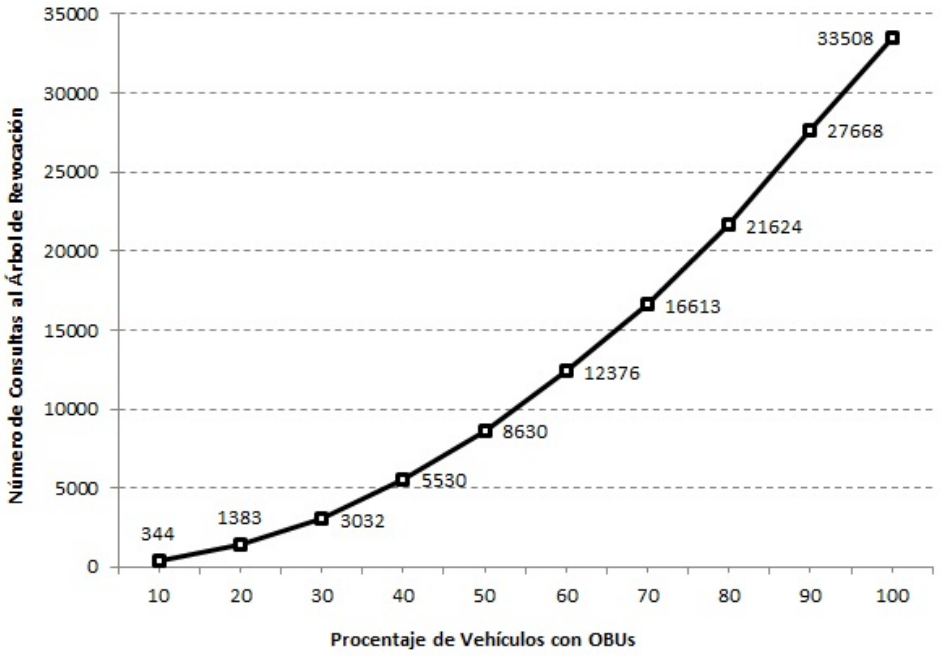

Figura 3.8: Consultas Realizadas en el Escenario Simulado

Las conexiones y autenticaciones generadas en la VANET se han estimado de forma aleatoria con el fin de comprobar la revocación de los nodos. Esta estimación se ha elaborado sobre el escenario de simulación y se ha comparado con el tráfico generado por las listas de revocación tradicionales en el mismo escenario. Los resultados de la comparación se pueden ver en la Figura 3.9, y demuestran la eficacia de nuestra propuesta.

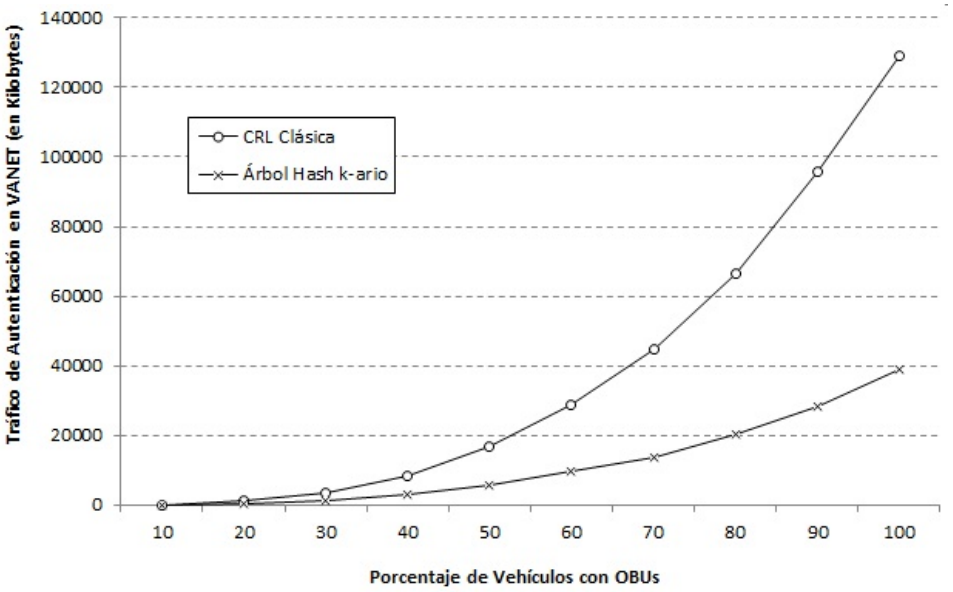

Figura 3.9: Tráfico Generado en el Proceso de Verificación

3.3.2. Árboles de Huffman

Para la simulación escogida, que se ha detallado en secciones anteriores, cabe destacar que de los 1,7 millones de vehículos de la ciudad de Madrid, alrededor de 80 por ciento son vehículos particulares. De esa manera, existen 15646 licencias de taxi y 2022 autobuses de la compañía de transporte público de Madrid. Los vehículos de reparto representan el 14 por ciento de los desplazamientos totales [65].

Después de analizar estos datos, se ha dividido a la flota de vehículos en 4 tipos diferentes: Los de Tipo A representan a los autobuses públicos, Los de Tipo B representan a los taxis y vehículos de emergencia, los de Tipo C representan a los autobuses privados y resto de vehículo de reparto, y los de Tipo D son los vehículos particulares y las motocicletas.

Sabiendo que el número total de nodos que pueden llegar a ser revocados ronda los 17000, se puede estimar que, como máximo, la estructura de revocación contendrá el siguiente número de nodos por tipo:

- Número Máximo de Nodos Revocados Estimados de Tipo A: 20.
- Número Máximo de Nodos Revocados Estimados de Tipo B: 390.
- Número Máximo de Nodos Revocados Estimados de Tipo C: 2990.
- Número Máximo de Nodos Revocados Estimados de Tipo D: 13600.

Con estos datos, se ha seleccionado un valor inicial k=21, que ha servido para ejecutar un procedimiento de construcción iterativo del árbol desde

los niveles inferiores a los niveles superiores. Este proceso ha servido para estimar el rango de valores óptimos de k, que se ha situado entre $(20, 49]$, siendo el valor óptimo resultante el punto intermedio k=35. De esa manera, el árbol de revocación propuesto para la simulación realizada contiene 35 nodos en el primer nivel, 525 en el segundo, 4725 en el tercer nivel, y 60725 potenciales nodos revocados en el cuarto nivel.

El tamaño de la prueba verificable de revocación en esta simulación realista, considerando 224 los bits de cada certificado, aumenta en función de la profundidad del árbol donde se ubique $(224 \cdot 35 \cdot d_x)$, de modo que el tamaño máximo de una prueba verificable están en torno a los siguientes valores:

- Tamaño máximo de la prueba verificable para los nodos de Tipo A: 8 Kilobits.
- Tamaño máximo de la prueba verificable para los nodos de Tipo B: 16 Kilobits.
- Tamaño máximo de la prueba verificable para los nodos de Tipo C: 32 Kilobits.
- Tamaño máximo de la prueba verificable para los nodos de Tipo D: 64 Kilobits.

Al utilizar CRL clásicas, la prueba que indica que un determinado nodo está revocado implica el envío de toda la CRL, por lo que en esta simulación significaría un total de $17000 * 228 = 3876000$ bits (3876 Kilobits) para cualquier caso. Sin embargo, usando el esquema propuesto, se conseguiría una mejora considerable en todos los casos (véase la Figura 3.10). Para las comparaciones mostradas en la Figura 3.10, se incluye la propuesta explicada en la sección anterior de usar un árbol 5-ario perfecto como estructura de revocación.

Como puede verse en la Figura 3.10, el sistema propuesto mejora en todos los casos, el tamaño total de peticiones que se hacen a la estructura de revocación. Esto se debe a la utilización de diferentes tamaños en función del tipo de vehículo que es revocado, lo que implica generar un tamaño más corto para los vehículos más comunes en las carreteras (ver Figura 3.11).

Por último, y para finalizar las evaluaciones, como se deduce de la Figura 3.11, en las simulaciones, también se optimizan las consultas más frecuentes, que son aquellas relacionadas con los vehículos que transitan con mayor asiduidad las carreteras.

Todo esto lleva a la conclusión de que las propuestas descritas en este capítulo mejoran considerablemente a las soluciones clásicas de revocación que son aún hoy en día las especificadas en los estándares de las VANETs. Nuestros esquemas permiten mejorar sustancialmente el tamaño de las pruebas

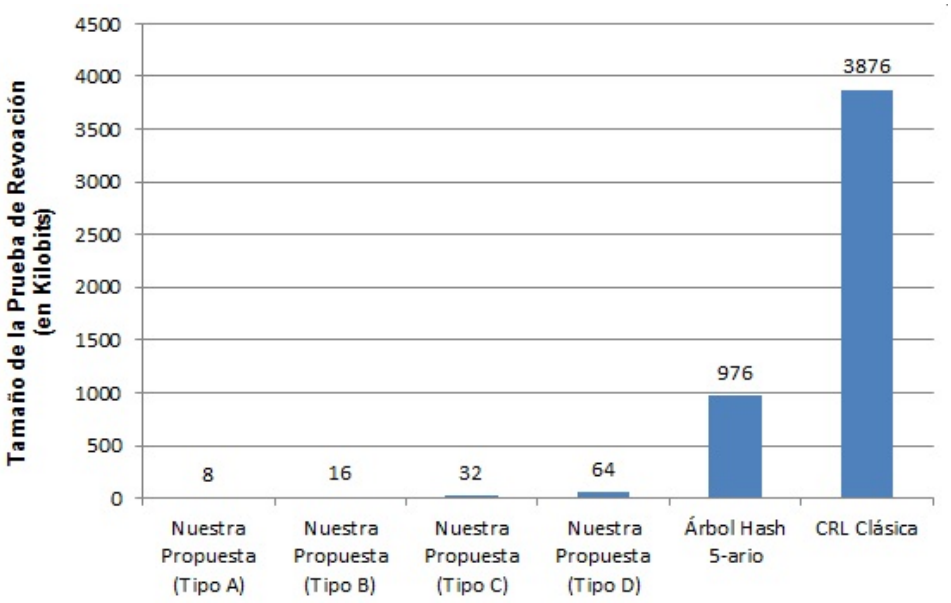

Figura 3.10: Comparativa entre los Tamaños de las Pruebas de Revocación

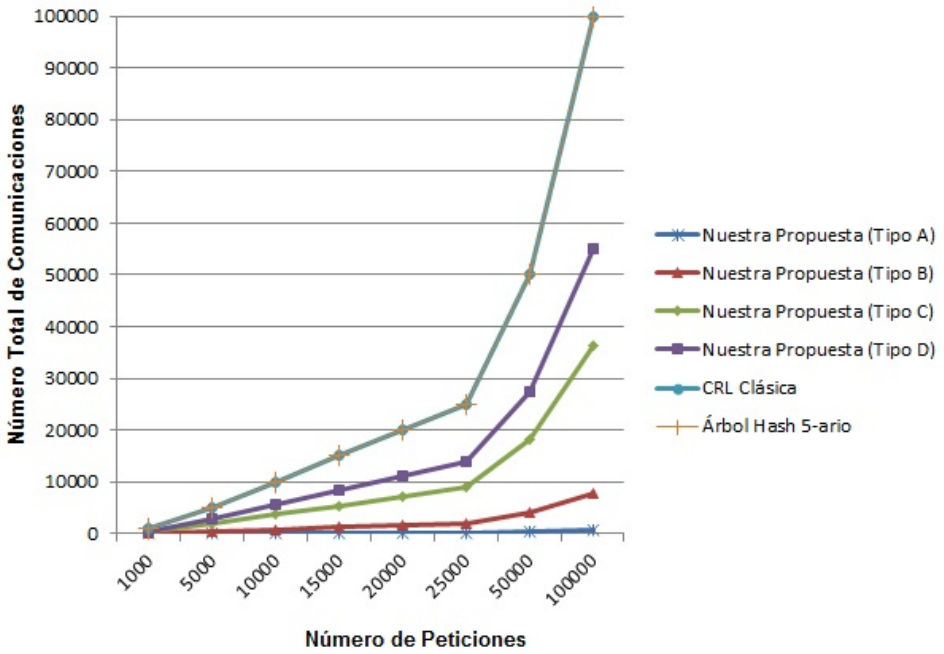

Figura 3.11: Número de Peticiones por Tipo de Vehículo

de revocación, consiguiendo que la velocidad de consulta se reduzca considerablemente. Este hecho es muy importante en una red de este tipo, debido a que se necesitan comunicaciones ágiles y rápidas que permitan a los vehículos

enviar y corroborar información a altas velocidades.

Capítulo 4

Aplicaciones Móviles

Este capítulo se centra en explicar diversas aplicaciones móviles realizadas durante los últimos 4 años para poner en práctica trabajos desarrollados para resolver distintos problemas. Durante el transcurso de la presente Tesis siempre se ha intentado llevar las investigaciones de la teoría a la práctica. Por ello, desde el primer momento, se han implementado todos los algoritmos propuestos para distintas situaciones, en diferentes aplicaciones móviles, o apps, que sirvan para demostrar la utilidad de las propuestas realizadas. En los tiempos que corren, la tecnología más portable por antonomasia, que ha llegado a un mayor nicho de la población mundial, son los teléfonos móviles. Además, su evolución tecnológica ha sido frenética, conjugando la experiencia de usuario con la capacidad de cómputo. Todo esto ha permitido que las investigaciones que se han realizado se hayan hecho accesibles al público general mediante la elaboración de apps de uso cotidiano, mayoritariamente centradas en entornos vehiculares. De cara a los usuarios, los algoritmos implementados son totalmente transparentes, por lo que se ha conseguido conjugar aplicaciones seguras con funcionalidades innovadoras al alcance de cualquier persona. Algunas de estas aplicaciones han sido premiadas en varios concursos tanto nacionales como internacionales. Además, se han presentado en diferentes congresos internacionales indexadas en la base CORE [166] [159].

4.1. Carpooling

El rápido incremento del número de vehículos a nivel mundial ha provocado una mayor contaminación en el aire, que es uno de los principales factores del calentamiento global. Por ello, los gobiernos de todo el mundo están tomando diferentes medidas para intentar disminuir la contaminación, sobre todo en las grandes ciudades. Una de esas medidas se basa en fomentar el uso del transporte público. Sin embargo, el transporte público no siempre es una solución viable para muchos de los ciudadanos que transitan asidua-

mente las grandes urbes. Otra solución práctica al problema se basa en evitar la baja ocupación de la mayoría de los vehículos en las ciudades, a través de fomentar el uso compartido de los coches. De esta manera se optimiza el uso de los asientos cuando los trayectos se realizan con los vehículos al completo. Esta modalidad es conocida por el término de Carpooling y es una de las formas más efectivas para reducir la contaminación y el gasto diario generado por el mantenimiento de los vehículos. Este tipo de soluciones colaborativas ha venido incrementando su tasa de éxito desde el inicio de la crisis económica, gracias a la tecnologías 2.0. Además, estas soluciones pueden ser aplicadas a casi cualquier escenario, siendo especialmente útiles en entornos universitarios, vacacionales, de viajes largos y en los desplazamientos dentro de los grandes núcleos urbanos. Su éxito radica en que satisface las necesidades tanto de los usuarios que ofrecen su coche para compartirlo, como de los que se benefician del uso compartido de vehículos ajenos. Por lo general, su objetivo principal es compartir el costo del combustible, pero puede haber otras razones motivantes como problemas de aparcamiento, interacción social o la protección del medio ambiente. Por otro lado, el principal problema de estas alternativas es la confianza, ya que el servicio requiere que los usuarios confíen entre sí y que sus comportamientos sean adecuados. Por eso, en esta sección se propone una solución innovadora para mejorar y aumentar el nivel de confianza de las soluciones existentes de carpooling, a través de técnicas avanzadas basadas en las redes sociales. La propuesta realizada [171] permite generar una cadena de confianza mediante algoritmos dinámicos de medida de la reputación de los usuarios, así como proteger su privacidad.

4.1.1. Estado del Arte

Los primeros proyectos de carpooling surgieron a finales de 1980 [240]. Sin embargo, por aquella época, los usuarios tenían que pertenecer a determinadas redes y conocerse entre ellos de primera mano, para comunicarse y establecer la relación oportuna. Poco a poco, los medios utilizados para organizar los viajes fueron cambiando gracias primero al teléfono por cable, luego a la aparición de Internet y el correo electrónico, y finalmente a los teléfonos inteligentes. De esa manera, el sistema empezó a crecer tan rápido que en el año 2009, el carpooling representó el 43,5 % [78] de todos los viajes realizados en Estados Unidos, donde el 60 % de estos viajes fueron realizados entre familiares. En Europa, el carpooling se ha ido haciendo cada vez más popular, gracias a plataformas y aplicaciones como la francesa BlaBlaCar o la alemana Carpooling. Hoy en día, existen diferentes plataformas y servicios de carpooling, pero aún no se ha conseguido establecer una masa crítica de usuarios suficiente como para que sea algo estandarizado entre la población. La Tabla 4.1 muestra varias de las características de los principales sistemas existentes, haciendo especial énfasis en la seguridad de cada uno de ellos. En particular, se han elegido los más populares: Amovens [233], Blablacar [20],

CarPooling [91], Compartir.org [54], y ZimRide [257].

Plataforma	Red Social	Privacidad	Sistema de Reputación	Usuarios Certificados	Cálculo de Confianza
Amovens [233]	si	si	si	no	no
Blablacar [20]	si	si	si	si	no
Carpooling [91]	si	no	si	no	no
Compartir [54]	no	si	no	no	no
ZimRide [257]	si	si	no	no	no
Nuestra Propuesta	**si**	**si**	**si**	**no**	**si**

Tabla 4.1: Plataformas de Carpooling

BlaBlaCar [20] es el principal servicio de transporte compartido del mundo. Se centra en los viajes de larga distancia, y utiliza las redes sociales para el registro, evaluación y comunicaciones de los usuarios. Por otro lado, el mayor servicio de coche compartido en los Estados Unidos es Zimride [257], en donde se realizan los pagos a través de tarjetas de crédito o cuentas de PayPal. El motor de confianza de todos estos sistemas se basa solo en las valoraciones otorgadas directamente por los usuarios. Sin embargo, un sistema basado en este tipo de confianza es bastante débil, debido a la posibilidad de crear nuevas cuentas o a la generación de trayectos falsos para aumentar el ratio de valoraciones positivas de determinados usuarios.

Aparte de estas plataformas comerciales, existen distintos trabajos de investigación que proponen diferentes alternativas basadas en el carpooling. El trabajo [47] muestra un sistema integral para la organización de trayectos compartidos mediante el uso de diferentes tecnologías web, sistemas de información geográficos y el tradicional SMS. Los autores de [217] proponen una innovadora plataforma web para compartir vehículo. En [81] se presenta una arquitectura de carpooling que utiliza un mecanismo de créditos para fomentar la cooperación entre los usuarios. Más recientemente, el trabajo de investigación [52] propone un algoritmo para fomentar el uso compartido de los vehículos basado en lógica difusa para generar prioridades entre usuarios a través de valoraciones. En [86] se define un algoritmo innovador para promover el carpooling a través del denominado instant processing. Por último, una de las propuestas más interesantes [26], utiliza una plataforma segura multi-agente, centrada en algoritmos seguros que permiten la autenticación mutua tanto de los usuarios, como de otros componentes externos. Nuestra propuesta se diferencia de todas las soluciones anteriores, ya que utiliza un algoritmo propio que facilita la decisión de los usuarios para confiar en personas desconocidas.

4.1.2. Plataforma

El objetivo principal del esquema propuesto es conseguir un incremento de usuarios en las plataformas de carpooling a través de la mejora de la seguridad y confianza de los usuarios. Una de las principales características de la propuesta es el hecho de que los usuarios que ofrecen los trayectos con

sus vehículos tienen su privacidad totalmente protegida [45]. A diferencia de otras plataformas similares, el sistema descrito no permite a ningún usuario acceder a datos confidenciales de otros usuarios, como el correo electrónico, el número de teléfono o el propio nombre completo, a menos que se haya autenticado en la plataforma y el algoritmo inteligente de confianza mutua lo estime conveniente. En este caso, un usuario con rol de acompañante puede ver todos los datos que un usuario con rol de conductor considere oportuno ofrecer. De lo contrario, sólo puede enviar una solicitud para que el propio conductor pueda decidir si el solicitante es de fiar o no. El algoritmo se basa en las relaciones de confianza entre los usuarios de la plataforma. Para recabar los datos que nutren al algoritmo, estos usuarios deben autenticarse en la plataforma a través de las principales redes sociales (Facebook, Twitter o Google+) [152]. De esta manera, el algoritmo calcula una cadena de confianza entre el conductor y el solicitante, en base a la denominada regla de los seis grados de separación [250]. La Teoría de los Seis Grados de Separación afirma que cualquier persona del planeta está conectada con cualquier otra, a través de una cadena de conocidos con no más de cinco eslabones o puntos de unión. Según esta teoría sólo seis niveles como máximo nos separan de cualquier persona del planeta. Además, el algoritmo considera otra serie de parámetros variables, como la reputación obtenida a través del uso de la aplicación. Para ello, al final de cada viaje compartido, la aplicación requiere a los usuarios que valoren al resto de usuarios con los que ha compartido trayecto. Las puntuaciones sirven al algoritmo para detectar los comportamientos correctos o incorrectos de los usuarios.

La arquitectura del sistema propuesto utiliza como modelo el conocido paradigma de cliente-servidor (Ver Figura 4.3). Los dos elementos de este modelo se definen a continuación:

- **Cliente**: Los dispositivos móviles usados por los usuarios.
- **Servidor**: El núcleo central de la plataforma, alojado en la nube. El servidor a su vez está dividido en dos partes:
  - Servidor de mensajería GCM (Google Cloud Messaging): encargado de manejar todas las notificaciones de tipo push que genera la plataforma para avisar a los usuarios de todas las comunicaciones en tiempo real.
  - Base de Datos del servidor dedicado: gestiona todos los datos de los usuarios y genera la comunicación entre los dispositivos móviles y el servidor de mensajería.

El esquema propuesto protege la privacidad del usuario a través del acceso limitado y controlado a los datos de los usuarios, de acuerdo con el nivel de confianza declarado por la relación entre cada par de usuarios. Este nivel de

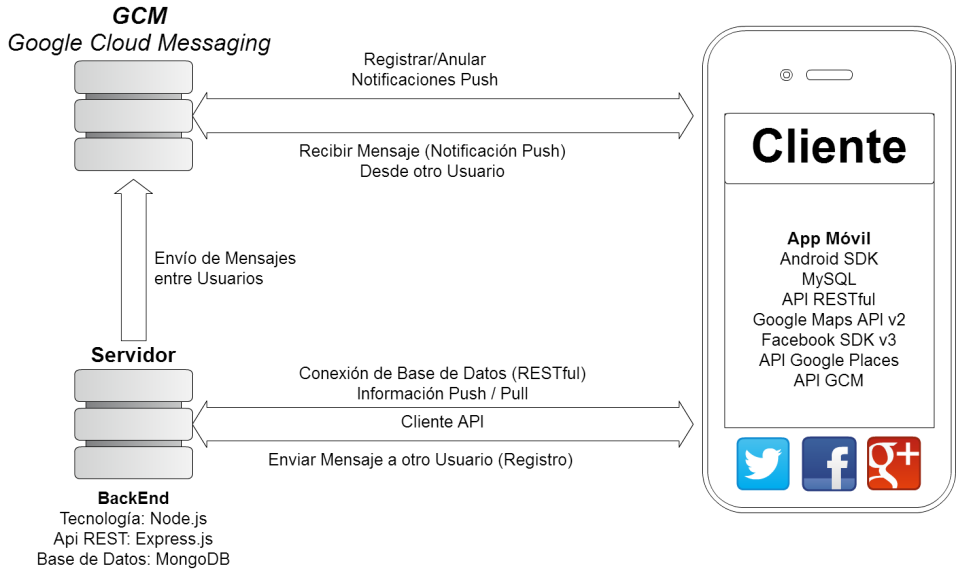

Figura 4.1: Arquitectura del Sistema Propuesta

confianza se consigue a través de la combinación de las valoraciones en el sistema y de los grados de amistad de separación entre los usuarios. De este modo, se proporciona una puntuación objetiva que sirve para que los usuarios puedan discernir de qué otros usuarios se pueden fiar. De esta manera, la privacidad es tratada como uno de los aspectos más importantes del sistema de carpooling. Una primera aproximación para desarrollar el algoritmo que mide la confianza de cada par de usuarios, se basa en el uso del algoritmo PageRank para predecir si dos personas pueden confiar entre sí. Sin embargo, dado que este algoritmo no se ajusta bien a la morfología del problema en cuestión, se ha utilizado un segundo enfoque que lo complementa, basado en redes bayesianas y que sirve para conocer si una persona puede confiar en otra. El algoritmo resultante ha sido utilizado para realizar la aplicación de carpooling que se describe en esta sección.

Ningún sistema de carpooling existente ofrece al usuario un método cuantitativo que pueda utilizarse para decidir si otro usuario es de confianza como para compartir con él el vehículo o no. La propuesta aquí descrita ofrece esta característica basada en la teoría de los seis grados de separación. Además, algunas de las propuestas existentes ni siquiera permiten a los usuarios decidir quién puede o no participar en un trayecto. Por otro lado, existen algunas propuestas que usan un método cuantitativo basado en las similitudes entre los usuarios [50] [205], para ponerlos en contacto. El principal problema de estas propuestas, es que necesitan recopilar información precisa sobre ciertos atributos y características de los usuarios, lo que requiere una interacción de estos con la plataforma. Nuestra propuesta pretende ser sencilla para

los usuarios, por lo que no necesitan rellenar ningún campo de información acerca de sus gustos o comportamientos. Con un simple click que habilita el inicio de sesión en las redes sociales, el usuario puede entrar en el sistema y comenzar a usar la plataforma. La recolección de la información es realizada de forma automática por el sistema, nutriéndose del contenido generado por los propios usuarios en las redes sociales.

El punto fuerte de la solución propuesta, es el algoritmo de reputación. Como se ha descrito antes, el algoritmo se basa en la teoría de los seis grados de separación y en las puntuaciones individuales entre los usuarios, dentro de la plataforma. Se ha demostrado que este número de grados de separación puede ser reducido significativamente mediante la introducción del concepto de redes sociales. La aplicación utiliza las redes sociales para crear los perfiles de los usuarios y para que estos puedan iniciar sesión de forma fácil. La información de las redes sociales se utiliza para interconectar a los usuarios y proporcionarles una medida fiable de confianza. A través del uso de las redes sociales se puede asegurar que la teoría de los seis grados de separación es reducida a tan sólo cuatro. En particular, de acuerdo con varios estudios de investigación sobre la red social más conocida, Facebook [63] [248] [11], los grados de separación de media obtenidos fueron de tan sólo 3.9, lo que demuestra que el mundo es aún más pequeño de lo esperado.

La medida final de reputación en el modelo empleado es un valor decimal entre -0.25 y 1, pero se muestra al usuario como un carácter (S, A, B, C, D). Dicha cantidad se calcula a partir de valores dinámicos por cada par de usuarios. Esta medida tiene en cuenta los dos parámetros mencionados con anterioridad: el grado de amistad de los usuarios dado por las redes sociales, y su valoración obtenida directamente en la plataforma.

El sistema representa la red social como un grafo ponderado dirigido definido por $G = (V, E)$ donde $|V| = n$, $V = \{v_1, v_2, ..., v_n\}$, $|E| = m$, $e_{ij} \in E$, $e_{ij} = v_i \rightarrow v_j$ si existe una conexión de v_i a v_j para todo $1 \leq i \leq n$ y $1 \leq j \leq n$, $v_i \in V$, $v_j \in V$. Por una parte, $A = [a_{ij}]_{n \times n}$ se refiere a la matriz de adyacencia donde $a_{ij} = 1$ si $e_{ij} \in E$, y $a_{ij} = 0$ en otro caso. Por otra parte, $P = [p_{ij}]_{n \times n}$ se refiere a la matriz que representa los grados de amistad, que son valores $\in [0, 1]$ definidos por la ecuación 4.1.

$$p_{ij} = \begin{cases} \frac{\sum_k (nA_{ij}^k \cdot wA_{ij}^k)}{MFC}, & sie_{ij} \in E \\ \\ 0, & enotrocaso \end{cases} \tag{4.1}$$

donde:

- nA_{ij}^k es el número de repeticiones de una acción k de Amistad de v_i a v_j. Está acción puede ser un comentario, un 'me gusta', o cualquier otra acción entre dos usuarios en una red social.
- wA_{ij}^k es el peso asignado por el sistema a cada tipo de acción k de

Amistad del usuario v_i al usuario v_j. Representa la importancia de un determinado tipo de acción. Su valor está comprendido entre 0 y 1. Por ejemplo, con datos de Facebook, el peso de un comentario es 0,727, y el peso de un 'me gustaés de 0,273. Estos valores se obtienen a partir del estudio psicológico realizado en [15], donde se concluye que por cada comentario en Facebook existen 2.667 "me gusta". Por lo tanto, un comentario es más importante que un "me gusta". El sistema es capaz de detectar los comentarios positivos, negativos y neutros, utilizando algoritmos de aprendizaje automático o machine learning, mediante técnicas de deep learning [247]. De este modo es capaz de entender lo que el usuario escribe y saber el sentimiento social de cada acción.

- MFC (Maximal Friendship Coefficient) es el Coeficiente Máximo de Amistad y representa el peso máximo de las posibles acciones positivas entre cualquier par de usuarios. Su valor vienen definido por: $max_{ij}(\sum_k(nA_{ij}^k \cdot wA_{ij}^k))$.

Cuando se completa una ruta, los usuarios que participaron en ella, pueden votarse entre sí con valoraciones que van desde 1 hasta 5 estrellas. Cada pasajero evalúa individualmente el conductor, y el conductor evalúa individualmente a cada pasajero. El peso en el sistema es mayor para las valoraciones del conductor a los pasajeros que de los pasajeros al conductor, ya que el conductor pone su vehículo a disposición de los pasajeros. Con el fin de agrupar las diferentes valoraciones de un usuario, se utiliza una simple media aritmética. El tipo de métrica que se tiene en cuenta para estas valoraciones se muestra en la Tabla 4.2. Se ha estimado como valor neutro la puntuación con 3 estrellas. Un valoración con 4 o 5 estrellas se considera como positiva y sobre ellos la valoración con 1 o 2 estrellas es procesada como negativa. Para los conductores, una valoración neutra se considera como positiva y las valoraciones positivas tienen más repercusión que las valoraciones positivas de los pasajeros. Por otra parte las valoraciones negativas tienen menos impacto en el conductor que en el pasajero. La equivalencia entre las estrellas y la valoración numérica procesada por el sistema se basa en el estudio publicado en [252]. La base principal de un sistema de este tipo se caracteriza por la capacidad de catalogar los diferentes tipos de comportamientos de los usuarios, ya sean buenos o malos. Por ejemplo, 2 estrellas es una mala puntuación, mientras que 3 estrellas es una puntuación aceptable. Por lo tanto, la diferencia entre obtener 2 u obtener 3 estrellas es más relevante que la diferencia entre obtener 3 u obtener 4 estrellas, ya que implica la transición de una mala puntuación hacia una puntuación aceptable. Si un usuario valora con 2 estrellas, significa que el viaje era malo. Sin embargo, si un usuario valora con 3 o 4 estrellas, es una señal de que el viaje no ha sido malo.

La métrica final $MeanR_j$ de valoración de las puntuaciones de un usuario j es un valor $\in [-0{,}75, 1]$ que viene dado por la ecuación 4.2.

Valoración de Estrellas	Impacto de la Valoración en el Conductor	Impacto de la Valoración en el Pasajero
1 estrella	-0.5 puntos	-0.666 puntos
2 estrellas	-0.25 puntos	-0.333 puntos
3 estrellas	0.5 puntos	0 puntos
4 estrellas	0.75 puntos	0.333 puntos
5 estrellas	1 punto	0.666 puntos

Tabla 4.2: Impacto de las Valoraciones

$$MeanR_j = \frac{\sum_i (PR_{ij} + NR_{ij})}{nR_j} \tag{4.2}$$

donde:

- PR_{ij} es la valoración positiva recibida por un usuario j de un usuario i. Se corresponde con la valoración de 3 o más estrellas.
- NR_{ij} es la valoración negativa recibida por un usuario j de un usuario i. Se corresponde con la valoración de 1 o 2 estrellas.
- nR_j es el número total de valoraciones recibidas por un usuario j.

La métrica que indica el grado de fiabilidad o tasa de reputación TR_{ij} de un usuario i sobre otro usuario j es un valor $\in [0, 1]$ que está definido por la ecuación 4.3.

$$TR_{ij} = \begin{cases} DT_{ij}, & \text{si el usuario i y el usuario j son amigos directos.} \\ IT_{ij}, & \text{si existe una cadena de amistad entre el usuario i y el usuario j.} \\ NT_{ij}, & \text{si no existe ninguna cadena de amistad el usuario i y el usuario j.} \end{cases} \tag{4.3}$$

donde:

- Si el usuario j es amigo directo del usuario i, se usa el valor de amistad directo DT_{ij} definido por la ecuación 4.4.

$$DT_{ij} = p_{ij} \cdot wP + MeanR_j \cdot wR \tag{4.4}$$

- Si el usuario j comparte una cadena de amistad con el usuario i, se usa el valor de amistad indirecta IT_{ij} definido por la ecuación 4.5.

$$IT_{ij} = (\prod_{(a,b) \in chain} p_{ab}) \cdot wP + MeanR_j \cdot wR \tag{4.5}$$

- Si no existe cadena de amistad entre el usuario i y el usuario j, se usa el valor NT_{ij} definido por la ecuación 4.6.

$$NT_{ij} = MeanR_j \cdot wR \tag{4.6}$$

donde:

- wP es el peso o importancia asociado a la amistad. Su valor estimado en 0,625 es resultado de la media de felicidad de los usuarios con sus amigos dentro de la red social Facebook [252].
- wR es el peso que se da a las valoraciones realizadas en la plataforma. Su valor es $(1 - wP)$, que en este caso se corresponde con el valor: 0,375.
- $\prod_{(a,b)\in chain} p_{ab}$ representa una cadena humana de amistades entre dos usuarios.

En caso de que exista más de una cadena de amistad se escoge la más corta, y en caso de empate, la más favorable. La relación entre el valor interno de Reputación TR del sistema y el valor ofrecido al usuario a través de la interfaz, es mostrado en la Tabla 4.3.

Tasa de Reputación	**Valor mostrado al Usuario**
[-0.25,0)	D
[0,0.25)	C
[0.25,0.5)	B
[0.5,0.75)	A
[0.75,1]	S

Tabla 4.3: Equivalencia Entre TR y el Valor Mostrado al Usuario

El valor máximo de reputación que un usuario puede recibir es de 1 (representado como la puntuación S para el usuario), lo que corresponde a la situación idílica en que ambos usuarios son amigos directos y han obtenido siempre las puntuaciones más altas posibles otorgadas por otros usuarios en la plataforma.

Un usuario puede tener una valoración total de reputación de otro usuario negativa o nula y recibir una D por ejemplo cuando no tiene ningún grado de amistad social con el otro usuario, y además tiene críticas negativas recibidas por la plataforma.

La reputación se calcula dinámicamente en base al grado de amistad que mantienen dos usuarios. Esto ayuda a los usuarios a tener una medida objetiva para decidir si confiar en otro usuario o no. Además, sólo los usuarios que tienen una valoración superior a B y/o los usuarios que han sido aceptados directamente por el conductor para hacer una ruta, pueden ver ciertos datos privados, tales como el número de teléfono del conductor o cualquier otro dato confidencial.

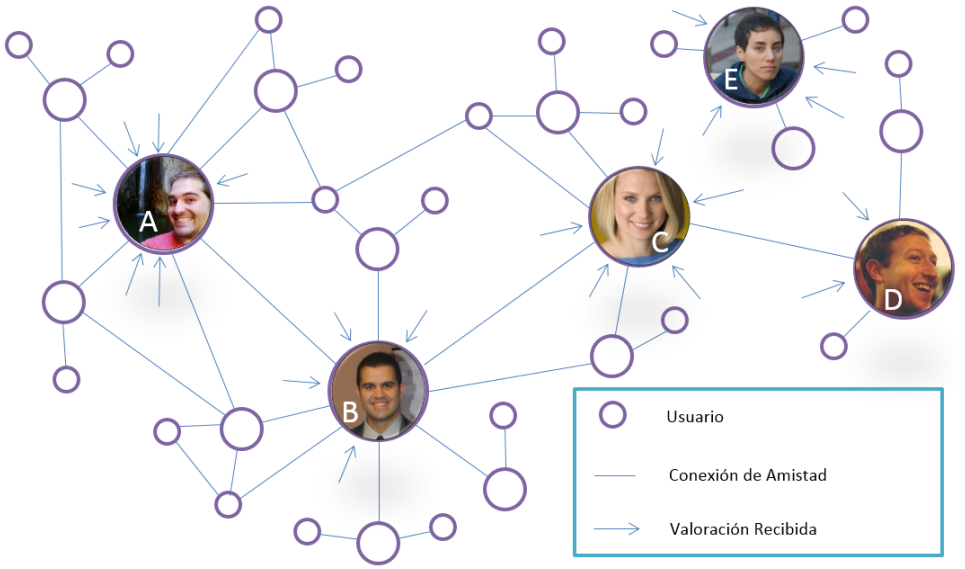

Figura 4.2: Ejemplo de la Red Gestionada por el Sistema

4.1.3. Implementación

El algoritmo diseñado se ejecuta en paralelo usando diferentes hilos para optimizar la eficiencia. Para el cálculo del valor IT_{ij}, que indica que dos usuarios no tienen una relación de amistad directa pero sí tienen una cadena de amistad, cada valor parcial P_{ab} se calcula en un hilo diferente. Una ejemplificación de la red utilizada por el sistema se muestra en la Figura 4.2 donde se puede ver el esquema completo de la interacción del usuario con el sistema.

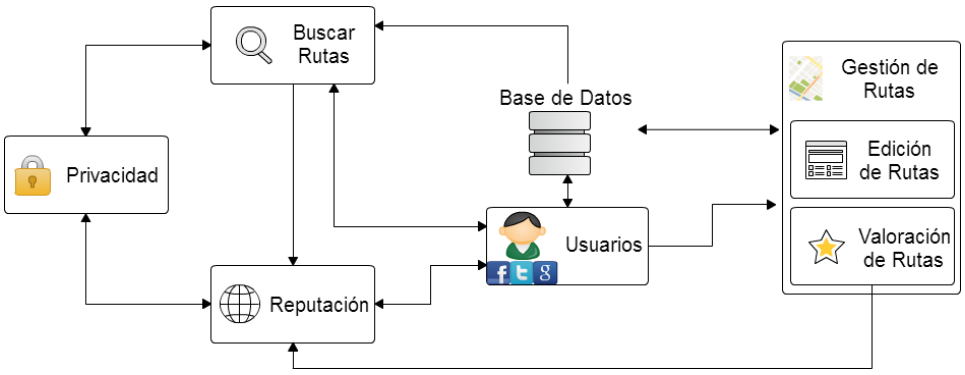

Figura 4.3: Diagrama de Caso de Uso

Con el fin de discernir si una acción tomada en una red social es positiva o negativa, el sistema propuesto utiliza AlchemyAPI, que a través de una red neuronal inteligente proporciona el sentimiento social de un texto. La

propuesta sólo tiene en cuenta las últimas 100 interacciones entre los usuarios en las redes sociales.

Cada valor TR_{ij} es variable o dinámico para cada par de usuarios ij. Este valor depende del grado de amistad y de las interacciones en las redes sociales de esos usuarios en concreto. Por otra parte, este valor se actualiza continuamente a medida que se vayan generando nuevas interacciones. Así, conforme pasa el tiempo el sistema se va nutriendo de nuevos comentarios y calificaciones de los usuarios.

Con el fin de analizar el funcionamiento del algoritmo de reputación propuesto, en la Tabla 4.4 se proporciona una muestra de parámetros generados mediante el sistema, teniendo en cuenta la red de la Figura 4.2.

Relación	**Datos sobre las Relación**	
Usuario A → Usuario B	likes en Facebook	78
	comentarios en Facebook (positivos - negativos)	19
	Valor de Amistad	*0.860*
Usuario B → Usuario A	likes en Facebook	82
	comentarios en Facebook (positivos - negativos)	12
	Valor de Amistad	*0.761*
Usuario B → Usuario C	likes en Facebook	53
	comentarios en Facebook (positivos - negativos)	8
	Valor de Amistad	*0.497*
Usuario C → Usuario B	likes en Facebook	4
	comentarios en Facebook (positivos - negativos)	2
	Valor de Amistad	*0.061*
Usuario C → Usuario D	likes en Facebook	76
	comentarios en Facebook (positivos - negativos)	21
	Valor de Amistad	*0.882*
Usuario D → Usuario C	likes en Facebook	42
	comentarios en Facebook (positivos - negativos)	7
	Valor de Amistad	*0.405*

Tabla 4.4: Ejemplo de Datos de Amistad con $MFC = 40{,}831$

La Tabla 4.5 muestra como ejemplo las valoraciones recibidas por los usuarios dentro de la aplicación una vez completadas las rutas.

A través de los datos de la Tabla 4.2 y la Tabla 4.5, y considerando como MFC el valor 40,831 para todos los usuarios, las tasas de reputación

	Usuario A	Usuario B	Usuario C	Usuario D	Usuario E
Valoraciones Recibidas como Conductor	4 estrellas 5 estrellas	3 estrellas 4 estrellas 4 estrellas	5 estrellas	-	2 stars 3 stars
Valoraciones Recibidas como Pasajero	4 estrellas 4 estrellas 5 estrellas 4 estrellas	5 estrellas 4 estrellas 2 estrellas	3 estrellas 4 estrellas 3 estrellas	2 estrellas 3 estrellas 2 estrellas	3 estrellas 1 estrellas 2 estrellas

Tabla 4.5: Muestra de Valoraciones

obtenidas a través del algoritmo descrito se pueden ver en la Tabla 4.6.

	Usuario A	Usuario B	Usuario C	Usuario D	Usuario E
Usuario A	-	0.704 (Puntuación A)	0.392 (Puntuación B)	0.152 (Puntuación C)	-0.075 (Puntuación D)
Usuario B	0.690 (Puntuación A)	-	0.435 (Puntuación B)	0.191 (Puntuación C)	-0.075 (Puntuación D)
Usuario C	0.242 (Puntuación C)	0.211 (Puntuación C)	-	0.468 (Puntuación B)	-0.075 (Puntuación D)
Usuario D	0.224 (Puntuación C)	0.187 (Puntuación C	0.378 (Puntuación B)	-	-0,075 (Puntuación D)
Usuario E	0.213 (Puntuación C)	0,172 (Puntuación C)	0.125 (Puntuación C)	-0,083 (Puntuación D)	-

Tabla 4.6: Muestra de Valores de Reputación

Por ejemplo, la confianza del usuario A sobre el usuario B, que son amigos directos, se muestra en la expresión 4.7.

$$\begin{aligned} DT_{AB} = \frac{(19 \cdot 0{,}727) + (78 \cdot 0{,}273)}{40{,}831} \cdot 0{,}625 + \\ + \frac{0{,}5 + 0{,}75 + 0{,}666 + 0{,}333 - 0{,}333}{6} \cdot 0{,}375 = \\ = \frac{35{,}107}{40{,}831} \cdot 0{,}625 + \frac{2{,}66}{6} \cdot 0{,}375 = \\ = 0{,}537 + 0{,}167 = 0{,}704 \end{aligned} \tag{4.7}$$

La tasa de confianza del usuario D sobre el usuario A, que no son amigos directos, se muestra en la expresión 4.8.

$$\begin{aligned} IT_{DA} = p_{DC} \cdot p_{CB} \cdot p_{BA} \cdot wP + \\ + MeanR_A \cdot wR = \\ = (0{,}405 \cdot 0{,}061 \cdot 0{,}761) \cdot 0{,}625 + \\ + \frac{0{,}75 + 1 + 0{,}33 + 0{,}33 + 0{,}66 + 0{,}33}{6} \cdot 0{,}375 = \\ = 0{,}019 \cdot 0{,}625 + \frac{3{,}41}{6} \cdot 0{,}375 = \end{aligned}$$

$$= 0{,}012 + 0{,}212 = 0{,}224 \tag{4.8}$$

Por lo tanto, el sistema propuesto se puede considerar totalmente dinámico en función de las relaciones entre cada par de usuarios. La Figura 4.4 muestra una comparación entre las tasas de confianza entre cada par de usuarios y la valoración típica que se obtendría mediante los sistemas convencionales. En particular, la Figura 4.4 muestra en el eje Y el nivel de confianza y en el eje X el usuario en cuestión.

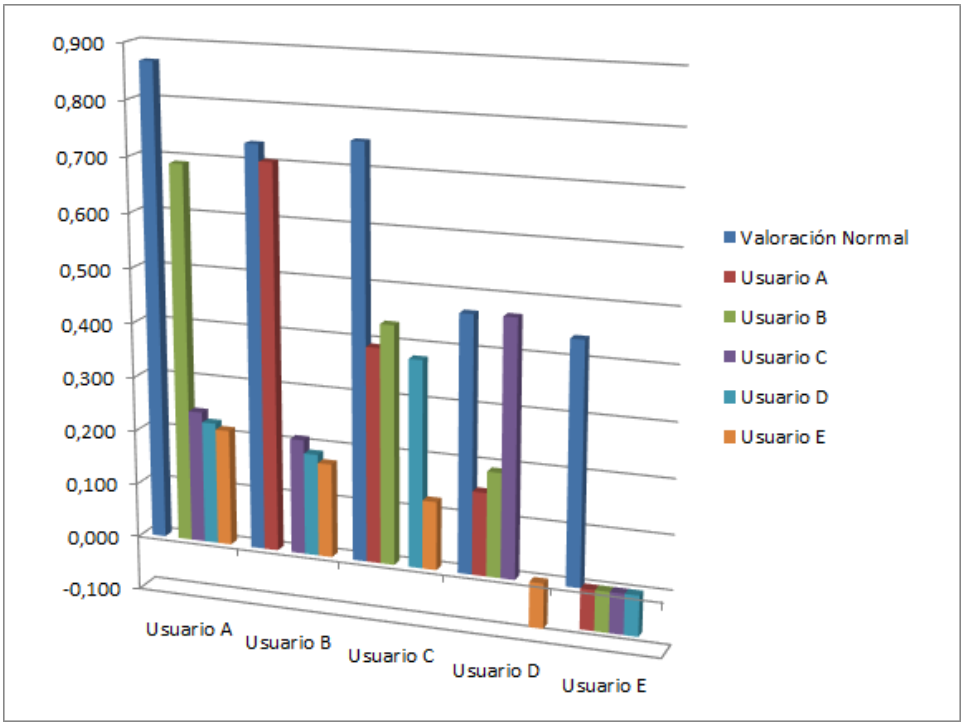

Figura 4.4: Propuesta VS Sistemas de Valoración Clásicos

Para el cálculo de la valoración dentro del sistema, se han adoptado los datos mostrados en la Tabla 4.5, teniendo en cuenta las puntuaciones con los valores equidistantes entre 0 y 1 para representar las valoraciones con 1, 2, 3, 4 o 5 estrellas.

Aunque el sistema descrito en este trabajo se puede combinar con plataformas existentes, para una primera prueba de concepto, se ha diseñado una nueva aplicación para Android que ya se ha publicado en la Google Play Store bajo el nombre de Carpoolap (ver Figura 4.5).

La aplicación Android Carpoolap se ha desarrollado para las versiones 3.0 o superiores del sistema operativo más usado para dispositivos móviles. Para ello se ha implementado usando algunas de las funcionalidades más avanzada del sistema operativo, como la API de Google Maps en su versión 3.0, la API de Google Places, la API de mensajería en la nube de Google, el

Figura 4.5: Capturas de Pantalla

SDK de Facebook en su versión 3.0 y múltiples librerías y servicios. Por lo tanto, la aplicación facilita al usuario el autocompletado en las búsquedas de direcciones, mapas en 3D, diseño minimista y plano basado en las últimas versiones de Android, notificaciones push con las peticiones o respuestas de pasajeros o conductores, etc.

Cada usuario puede ver las rutas que ha propuesto como conductor, y si hay pasajeros potenciales para esas rutas. Además, con colores llamativos, los usuarios pueden conocer las rutas que ya han realizado y las rutas que tienen planificadas para el futuro. Para la valoración de los usuarios que participan en una ruta, después de terminar un trayecto cada uno puede dar una puntuación. El lado del servidor, donde se realiza toda la lógica de los algoritmos inteligentes de reputación, y donde se almacenan de forma segura todos los datos, se ha desarrollado utilizando tecnologías JavaScript mediante frameworks como Node.js y Express.js. Como base de datos para los datos centralizados en el servidor, se ha decidido adoptar una base de datos No SQL, como MongoDB [190]. Se ha desplegado el servidor en una instancia de Amazon Web Services, específicamente en una máquina Ubuntu con cuenta de Amazon EC2.

La aplicación ha sido instalado por más de 5.000 usuarios distintos en todo el mundo, y ha generado más de 1000 viajes diferentes. La información recibida de los usuarios ha sido positiva, con una puntuación media de más de 4 estrellas de 5 en la Google Play Store. Actualmente se encuentra integrada con Facebook, por lo que la integración con otras redes sociales sería la siguiente evolución de la plataforma. Además está preparada para disponer de una API para que plataformas de terceros puedan integrar el algoritmo inteligente de reputación en sus sistemas.

4.1.4. Seguridad

En cuanto a la seguridad de la plataforma, puede haber diferentes motivos por los que un usuario quiera atacar el sistema. Por un lado, la razón principal para un atacante puede ser la satisfacción de romper un sistema que se considera seguro. Otra motivación puede ser acceder a la información sensible o conseguir datos privados para su uso fraudulento. Además, para este esquema en particular, un usuario puede desear aumentar su estatus social en el sistema aumentando su puntuación de forma ilegítima. Por último, un usuario malintencionado puede querer denigrar a otro usuario modificando su puntuación con valoraciones bajas, para que ese usuario legítimo y con un buen comportamiento de repente empiece a ser considerado por otros como un mal usuario para compartir ruta.

El Sybil Attack o ataque Sybil es uno de los ataques más frecuentes y notorios de los sistemas tradicionales de carpooling. Este tipo de ataques son ataques típicos de redes P2P (Peer to Peer), donde un dispositivo malicioso toma múltiples identidades falsas. Debido al entorno de los esquemas de carpooling, donde se busca ante todo preservar la privacidad de cualquier usuario, una vulnerabilidad de tipo Sybil es generalmente difícil de combatir.

En un ataque de tipo Sybil, el atacante va en contra del sistema de reputación de una red P2P mediante la creación de un gran número de identidades con seudónimo falsos. De esta manera consigue obtener una gran reputación en un periodo corto de tiempo, inyectando puntuaciones y valoraciones positivas de manera fraudulenta, desde las diferentes cuentas generadas de forma masiva. La vulnerabilidad de un sistema de reputación ante un ataque Sybil depende de cómo de fácil sea la generación de nuevas identidades. Por tanto, un sistema donde se trate a todos los usuarios por igual es más propenso a este tipo de ataques que un sistema que se basa en cadenas de confianza entre usuarios, como es el caso de la propuesta aquí descrita.

Una entidad en una red P2P es una porción de software que tiene acceso a los recursos locales. Las entidades son anunciadas en la red P2P mediante una determinada identidad, pero una entidad puede adoptar diferentes identidades. En otras palabras, la asignación de las identidades a las entidades es de muchos a uno. Las entidades en estas redes utilizan múltiples identidades con el propósito de conseguir que los sistemas sean redundantes, para

el intercambio de recursos, la fiabilidad y la integridad. En las redes P2P, la identidad se utiliza como una abstracción de modo que una entidad remota puede conocer las identidades de otras entidades sin que sea necesario saber a que entidad representan.

Un usuario deshonesto que lanza un ataque Sybil adopta múltiples identidades en una red P2P para hacerse pasar por múltiples nodos. A través del uso de múltiples identidades, el adversario puede llegar a controlar la red.

Aplicado un ataque Sybil al esquema propuesto, se corresponde por ejemplo con el siguiente escenario. Un usuario malicioso (B0) genera varias cuentas falsas en las redes sociales (B1, B2, B3, etc.). Este usuario genera y anuncia un posible viaje $X \mapsto Y \mapsto Z$. Para el trayecto ($X \mapsto Y$) el usuario fraudulento utiliza las cuentas falsas para hacer creer que su vehículo está lleno y que en el punto Y se queda vacío. De esta manera, puede obtener valoraciones positivas y aumentar su reputación de forma fraudulenta. Con ello, consigue que para el trayecto $Y \mapsto Z$, el resto de usuarios vean que tiene un número de valoraciones muy buenas en el trayecto justo anterior.

Sin embargo, el método de reputación propuesto se basa en la confianza entre usuarios a partir de su amistad, de modo que si un usuario no es amigo de otro usuario, no es posible generar la cadena de amistad pertinente, y menos aún generar un número suficiente de interacciones en las redes sociales. Por lo tanto, aunque un adversario intente engañar a través de la generación de valoraciones falsas, se necesita un vinculo fuerte de amistad para que el sistema establezca una reputación aceptable y así otorgar más privilegios.

Por tanto, el algoritmo propuesto reduce, de manera significativa, la vulnerabilidad al ataque Sybil, porque la mayoría de la puntuación del algoritmo está calculado en base a la confianza generada por la cadena de amistad que une a cada par de usuarios. Por lo tanto, si un usuario no conoce (en absoluto) a otro usuario, aún que este tenga valoraciones muy buenas en el sistema, para el primero no es lo suficientemente fiable como para compartir un viaje. Cabe destacar que en nuestro sistema, en base a la media de felicidad actual de Facebook, se ha usado como peso de las valoraciones de los usuarios es de 37,5 % del total de la tasa de reputación, mientras que la fiabilidad de la amistad es del 62,5 %, pero estos pesos son configurables en caso de que haya estudios que sugieran una modificación.

4.2. DEPHISIT

Actualmente el despliegue real de una VANET parece más una utopía que un hecho que se pueda llevar a cabo de forma generalizada. Debido a la crisis económica y a las grandes necesidades que requiere una red de este tipo, parece casi inviable que un gobierno apueste por un sistema de este tipo, teniendo que instalar RSUs a lo largo de todas las carreteras. Además según la definición original, los usuarios tendrían que adaptar sus

vehículos para la utilización de OBUs o cambiarlos, lo que conllevaría a un desembolso de dinero que muchos no están dispuestos a realizar. Por ello, las investigaciones y experimentos relacionados con las redes vehiculares inteligentes nunca han prosperado más allá de demos técnicas y escenarios limitados. En esta sección se explica una plataforma que ha sido desarrollada dentro del seno del grupo de investigación donde el autor de la presente Tesis ha realizado sus investigaciones de doctorado, junto a otras universidades y departamentos de investigación de grandes empresas en el marco del proyecto nacional [114]. Dicha plataforma pretende llevar a la realidad muchas de las funcionalidades de las redes vehiculares, utilizando únicamente dispositivos móviles y redes de sensores low cost, para sustituir en el concepto de las VANETs a las RSUs y a las OBUs. Un trabajo que describe la estructura general de la plataforma, ha sido aceptado recientemente en un congreso clasificado como CORE A [168]. Además se detalla en particular una de las funcionalidades de la plataforma que ha sido totalmente implementada y publicada en un congreso internacional [159].

4.2.1. Plataforma

DEPHISIT es el nombre dado al proyecto en el marco del cual se ha diseñado e implementado la plataforma en cuestión. Sus siglas corresponde con el título: Desarrollo Experimental de una Plataforma Híbrida Inalámbrica para Sistemas Inteligentes de Transporte. El resultado final de dicho proyecto incluye un conjunto de varias aplicaciones para vehículos que pueden llegar a significar una mejora sustancial en el nivel de eficiencia y seguridad vial [46]. Además, permite gestionar en tiempo real las condiciones del tráfico con el fin de fomentar una conducción cooperativa, mostrando información de interés tanto para los conductores como para los peatones. La plataforma representa un paso tecnológico importante en el desarrollo práctico de una VANET ya que proporciona grandes innovaciones tecnológicas en comparación con los sistemas disponibles en la actualidad. La red inalámbrica híbrida desarrollada funciona como una VANET, permitiendo recibir información en tiempo real a través de conexiones GSM/UMTS/LTE y puntos de acceso WiFi. Por un lado, mediante el uso de técnicas de cifrado, se logra la transferencia de información sensible de forma segura. Por otro lado, el uso de diferentes puntos de acceso proporciona la ventaja adicional de tener un coste menor frente a algunos sistemas específicos que utilizan únicamente conexiones de datos. Por lo tanto, esta plataforma se puede utilizar en aplicaciones que hasta ahora estaban restringidas a áreas como puertos, aeropuertos, zonas militares, zonas de desastre [151], etc., llegando a un público más amplio.

DEPHISIT ofrece dos modos de funcionamiento en función de si los usuarios tienen acceso a Internet o no. En particular, la arquitectura utilizada en el sistema se muestra en detalle en la Figura 4.6.

La arquitectura general del sistema se basa en el modelo de aplicación

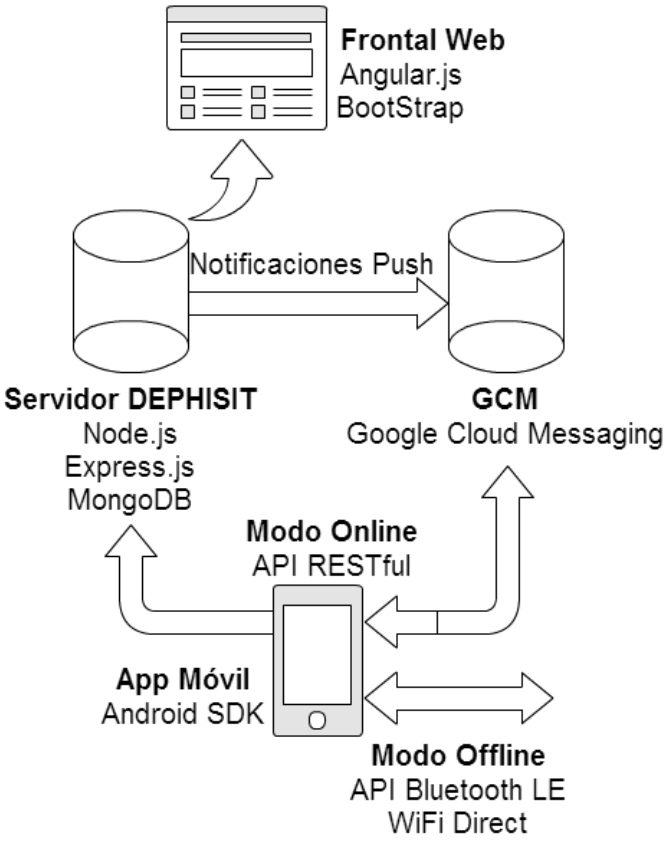

Figura 4.6: Arquitectura de la Plataforma DEPHISIT

cliente-servidor. Existen varios tipos de clientes. Por un lado se encuentran los dispositivos móviles que se utilizan como OBUs en el sistema. Por otro lado se hallan las plataformas de sensores que permiten detectar eventos y comunicarse con los dispositivos móviles. Del lado del servidor, se encuentra un barebone dedicado que sirve de nexo entre los diferentes dispositivos móviles, para realizar las comunicaciones en tiempo real y poder almacenar toda la información que se genera en la red.

La arquitectura online está destinada para los usuarios que tienen acceso a Internet, lo que es frecuente hoy en día, sobre todo en las zonas urbanas, donde hay una buena cobertura de las redes móviles y existen redes inalámbricas públicas. La arquitectura offline puede ser utilizada por aquellos usuarios que en un determinado momento no poseen acceso a Internet. Esto permite que DEPHISIT esté diseñado para funcionar también en zonas remotas o zonas rurales donde no existe acceso a Internet. Para ello, se crea una red ad-hoc que permite la comunicación directa entre los vehículos (dispositivos móviles). Para la creación de esta red ad-hoc, se utilizan dos tipos de tecnologías inalámbricas: Wi-Fi Direct y Bluetooth Low Energy.

Si un vehículo quiere informar de un evento, debe generar un mensaje en modo broadcast para que todos los vehículos de su alrededor puedan estar al tanto del evento que se ha generado. Además, DEPHISIT posee una arquitectura de saltos, que permite que el evento se propague por la red mediante comunicaciones en cadena. De esta manera, un usuario que no está en el rango adecuado donde se originó un determinado evento, podrá recibirlo a través de un vehículo que lo ha recibido o bien por estar en el rango adecuado, o bien porque a su vez lo ha recibido de otro vehículo.

En un sistema de comunicación ad-hoc con envío masivo de mensajes a la red, es posible que la generación masiva de estos mensajes puedan sobrecargar la red. Por tanto, DEPHISIT utiliza la idea explicada en [189] para evitar la sobrecarga de la red. Dicha idea se basa en agrupaciones de un solo salto para reducir el número de comunicaciones en las VANETs, manteniendo su seguridad, en escenarios donde existe un elevado nivel de tráfico.

Con el fin de certificar la autenticidad de los usuarios de la red, se utiliza el esquema de autenticación explicado en el capítulo 2 de la presente memoria. Para escenarios donde existan más de 100 usuarios concurrentes, la plataforma posee una arquitectura adaptativa que usa grupos como los propuestos en [35].

Entre las funcionalidades con las que cuenta la plataforma, los dispositivos móviles son capaces de detectar de manera autónoma y automática los siguientes eventos:

- Colisiones: El smartphone utiliza sensores internos que calculan las fuerzas G producida. Esto se consigue gracias a la combinación de los valores de los giroscopios y los acelerómetros. El sistema tiene establecido como umbral de colisión el valor de 2.7, ya que es el promedio de fuerzas G que se producen en una colisión entre vehículos [48].

- Detección de atasco: Los dispositivos móviles que utilizan las aplicaciones de DEPHISIT son capaces de detectar si la velocidad de un vehículo se ha reducido anormalmente durante un cierto período de tiempo. Esto provoca la generación de un evento de posible congestión en ese tramo de carretera. Si varios vehículos en la misma zona generan este mismo tipo de evento, entonces se confirma la aparición de un atasco de tráfico.

- Aviso de señales de tráfico: La aplicación es capaz de alertar acerca de ciertas señales de tráfico en zonas peligrosas. Estas señales han sido precargadas en el sistema a través de su geolocalización. De esa manera, el teléfono inteligente sabrá si está en las proximidades de una de estas señales y será capaz de advertir al usuario para que pueda cumplir con su obligación.

- Notificación de plazas de aparcamiento: El dispositivo móvil detecta

cuando se arranca el vehículo y se pone en marcha. En dicho caso, se genera un evento de posible plaza de aparcamiento libre, para que los vehículos de la zona puedan detectar donde es posible que puedan dejar su vehículo.

- Aviso de infracción en semáforos. El sistema es capaz de detectar de forma anónima cuando un usuario se ha saltado el semáforo, para alertar a los usuarios de la zona. Esta funcionalidad es la más compleja y por tanto la que se explicará con más detalle en las próximas subsecciones.

4.2.2. Infracción en Semáforos

Un problema particularmente difícil de resolver en la seguridad vial es la detección de usuarios que infringen los semáforos en rojo. Existen varias causas por la que los usuarios llegan a saltarse un semáforo. Por ejemplo, la duración en la que un semáforo está en verde puede ser una de estas causas. Los semáforos con una duración muy corta pueden causar que los usuarios ignoren cuando se pone en rojo, lo que puede producir un efecto dominó que puede causar muchos accidentes en determinadas intersecciones. Con el fin de tratar de resolver el problema de este tipo de infracciones, se han propuesto en la bibliografía científica diferentes soluciones que van desde nuevos mecanismos para señalar las intersecciones, a cámaras de velocidad que fotografían a los infractores, entre muchas otras. Algunas de estas investigaciones, como se comenta más tarde, han sido puestas en práctica con éxito, y han llegado a provocar la reducción de accidentes por saltarse semáforos en los centros urbanos de algunas de las ciudades más importantes del mundo. Por tanto estas soluciones pueden en muchos casos considerarse eficaces, aunque no es posible adoptarlas en todas las ciudades, ya que su costo es demasiado elevado para algunos gobiernos locales. También se plantean en el caso de las cámaras, posibles conflictos legales relacionados con la privacidad de los usuarios.

Los autores del trabajo [214] han concluido que los accidentes más frecuentes y peligrosos en entornos urbanos tienen que ver con las infracciones en semáforos, las infracciones en señales de STOP y otros tipos de infracciones concernientes a las intersecciones. En concreto, de acuerdo con un informe de la Administración Nacional de Seguridad del Tráfico en las Carreteras de los Estados Unidos o NHTSA (National Highway Traffic Safety Administration) [198], se registraron en 2014 más de 2.3 millones de accidentes en intersecciones en los Estados Unidos, que provocaron más de 7770 muertes y aproximadamente 733000 lesiones severas. El sistema de información de análisis de víctimas mortales de la NHTSA señala que las colisiones provocadas por las infracciones en semáforos causan cerca de 762 muertes anuales, y que 165000 personas sufren lesiones severas cada año en este tipo de accidentes. Además, el Instituto para Seguridad en las Carreteras o IIHS

(Insurance Institute for Highway Safety) [83] informa que la mitad de las personas que mueren en accidentes provocados por alguna infracción en semáforos, no son los infractores, sino otros conductores y peatones afectados.

Referente a las medidas implementadas en la práctica, se han propuesto diferentes soluciones para tratar de reducir este tipo de infracciones. Por ejemplo, en [213] se propone tres soluciones. Una de ellas propone sustituir las intersecciones con semáforos por rotondas. Otra indica la necesidad de incrementar el tiempo en el que el semáforo se sitúa en ámbar. La última quiere instaurar un periodo de tiempo síncrono donde todos los semáforos de una intersección estén en rojo al mismo tiempo. Sin embargo, los resultados muestran que ninguna de estas medidas ha llegado a triunfar completamente como para ser considerada una solución global.

Una de las posibles aplicaciones de ITS para proporcionar una solución al problema de estas infracciones, son los semáforos inteligentes, que permiten, por ejemplo, cambiar de forma dinámica la duración de los semáforos para los peatones ciegos. Además, los semáforos inteligentes también pueden ser auto-controlables para maximizar el tráfico de un determinado tramo de carretera. Otra de las funcionalidades que tienen, es la de poder denunciar a los usuarios que los infrinjan [64] [51]. Esta última solución, llamada red light camera o foto-rojo, lleva en funcionamiento desde hace varios años en muchas regiones del mundo [24]. Una cámara de este tipo no es más que un radar de tráfico que captura automáticamente una imagen de cualquier vehículo que se salta un semáforo. Acto seguido envía esta fotografía al sistema de control de semáforos de modo que la foto se puede utilizar como prueba para las autoridades pertinentes. En general, la cámara se activa cuando un vehículo entra en la intersección justo después de que el semáforo se situé en rojo. Como ya se dijo, esta solución tiene varios problemas de costo y privacidad.

En [99], los autores presentan un sistema de semáforos adaptables basados en comunicaciones inalámbricas entre vehículos y controladores fijos desplegados en las intersecciones. Tal sistema utiliza concretamente la comunicación inalámbrica de corto alcance entre los vehículos para comunicarse con el controlador colocado en la intersección, que es el encargado de determinar los valores óptimos para las diferentes fases del semáforo.

Google [77] también posee varios métodos para la detección automática de los estados de los semáforos y así poder detectar de forma robusta el color de la luz del semáforo por parte de los vehículos. Utilizan estos métodos para detectar el estado de más de cuatro mil semáforos, y así permitir que miles de conductores puedan actuar de forma consciente en estas intersecciones.

El trabajo [2] propone el uso de tecnología RFID para detectar el estado del semáforo de forma dinámica y poder evitar los problemas que suelen surgir con los sistemas que utilizan técnicas de procesamiento de imágenes. La tecnología RFID se aplica en la intersección de múltiples carretera, con varios carriles, y con multitud de vehículos, para ofrecer un sistema de gestión

eficiente. Se han realizado diferentes simulaciones en entornos reales, que han concluido que el sistema podría emular el trabajo de un policía de tráfico en una intersección congestionada.

Otra de las propuestas innovadoras es la creación de un semáforo en una carretera de seis vías y cuatro intersecciones, mediante el uso del microcontrolador PIC16F877A [124]. El sistema funciona de manera inteligente a través del sistema de control de tráfico, mejorando los tiempos de espera de los vehículos y haciendo más eficiente la situación provocada por situaciones de emergencia, llegando a sugerir posibles rutas alternativas.

A diferencia de todas las propuestas mencionadas, el objetivo principal de la aquí descrita es la creación de un sistema para detectar infracciones en semáforos con el fin de advertir a los conductores y peatones cercanos, y evitar posibles accidentes. Según nuestros datos, no hay ninguna propuesta parecida que permita notificar a los vehículos en una zona concreta donde existe un vehículo que ha infringido un semáforo. Tampoco existe una solución que permita a un vehículo informar de que ha violado la ley saltándose el semáforo, de forma completamente anónima. El anonimato es necesario en nuestro sistema para fomentar su. Las autoridades también pueden beneficiarse mediante el análisis de los datos generados por el sistema, con el fin de detectar aquellos semáforos más propensos a infringir y optimizar así, el tiempo requerido para cada estado de cada uno de los semáforos.

4.2.3. Funcionamiento

El sistema propuesto funciona como se explica a continuación. Cuando un vehículo se aproxima a un semáforo, recibe una baliza con el estado del semáforo, advirtiendo al conductor en caso de que esté en color rojo. En caso de que no cumpla con su obligación de detenerse, el dispositivo móvil del conductor automáticamente detecta la infracción y envía una advertencia en modo evento de la plataforma DEPHISIT a todos los vehículos cercanos para que los conductores puedan recibir una advertencia en tiempo real de la situación del peligro que existe en sus inmediaciones.

El sistema implementado utiliza sensores, teléfonos inteligentes y servidores en la nube para detectar automáticamente la infracción, y de forma anónima informar del peligro que ha ocasionado una infracción [129]. La Figura 4.7 muestra una visión general del funcionamiento del sistema.

Como se puede ver en el esquema, cada semáforo está enviando continuamente mensajes en forma de balizas con su estado. Si está en ámbar o rojo, el dispositivo móvil que recibe el estado comprueba la velocidad y la dirección del vehículo. En caso de detectar que el semáforo ha sido infringido, se envía un evento al servidor de DEPHISIT. El servidor es el encargado de localizar a los vehículos cercanos y enviarles una notificación en tiempo real a sus dispositivos móviles con el fin de alertar a sus conductores.

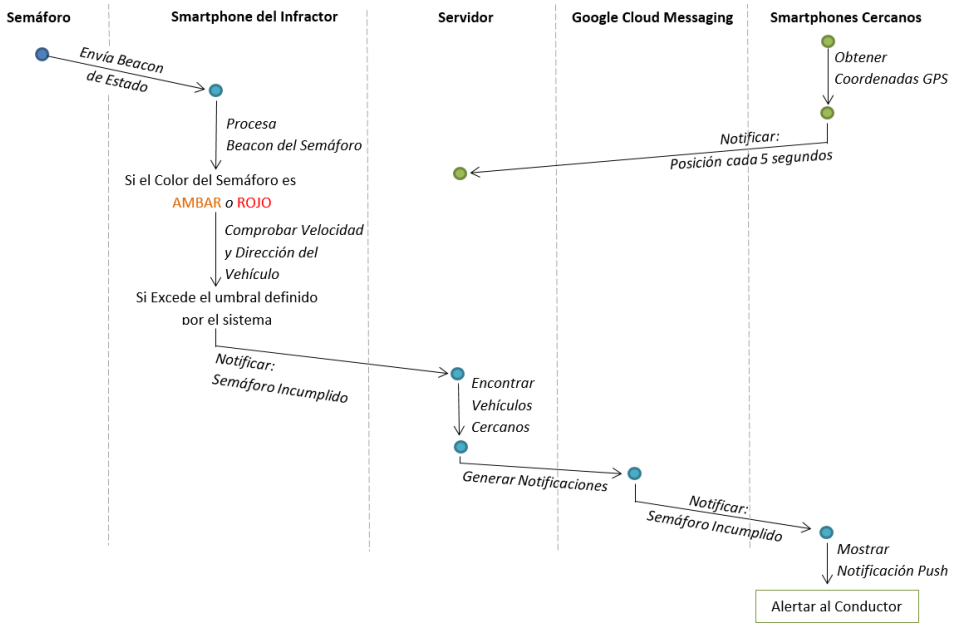

Figura 4.7: Funcionamiento del Sistema de Infracción de Semáforos

Las principales tecnologías utilizadas por el sistema propuesto en los semáforos se detallan a continuación.

- Sensores de luz para detectar en tiempo real el estado de los semáforos.
- Módulos Bluetooth Low Energy (BLE) que permiten transmitir continuamente el estado de los semáforos a los vehículos cercanos a través de mensajes de notificación.
- Dispositivos Arduino que permiten procesar enviar y recibir los datos generados en el semáforo mediante el módulo BLE.

Por ello, se considera que el sistema propuesto está basado en comunicaciones I2V y V2I a través de los siguientes elementos:

- Plataforma de sensores: Localizada en los semáforos y descrita anteriormente que permite comunicarse con los smartphones cercanos.
- Smartphones: Usados para identificar a los vehículos y servir de nexo entre el usuario y el sistema.
- Servidor en la nube: Responsable de procesar los datos generados en el sistema para notificar a los vehículos cercanos de que existe un peligro.

Como se ha mencionado, el sistema puede también ser utilizado por las autoridades de tráfico para detectar aquellos semáforos que son menos respetados. De esa manera, se pueda establecer un plan de acción e investigar las

causas para buscar soluciones (elevar el tiempo de duración de cada estado, etc.).

4.2.4. Seguridad

El sistema propuesto protege el anonimato del usuario, así como la integridad y autenticidad de la información, con el fin de promover el uso de la aplicación. El objetivo no es detectar a los usuarios fraudulentos para castigarlos, como ocurre con los semáforos foto-rojo, sino poder advertir al resto de usuarios que hay un peligro cercano, para que ellos estén más atentos. Por lo tanto, se necesita un sistema de protección de anonimato confiable y seguro para inspirar confianza a todos los usuarios y que les anime a hacer uso de la aplicación. Para lograr este objetivo la propuesta utiliza un servidor en la nube, una plataforma de sensores y teléfonos inteligentes. Los teléfonos inteligentes se utilizan para identificar a los vehículos. La plataforma de sensores sirve para otorgar inteligencia a los semáforos y comunicarse con los teléfonos inteligentes. El servidor es el responsable de notificar del peligro a los vehículos cercanos.

Con el fin de mantener el nivel de seguridad requerido, se utiliza OpenSSL dentro de la aplicación. OpenSSL es una implementación de los protocolos SSL y TLS, desarrollada como código libre. Soporta múltiples algoritmos criptográficos, muchos de ellos estándares. En particular, este trabajo utiliza la versión 1.0.2g publicada en Marzo de 2016. Para el establecimiento de un canal de comunicación seguro, se ha implementado una autoridad de certificación en el servidor, a través de OpenSSL.

La integridad del mensaje y la autenticidad del emisor se protege mediante el uso de un esquema de firma digital. Por tanto, el vehículo utiliza su clave privada durante el proceso de firma digital del mensaje enviado al servidor, y el servidor utiliza la clave pública del usuario para verificar la firma digital del mensaje. En concreto, el sistema usa el algoritmo de firma digital basado en curvas elípticas ECDSA (Elliptic Curve Digital Signature Algorithm) [119] que ofrece una variante del algoritmo de firma digital o DSA (Digital Signature Algorithm).

La implementación está basada en el esquema de firma digital configurado con los siguiente parámetros, donde $\times$ denota la multiplicación de una constante por un punto en la curva elíptica:

- *Curva*: Conjuntos de puntos (x, y) que satisfacen una ecuación definida sobre un cuerpo K, que cuando su característica no es ni 2 ni 3, toma la forma $y^2 = x^3 + ax + b$, $a, b \in K$.

- G: Punto base de la curva elíptica, generador de la *Curva*.

- n: Orden entero del punto G, por lo que $n \times G = O$, punto en el infinito.

- d_A: Clave privada representada como un entero elegido en el intervalo $[1, n-1]$.
- Q_A: Clave pública representada como un punto de la curva denotado por $Q_A = d_A \times G$.
- m: Mensaje a firmar.

De esta manera, por un lado, se ha diseñado el Algoritmo 9 para firmar un mensaje m.

Algorithm 9: Algoritmo de Firma

1 Calcular $e = h(m)$, donde $h(\cdots)$ es la función hash SHA-3;
2 Dado z, sean L_n los bits más a la izquierda de e, donde L_n es la longitud de bit de n;
3 Seleccionar un número entero aleatorio criptográficamente seguro k a partir de $[1, n-1]$;
4 Calcular el punto de la curva $(x_1, y_1) = k \times G$;
5 Calcular $r = x_1 \bmod n$. Si $r = 0$, volver al paso 3;
6 Calcular $s = k^{-1}(z + rd_A) \bmod n$. Si $s = 0$, volver al paso 3;
7 La firma es el par (r, s)

Por otro lado, el Algoritmo 10 permite al servidor verificar cada firma recibida.

Algorithm 10: Algoritmo de Verificación de Firma

1 Comprobar que Q_A no es igual a la identidad del elemento O;
2 Comprobar que Q_A pertenezca a la curva;
3 Comprobar que $n \times Q_A = O$;
4 Verificar que r y s son enteros en el rango $[1, n-1]$. En otro caso, la firma es inválida;
5 Calcular $e = h(m)$, donde $h(\cdots)$ es la misma función SHA-3 usada en la generación de la firma;
6 Dado z, sean L_n los bits más a la izquierda de e;
7 Calcular $w = s^{-1} \bmod n$;
8 Calcular $u_1 = zw$ mód n y $u_2 = rw \bmod n$;
9 Calcular el punto de la curva $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$;
10 La firma es válida si $r \equiv x_1$ mód n. En otro caso es inválida

Con el fin de proteger el anonimato del usuario, se utiliza el concepto de k-anonimato para la firma digital. El concepto de k-anonimato fue formulado por primera vez en [238] para intentar resolver el problema de los datos estructurados de una persona específica, produciendo un nuevo método que

permite distribuir dichos datos con la garantía de que es imposible identificar a las personas propietaria de los datos, sin afectar al contenido de los datos.

En particular, se dice que los datos pueden ser distribuidos con k-anonimato si la información revelada de un individuo no se puede distinguir de al menos $k-1$ individuos cuya información también aparece en la distribución.

El esquema implementado garantiza el k-anonimato a través de las ideas propuestas en [34], de acuerdo con que cada usuario se asocia aleatoriamente a un determinado grupo en el que comparte todos los datos criptográficos, como puede ser las claves privadas, las claves púbicas y un certificado de grupo utilizado para la firma. De esta manera, los usuarios no revelan sus identidades particulares sino sólo su identificador de grupo.

4.2.5. Sensores

Los sistemas de detección para ITS se basan en los sistemas instalados en los vehículos y en las infraestructuras, es decir, en las tecnologías aplicadas. La infraestructura de sensores son por lo general, dispositivos robustos que se instalan en la carretera. Estos sensores pueden ser instalados durante la construcción de las propias carreteras o inyectados a través de maquinaria especializada. Hay muchos tipos de sensores: contadores del número de vehículos, estaciones meteorológicas, cámaras para detectar atascos de tráfico, radares para detectar altas velocidades, etc. Estos sensores pueden ser de tipo desde muy básico (por ejemplo, sensores para detectar el número de vehículos en un tramo de carretera) a muy avanzado (como cámaras para detectar vehículos). Por lo general, los sensores más complejos son los más caros. Por ejemplo, las cámaras de detección visual de vehículos son un sistema muy costoso, y se suelen utilizar en contados casos para detectar las infracciones en los semáforos.

Con el fin de añadir inteligencia a los semáforos, el sistema propuesto utiliza un sistema muy baratos basado en un sensor de luz que proporciona información en tiempo real sobre el color del semáforo. Esto, junto con un módulo Bluetooth Low Energy (BLE), permite transmitir el estado del semáforo hacia los vehículos cercanos, mediante una notificación de tipo señalización dentro de DEPHISIT.

Los sensores Bluetooth utilizados son capaces de detectar las direcciones MAC Bluetooth de cada dispositivo móvil. Si estos sensores están interconectados de forma inteligente, son capaces de proporcionar datos de valor. En comparación con otras tecnologías usadas para entornos vehiculares, la tecnología Bluetooth aplicada a este escenario tiene algunas diferencias:

- Alta precisión.
- Permite instalar de forma rápida los dispositivos.

- El número de dispositivos que pueden emitir desde dentro de un vehículo puede limitarse.
- Ofrece la alternativa de mediciones no intrusivas de bajo coste.
- Permite instalar sensores de forma permanente o temporal.

La plataforma de sensores utilizada consiste en varios módulos electrónicos que componen un sistema pequeño, totalmente integrable en cualquier tipo de semáforo.

El sistema utiliza una placa RFDuino [215], que es un tipo de Arduino del tamaño de un dedo que emite de manera inalámbrica. Exactamente se utiliza el modelo 2216, con carcasa de doble batería AAA. La placa tiene un regulador conmutable que permite que las baterías puedan consumir en voltajes bajos sin dejar de ofrecer un voltaje estable de 3.3V al sistema.

El módulo Bluetooth de baja energía utilizado en el RFDuino es el modelo RFD22102 RFduino DIP. Este módulo tiene las especificaciones técnicas que se muestran en la Tabla 4.7.

Especificación	**Valor**
Número de Modelo	RFD22102
Categoría	Modulo Bluetooth LE RF
Tipo	Transceptor / Controlador
Banda	2.4 GHz
CPU	16MHz ARM Cortex-M0
Flash	128kb
Ram	8kb
MultiFrecuencia	Sí
Tipo de Paquete	DIP RFduino Footprint
Empaquetado	Bulk Clamshell
RoHS Compliant	Sí
Voltaje Bajo	1.9V
Voltaje Normal	3V
Voltaje Alto	3.6V
Transmisión	18mA, 4uA ULP
Recepción	18mA, 4uA ULP
Regulado por FCC	Sí
Aprobado por IC	Sí
Testeado por ETSI - CE	Sí
Energía de Transmisión	4dbm

Tabla 4.7: Especificaciones del Módulo BLE RFD22102

El formato típico de un mensaje BLE incluye un preámbulo de 1 byte,

4 bytes de códigos de acceso correlacionados con el número de canal RF utilizado, una unidad de datos de paquetes o PDU (Packet Data Unit) que puede ser de entre 2 y 39 bytes y 3 bytes de CRC. De este modo, el paquete más corto posible tiene 10 bytes y el paquete más largo posee 47 bytes. Los tiempos de transmisión de estos paquetes van desde 8 microsegundos para el paquete más pequeño hasta 300 milisegundos para los más grandes. El PDU emitido en el canal de emisión pública se compone de una cabecera de 16 bits, y en función del tipo de aviso, de una dirección de dispositivo de hasta 31 bytes de información. Además, el escáner activo puede solicitar hasta 31 bytes de información adicional del emisor si el modo anuncio permite tal operación. Esto significa que una parte considerable de datos puede ser recibida desde el dispositivo emisor incluso sin establecer una conexión. Los intervalos de emisión se pueden ajustar en un rango que va desde los 20 ms. a los 10 s. Este valor especifica el intervalo entre los paquetes de enviados consecutivamente.

El sensor, que está conectado al estado del semáforo, captura el color del semáforo, y envía de forma constantemente esta información a todos los vehículos cercanos. Para asegurar la integridad de cada paquete enviado en modo beacon, se utiliza el esquema de firma digital ligero. Concretamente se utiliza el esquema 1 del estándar de cifrado ISO/IEC 9796-2 [116] basado en SHA-1, utilizando para la firma digital el algoritmo RSA. Se ha elegido dicho algoritmo de firma debido a que la longitud de salida es de sólo 22 bytes, por lo que cumple con los requisitos de tamaño disponible por los mensajes emitidos por los dispositivos BLE. ISO/IEC 9796-2 es un esquema de firma estándar ampliamente utilizado en la industria de tarjetas inteligentes para certificados de claves públicas y para la autenticación de mensajes, ya que es bastante simple de implementar y ofrece un alto nivel de seguridad a dispositivos de baja capacidad de cómputo.

La propuesta diseñada presupone que todos los semáforos tienen un certificado genérico para firmar las balizas enviadas, generado por la Dirección General de Tráfico.

El mensaje a enviar de forma constante por los semáforos tiene un formato como el mostrado en la Figura 4.8, donde:

- idSemáforo: Identificador único de cada semáforo.
- dirección: Dirección en la que el semáforo tiene jurisprudencia dentro de un determinado cruce. Se representa con un valor en grados (de 0 a 360).
- estado: Estado del semáforo (verde, rojo o ámbar).
- firma: Firma digital del mensaje.

El mensaje se recibe por cada teléfono inteligente que esté al alcance de

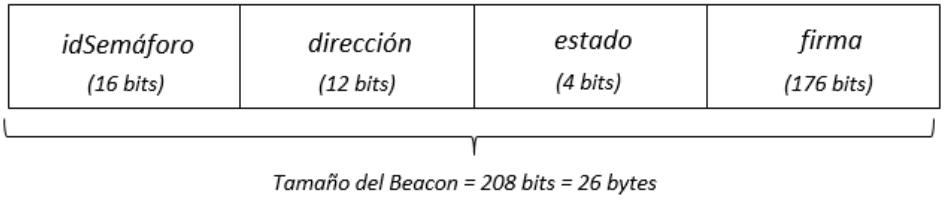

Figura 4.8: Formato del Mensaje Transmitido por los Semáforos

la comunicación BLE. Cada teléfono se encarga de procesar la información y avisar de forma anónima al servidor si infringe el semáforo.

4.2.6. Implementación

Para las pruebas se ha realizado un prototipo de funcionalidad dentro de la aplicación móvil DEPHISIT, para capturar los mensajes emitidos por los semáforos a través de Bluetooth Low Energy, y poder procesar la información. La Figura 4.9 muestra la interfaz de usuario basada en la plataforma DEPHISIT.

Figura 4.9: Interfaz de Usuario de la Aplicación DEPHISIT

El stack tecnológico utilizado en esta propuesta, adaptado al stack tecnológico de DEPHISIT, se puede ver en la Figura 4.10.

En función de los datos de la baliza, y la velocidad del vehículo, la aplicación detecta en segundo plano si el vehículo no respetó el semáforo. Si el conductor del vehículo infringió el semáforo, el dispositivo móvil envía un

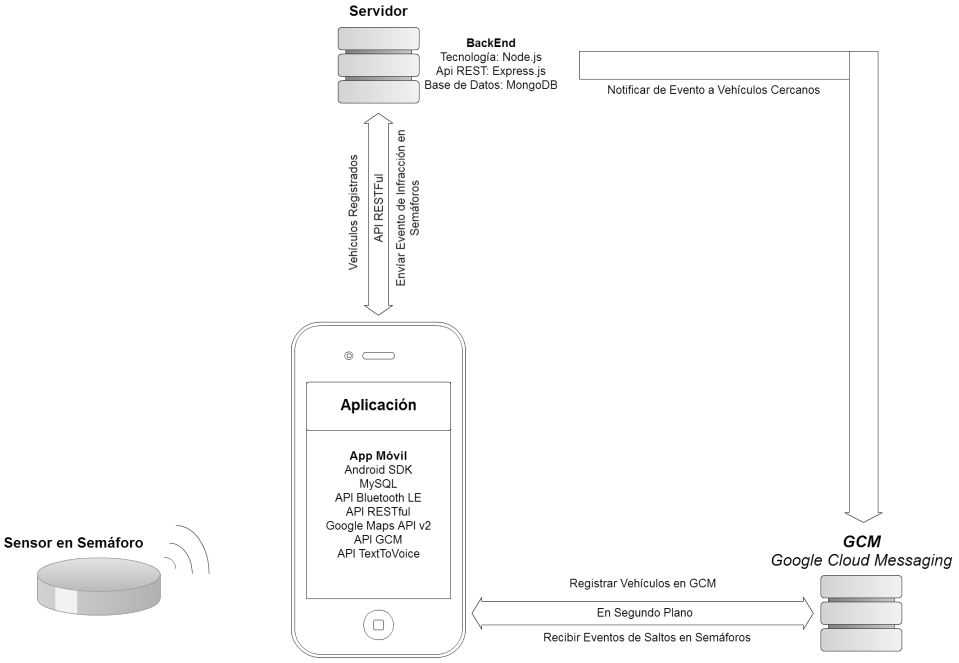

Figura 4.10: Flujo de las Tecnologías Utilizadas

mensaje al servidor que controla y maneja este tipo de eventos en DEPHISIT. El servidor se encarga de buscar en su base de datos a los vehículos cercanos en ese momento. Esto es posible debido a que todos los vehículos de la plataforma DEPHISIT envían cada 5 segundos sus posiciones al servidor. Una vez que el servidor ha localizado a todos los vehículos cercanos al semáforo, se genera una notificación que se envía a todos los teléfonos inteligentes de aquellos vehículos cercanos. Acto seguido, la aplicación informa a los conductores de esos vehículos, a través de un mensaje de voz, de la existencia de otro conductor cercano que se ha saltado un semáforo, por lo que les recomienda conducir con precaución. La aplicación también muestra en un mapa, la posición del semáforo infringido, así como la posición actual del conductor avisado para que pueda establecer el nivel de precaución deseable.

El sistema implementado utiliza varios procesos que envían varios paquetes de datos. En la Tabla 4.8 se muestra el tamaño de los diferentes paquetes de datos utilizados en el sistema propuesto.

Tipo de Paquete	**Tamaño (en bits)**
Mensaje emitido desde el Semáforo hacia los Smartphones (vía BLE)	208
Evento Generado desde el Smartphone hacia el Servidor (vía WiFi/GPRS/3G/LTE)	248
Notificación Push Generada desde el Servidor hacia los Smartphones	272

Tabla 4.8: Tamaño de los Paquetes Diseñados

Diferentes pruebas se realizaron para comprobar el tiempo requerido por cada tipo de mensaje así como el tiempo total requerido por el sistema para la detección y aviso de cada evento de infracción de semáforo. Las simulaciones se realizaron utilizando múltiples paquetes software para añadir credibilidad y realismo. El escenario utilizado para las simulaciones comprende una situación real del tráfico en la ciudad de Madrid (España) en 2014 [65]. Con el fin de simular la arquitectura y las comunicaciones de una VANET, se ha utilizado la herramienta NS-2, y para la generación del tráfico, la herramienta SUMO, la interacción entre el tráfico fue generada con SUMO y la VANET simulada con NS-2 se ha generado utilizando MOVE.

Como resultado, se han generado los tiempos medios de cada componente que se muestran en la Tabla 4.9, demostrando que el sistema es eficaz y puede ser utilizado en entornos reales.

Componente	**Tiempo Medio (en ms)**
Mensaje emitido desde el Semáforo hacia los Smartphones (vía BLE)	0,17
Procesado de los Datos en el Smartphone	51
Envío del Evento Generado desde el Smartphone hacia el Servidor (vía WiFi/GPRS/3G/LTE)	116
Procesado de los Datos en el Servidor	104
Notificación Push Generada desde el Servidor hacia los Smartphones	142
Tiempo Total	**413,17**

Tabla 4.9: Tiempos Requeridos para Enviar los Paquetes

Después de haber conseguido resultados prometedores con el sistema descrito en su fase beta, actualmente está en proceso de desarrollo una versión mejorada del sistema. La nueva versión permite notificar de la infracción de tráfico a los vehículos cercanos de forma directa, sin requerir el envío al servidor central. Esto se consigue gracias a las comunicaciones V2V que utilizan la tecnología Wi-Fi Direct. El sistema mejorado requiere un mayor nivel de seguridad de las comunicaciones por lo que cuando un evento se emite de forma automática, se garantiza no sólo la autenticidad del emisor y la integridad de los datos, sino también la frescura de los datos. Además, con el fin de promover el uso del sistema, la privacidad de los conductores se protege a través de un esquema de anonimato reversible, de manera que sólo los conductores fraudulentos que provocan accidentes, pueden ser detectados. Por último, en caso de accidente, se introducen técnicas de no repudio que son capaces de proporcionar las pruebas necesarias para demostrar la responsabilidad del infractor. Con el fin de garantizar la frescura de los datos, se añade una marca de tiempo a las firmas.

4.3. Otras Aplicaciones

Aparte de las aplicaciones descritas anteriormente, durante los últimos 3 años, se han elaborado otra serie de aplicaciones en paralelo, con diferentes

temáticas, pero siempre basadas en los algoritmos que se han desarrollado como producto de la investigación. De esa manera, las siguientes aplicaciones descritas no sólo son aplicables a entornos vehiculares, sino a entornos más generalistas que ponen a disposición de una gran masa de usuarios, aplicaciones específicas totalmente seguras. Todas las aplicaciones desarrolladas, tanto las explicadas a continuación como las anteriormente descritas, son de código libre, y su código está público en la plataforma GitHub. Además, las aplicaciones de esta sección han sido ganadoras de diferentes premios, tanto a nivel autonómico, como a nivel nacional e incluso internacional, participando junto a las mejoras aplicaciones móviles de sus sector elaboradas por las grandes multinacionales [84].

4.3.1. Shorcial

Shorcial [174] es una app desarrollada utilizando el algoritmo de confianza implementado basado en la teoría de los seis grados de separación aplicado a redes sociales descrito en la primera sección de este capítulo. La app ha sido ganadora del concurso The AppTourism Awards 2015 en FITUR 2015 [229], proclamándose como mejor app nacional en la categoría Sol y Playa. Antes, se alzó con el primer premio en el II Concurso Open Data Canarias, recibido en el año 2014 [199]. Por último, recientemente ha sido finalista del concurso internacional de app organizado por la ONU, los World Summit Awards, representando a toda España en la categoría de Cultura y Turismo [84].

La idea de Shorcial surge con el fin de presentar una herramienta útil y segura para explotar el turismo de playas en las Islas Canarias. Shorcial permite encontrar la playa ideal para cada momento, en base a la ubicación del usuario y las valoraciones y comentarios de otras personas. De esa manera, si estas valoraciones y comentarios son de personas relevantes del entorno de un determinado usuario, como amigos o familiares, la aplicación muestra la playa más idónea basándose en la información que tenga en ese momento.

El nombre de Shorcial, es un juego de las palabras inglesas [Shor]e + So[cial] (Costa + Social), queriendo dejar claro el aspecto Social de conocer las Costas de todo el mundo.

Shorcial es un proyecto con 2 dimensiones principales:

- Dimensión Social: Se busca la interacción entre la gente, que cada persona añada sus playas más cercanas o las que más le gusten, las valore e incluso mande 'mensajes en botellas'. De esta manera se logra ayudar al turismo de Canarias en particular, y la experiencia es extrapolable a cualquier parte del mundo en general donde haya playas. Cuanto más interactúen los usuarios dentro de la app, más rico será el algoritmo para mostrar la playa acorde a los gustos y conveniencias de cada usuario.

- Dimensión Abierta: Se utilizan datos abiertos del portal Open Data Canarias. Además, el código fuente es totalmente abierto para que quien quiera pueda participar en mejorar la aplicación y reutilizar los diferentes módulos en otros proyectos.

Entre las funcionalidades de la aplicación se encuentran:

- Multilenguaje: La aplicación está traducida a los 4 grandes idiomas europeos: inglés, español, alemán y francés.
- Encontrar playas cercanas a un determinado usuario en base a su geolocalización y a su conexión y confianza con otros usuarios.
- Buscar playas por nombre o cercanía a un sitio determinado.
- Subir fotos de las playas, permitiendo que cada usuario pueda incluir en la plataforma sus propias fotos.
- Ver mucha información interesante acerca de cada una de las playas, como:
  - Nombre de la playa
  - Temperatura actual en la playa
  - Localización (con un ¿cómo llegar?)
  - Ver en directo (si dispone de webcam)
  - Si tiene o no bandera azul
  - Grado de dificultad de acceso
  - Tipo de arena
  - Grado de limpieza
  - Si tiene o no hamacas
  - Si tiene o no sombrillas
  - Si tiene o no rompeolas
  - Si tiene o no chiringuitos
  - Si tiene o no duchas
  - Si tiene o no socorrista
  - Si admite o no perros
  - Si es apta para nudistas o no
- Poder comentar y valorar las playas.
- Poder lanzar 'mensajes en botellas'desde las playas. Para ello, se requiere estar presencialmente en la misma playa tanto para lanzar esos mensajes como para ver los mensajes de otros usuarios que están en esa misma playa.

- Hacer checkin en las playas.
- Añadir nuevas playas.
- Editar algunos parámetros de las playas ya existentes.
- Informar de playas incorrectas.
- Compartir en Facebook la playa en la que estás y la temperatura que hace.

En la aplicación inicial, en la fase de lanzamiento, se tenían registradas un total de 107 playas de toda Canarias. Exactamente:

- 50 Playas de Tenerife
- 20 Playas de Gran Canaria
- 10 Playas de Fuerteventura
- 10 Playas de Lanzarote
- 5 Playas de La Gomera
- 5 Playas de El Hierro
- 5 Playas de La Palma
- 1 Playa de La Graciosa
- 1 Playa de La Isla de Lobos

Algunas de estas playas cuentan con webcam, pues se tienen censadas unas 50 webcams a lo largo de las playas canarias. Dichas webcams permiten ver en tiempo real, el estado de la playa.

Estos datos posiblemente aumentarán de manera significativa en poco tiempo, aumentando con el tiempo la información con la que se dota a los usuarios. La comunidad juega un papel muy importante en esto, y entre todos se puede crear la mayor base de datos sobre playas que se haya visto nunca.

Se pueden ver algunas capturas de la aplicación en la Figura 4.11.

4.3.2. Patea La Palma

Otra de las aplicaciones desarrolladas utilizando el algoritmo inteligente de confianza presentado al comienzo de este capítulo para discernir lo que a un determinado usuario le puede interesar o no en base a sus amistades, es Patea la Palma [172]. Patea la Palma ha sido ganadora del I Concurso de Open Data La Palma en el año 2015 [173].

Figura 4.11: Interfaz de Usuario de la Aplicación Shorcial

La aplicación recomienda un sendero para realizar en La Palma, acorde a las preferencias de un determinado usuario en ese momento. Se basa tanto en las necesidades de los usuarios como en las valoraciones y comentarios de otros amigos de confianza para confeccionar los mejores senderos posibles.

La aplicación es el primer recomendador inteligente de senderos de la isla de La Palma. Permite que cualquier turista (al ser una app tanto en inglés, como en castellano y alemán), en la palma de su mano pueda tener una herramienta potente para recomendarle el mejor sendero que se adecua a sus gustos y necesidades en cada momento. La aplicación cuenta con un algoritmo inteligente de recomendación de senderos que aprende de los gustos del usuario de forma evolutiva, así como de las relaciones con sus amigos. La inteligencia artificial desarrollada selecciona una serie de senderos acorde a las preferencias descritas por el usuario. De esa manera, mediante una serie de preguntas fáciles, con una interfaz gráfica sencilla y muy visual, el usuario podrá saber qué sendero le conviene hacer. Esta recomendación también se aclimata a grupos de personas, por lo que en función de una serie de premisas (cuántas personas irán a realizar el sendero, si irán o no niños, etc.) la aplicación busca y recomienda los mejores senderos bajo estos requisitos.

Además, el recomendador desarrollado aconseja al usuario la cantidad de agua que debe llevar para la realización de dicho sendero, indicándole en un mapa la ruta del sendero completa, así como los puntos en el sendero donde hay una fuente de agua potable. El recomendador indica al usuario qué ropa llevar al sendero en cuestión en función del tiempo que se prevé

que haga el día que el usuario tiene previsto realizar el sendero. Por otra parte, el usuario puede ver toda la información necesaria y relevante del sendero (distancia, dificultad, etc.), saber como llegar a él desde su posición actual, si es circular o no, y mucha otra información que hace de la utilización de la app una experiencia imperdible para los amantes del senderismo. La valoración de los senderos es otras de las características de esta aplicación, así como la posibilidad de dejar comentarios y subir fotos, por lo que la dimensión social juega un papel fundamental. La información que se visualiza del sendero, proviene mayoritariamente de dos dataset del Portal de Open Data La Palma.

Este proyecto está desarrollado de forma modular, de manera que se permite la inclusión de nuevos datos provenientes de otras fuentes de forma rápida y sencilla.

Las tecnologías usadas en el proyecto se dividen en parte cliente y parte servidor.

- La parte cliente (aplicación Android nativa): Para el cliente, la app, que es el artefacto final que maneja el usuario, la tecnología usada ha sido el SDK Nativo de Android, que utiliza como lenguaje de programación Java 6. Para el desarrollo de la aplicación se ha usado el IDE Android Studio que utiliza como motor de construcción de la aplicación a Gradle. Los datos locales son almacenados por la app usando MySQL en Android. El motor de mapas usados en la app es Open Street Map, ya que la aplicación busca ser lo más Open Source posible, y puede funcionar con el tiempo de forma offline y gratuita. Para loguear a usuarios, y no tener que manejar datos confidenciales, se hace uso del SDK de Facebook para Android. Para el alojamiento de las fotos, se utiliza la API de IMGUR. Para conocer las condiciones meteorológicas de un determinado lugar en tiempo real se hace uso de la API de OpenWeather.

- Parte servidor (servidor node): Para la parte del servidor, que contiene toda la información centralizada de usuarios, senderos y demás datos relevantes que se usan en la app móvil, se ha usado un fullstack de JavaScript. Concretamente se ha usado como framework del backend Node.js. Se han implementado Web Services de tipo RESTful para la comunicación entre la app y el servidor. Para ello se ha utilizado como framework Express.js. Como base de datos del lado del servidor, y para que el proyecto fuera escalable, hemos usado MongoDB, una base de datos no relacional. El servidor está alojado en una MicroInstacia de Amazon, bajo una cuenta de Amazon EC2.

Se pueden ver algunas capturas de la aplicación en la Figura 4.12.

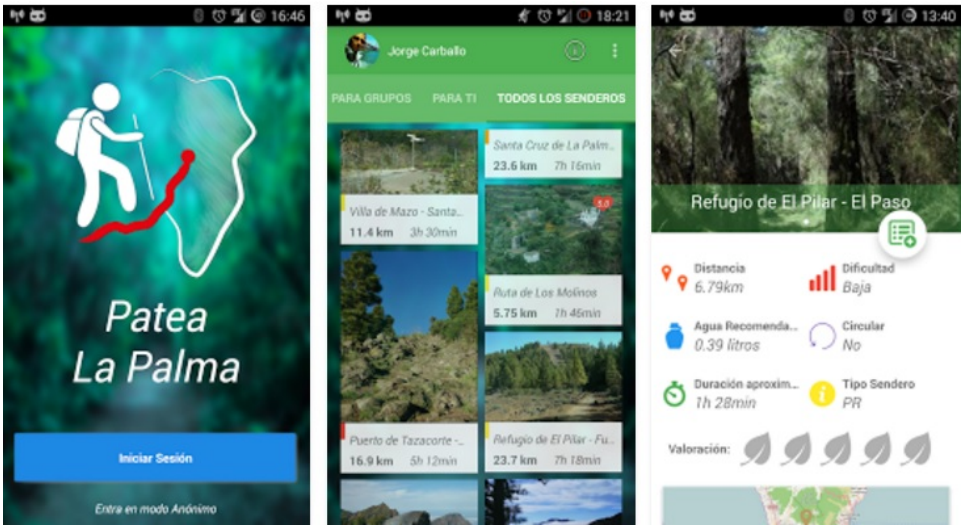

Figura 4.12: Interfaz de Usuario de la Aplicación Patea La Palma

4.3.3. Qdemos

Qdemos [149] es una aplicación para realizar quedadas entre amigos, y poder cuadrar la mejor fecha posible para que todos los involucrados puedan asistir. La aplicación ha sido la ganadora en la categoría senior, para alumnado de master y doctorandos, en el VIII Concurso Universitario de Software Libre a nivel nacional [149], además de alzarse con el 2º Premio absoluto en la fase local de la Universidad de La Laguna.

Qdemos surgió (como casi todas las ideas) de una necesidad que cubrir. Muchas personas se ven, frecuentemente, en la tediosa tarea de organizar eventos pequeños con amigos (cumpleaños, fiestas, meriendas, cenas, comidas varias, etc.), y lo más complejo de esto es poner de acuerdo a todo el mundo involucrado para decidir el día y la hora. Esto suele conllevar cientos de mensajes por Whatsapp, Facebook o cualquier otra red social. Existen aplicaciones web que permiten realizar estas gestiones (aunque no del todo fluidas y con una interfaz clara y limpia) y otras muchas apps móviles que están diseñadas para la creación de eventos. La necesidad que se observó es que todas ellas involucran un condicionante que muchas veces no es suficientemente restrictivo como para organizar un evento entre amigos. Ese no es otro que imponer una fecha y hora exacta con la que todos deben estar de acuerdo. Qdemos pretende ser una app móvil, concretamente bajo la plataforma Android que es de software libre bajo el proyecto AOSP (Android Open Source Project), que minimice toda esa tediosa organización de eventos pequeños entre amigos que hacen desesperar a más de uno cada vez que mira su Whatsapp. De esa manera, con una interfaz limpia y sencilla permita crear un evento, con varias posibles fechas, e invitar a los amigos

para que entre todos, y a través de la simplicidad de la app, se elija la fecha más acorde en función de las preferencias de cada cual. Todo esto mediante el uso de redes sociales para recabar la información necesaria de qué amigos interesa invitar a qué evento.

La estructura de Qdemos se puede ver en la Figura 4.13.

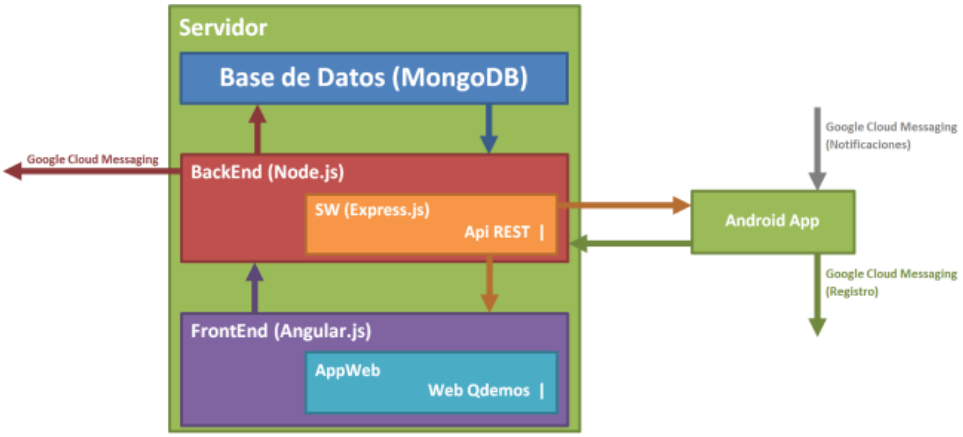

Figura 4.13: Estructura de Qdemos

Se pueden ver algunas capturas de la aplicación en la Figura 4.14.

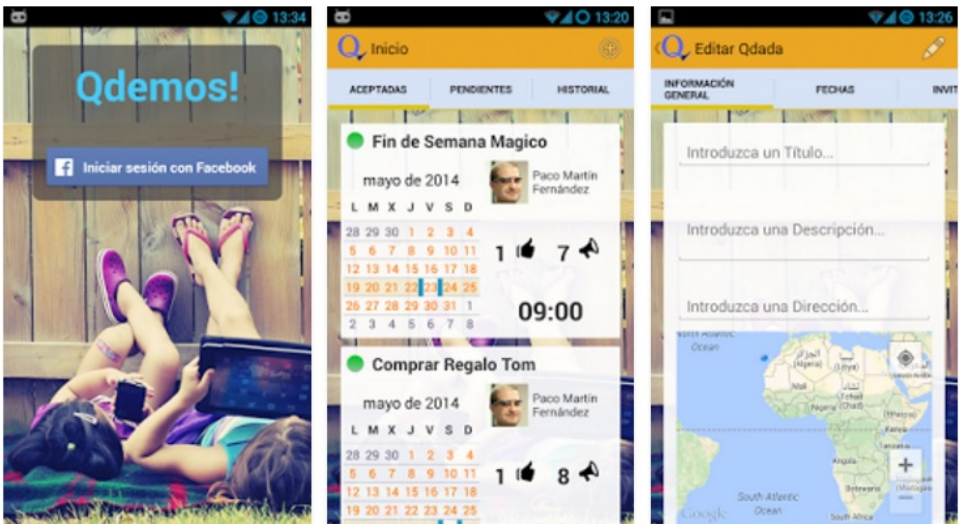

Figura 4.14: Interfaz de Usuario de la Aplicación Qdemos

4.3.4. Chascar en Tenerife

Otra aplicación que se basa en el algoritmo de confianza propuesto y descrito en secciones anteriores, pretende encontrar el mejor restaurante o sitio

para comer de la isla de Tenerife, en base a la localización de un determinado usuario y las valoraciones de otros usuarios.

Chascar en Tenerife [147] es una aplicación para conocer en todo instante los establecimientos de restauración cercanos a nuestra localización, o los establecimientos de un determinado municipio de la isla de Tenerife. Actualmente hay casi 5500 establecimientos geolocalizados en la base de datos del sistema, de un total de 11051 censados por el Cabildo de Tenerife. La aplicación Chascar en Tenerife fue la ganadora del I Concurso de Open Data Canarias, que se enmarcó en un proyecto financiado por el Área de Educación, Juventud e Igualdad del Excmo. Cabildo Insular de Tenerife.

El desarrollo de la aplicación tiene como clara motivación el ayudar a fomentar a la reutilización de los datos que pone en disposición el portal Open Data Canarias (ODC), además de servir como plataforma a los estudiantes para que encuentren lugares donde alimentarse a lo largo del curso. La Universidad de La Laguna tiene gran diversidad de alumnado que provienen de diferentes puntos geográficos. Muchos de ellos, los primeros años no conocen el entorno y siempre van a comer a los mismos sitios. Esta aplicación tiene como motivación el ayudar a los estudiantes (y a cualquier persona) que se mueva por la isla, a localizar los establecimientos de restauración censados en el Cabildo de Tenerife y recogidos en el fichero open data correspondiente que pone a disposición la plataforma ODC antes mencionada, de forma inteligente en base a las valoraciones de otros usuarios afines.

Las funcionalidades del servidor son (de momento el servidor sólo es utilizado por la aplicación Android):

- Algoritmo de procesado de datos a partir del fichero CSV en crudo (con los datos no normalizados) de los establecimientos de restauración de la isla de Tenerife, que comparte el portal Open Data Canarias. Existen establecimientos entre esos datos, que lamentablemente ya han echado el cierre, o sea, existen datos desactualizados. El algoritmo aprende para ser capaz de detectar estas situaciones.

- Algoritmo de cálculo de coordenadas geográficas de direcciones. Simplemente sabiendo el nombre de la calle, el número y el municipio, el servidor es capaz de conseguir la latitud y longitud de esa dirección.

- Servicios web de consulta de los datos una vez procesados. De esa manera, en un futuro puede haber una API totalmente documentada para que usuarios externos puedan consultar los datos de los establecimientos, mejorados por el algoritmo anterior, y hacer uso libremente de estos. Actualmente existe la API con servicios web implementados, pero que sólo son accesibles de forma segura a la Aplicación Android, debido a que se utiliza un servidor gratuito restringido en uso.

- La aplicación móvil permite buscar los establecimientos de restaura-

ción cercanos a un determinado punto. Ese punto puede ser nuestra ubicación actual u otra dirección que rellenemos ayudándonos de un motor de autocompletado que tiene implementado la aplicación. También se puede interactuar completamente con el Mapa 3D con una serie de gestos sobre él para mover la posición que queremos utilizar para calcular los establecimientos cercanos.

- Se permite buscar los establecimientos de restauración de un determinado municipio. Se puede ver su geolocalización en un mapa con funcionalidades 3D y calcular la ruta más cercana en modo navegador (a pie, en vehículo propio o en transporte público) desde la posición en ese momento hasta el establecimiento en cuestión. Se indica cómo llegar a ese establecimiento (dirección exacta paso a paso que hay que seguir), qué líneas concretas de transporte público (si procede) coger, a qué hora, etc.

Se pueden ver algunas capturas de la aplicación en la Figura 4.15.

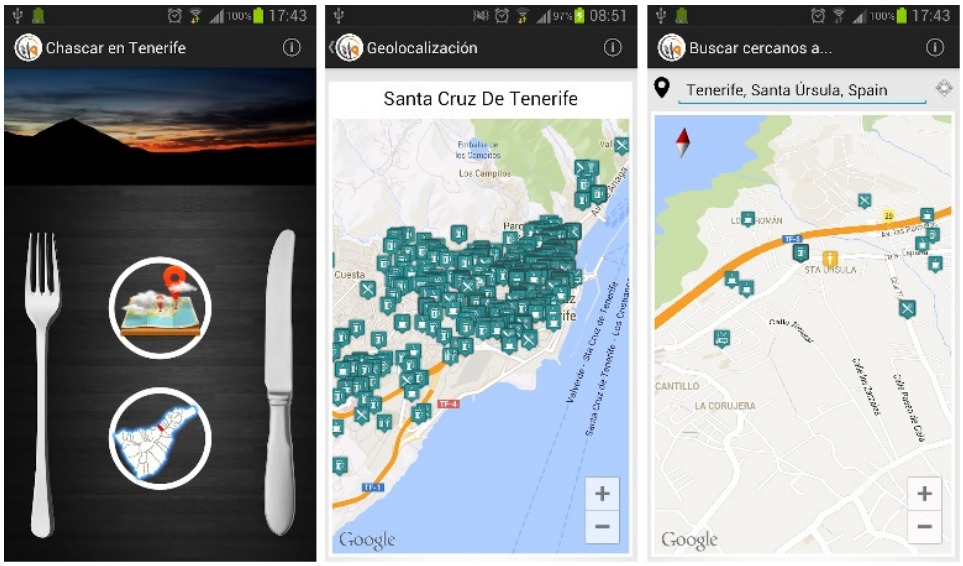

Figura 4.15: Interfaz de Usuario de la Aplicación Chascar en Tenerife

4.3.5. Otras Aplicaciones

Para finalizar, el autor de la presente Tesis ha realizado otras muchas aplicaciones en colaboración con alumnos del grado de ingeniería informática mediante la cotutorización de sus trabajos fin de grado. Una de ellas [3] permite realizar el seguimiento de personal aeroportuario para la asignación eficiente y automática de tareas de mantenimiento [4], mediante la tecnología Bluetooth Low Energy. Cabe señalar que este sistema no sólo puede

ser utilizado para asignar tareas según la localización de los empleados, sino que también tiene en cuenta otros factores como el nivel de ocupación o la categoría de la tarea. Por ello, el sistema cuenta con diferentes algoritmos criptográficos que permiten gestionar de forma segura la privacidad de los usuarios y de la entidad aeroportuaria. Además, se ha optimizado la eficiencia de las comunicaciones [5] para que los datos puedan ser generados y consumidos en tiempo real.

Otro de los trabajos fin de grado codirigidos permite localizar a personas que se han perdido en entornos rurales mediante la combinación de sensores y teléfonos móviles usando la tecnología BLE [130]. El sistema es capaz de saber si a 100 metros a la redonda hay un baliza BLE radiando un mensaje de forma constante. De esa manera, persona está asociada a una baliza que lleva siempre encima. En el momento en que un equipo de salvamento sale en su búsqueda, va buscando la señal de la baliza para confirmar la posición de la persona. Este mecanismo se podría automatizar con el uso de drones, de forma que vayan peinando áreas de forma inteligente.

Por último, otro trabajo fin de grado co-tutorizado hace uso de relojes inteligentes con sistema operativo Android Wear para monitorizar de forma inteligente el ritmo cardíaco de las personas, en función de su calendario [139]. Con ello, se puede detectar como varía las pulsaciones de una persona en función de la importancia o relevancia de las reuniones y citas que tenga.

Capítulo 5

Conclusiones

Esta Tesis doctoral está orientada a la propuesta e implementación de soluciones seguras e innovadoras para situaciones reales en escenarios de transporte. En el primer capítulo se presentan los fundamentos y conceptos básicos tratados en el resto de la Tesis, a modo de introducción. Esta parte adentra al lector en la Internet de las Cosas, y explica de forma razonada por qué es necesario una aproximación en materia de seguridad distinta de la conocida hasta ahora. Se introducen allí algunos conceptos básicos de seguridad así como la evolución de la seguridad clásica hacia la Internet de las Cosas. Con esos conceptos sencillos y concisos se abordan y detallan algunas pautas de los nuevos esquemas de seguridad en este paradigma reciente. Además, se introduce el concepto de redes vehiculares, como un tipo de red que surge en la Internet de las Cosas. De esta manera se presenta el foco principal de las soluciones criptográficas descritas en el resto de capítulos. Se presta especial atención a las comunicaciones inalámbricas entre los nodos de este tipo de redes para destacar la necesidad de que aparezcan nuevos algoritmos seguros en estas tecnologías. A continuación, en el resto de capítulos se explican y detallan las soluciones propuestas tras las investigaciones realizadas durante casi cuatro años. Para cada una de las propuestas se realiza un profundo análisis de los diseños y desarrollos llevados a cabo, así como la comparación con otras propuestas análogas que ponen en liza los resultados obtenidos. Finalmente, el presente capítulo cierra esta memoria de Tesis con las conclusiones más relevantes que han dejado las investigaciones, detallando una serie de trabajos futuros que se han quedado abiertos como producto de las investigaciones llevadas a cabo.

5.1. Contribuciones

En el capítulo 2 de la presente Tesis se describe un nuevo método de autenticación basado en las demostraciones de conocimiento nulas no interactivas. Dicho método ha sido diseñado para ser utilizado en entornos donde

los dispositivos se mueven de forma constante y a grandes velocidades, como los escenarios de transporte. La implementación y desarrollo del esquema propuesto se detallan a lo largo del capítulo, junto a los resultados obtenidos y las comparaciones con métodos similares. Gracias a ese esquema, se ha conseguido definir un nuevo método para compartir información de forma autenticada sin requerir un intercambio interactivo de mensajes en redes inalámbricas descentralizadas. Por ello, el protocolo creado tiene una gran aplicabilidad en entornos vehiculares, donde los nodos de la red se mueven a grandes velocidades en diferentes direcciones, teniendo una ventana corta de tiempo para comunicarse con el resto de nodos.

El capítulo 3 aporta nuevas soluciones para la gestión de usuarios revocados en las redes vehiculares. Se ha utilizado como base de los nuevos esquemas de revocación, las estructuras de datos autenticadas en forma de árboles hash. Para ello se propone un esquema de gestión de certificados revocados usando árboles k-arios, con parámetros totalmente configurables para adaptar la estructura a diferentes entornos vehiculares, desde grandes núcleos urbanos, hasta zonas rurales en la periferia. En particular, se ha propuesto una variante basada en la inclusión en el esquema básico de la idea de los códigos de Huffman. Esta nueva estructura de gestión de certificados y/o seudónimos revocados tiene la principal misión de poder modelizar el árbol en función de los tipos de vehículos revocados. Analizando el comportamiento de los vehículos en su día a día, se ha concluido que existen determinados vehículos que pasan más tiempo en las carreteras, por lo que son más propensos a comunicarse con el resto de vehículos. De esta manera, el árbol se ha adaptado para que la prueba de revocación sea mucho más optimizada en tamaño y tiempo de procesado para los vehículos más consultados. Gracias a un árbol cuyos nodos hoja están a diferentes profundidades, se ha diseñado la propuesta según la cual los vehículos que más transitan las carreteras, como pueden ser los del transporte público y taxis o los vehículos comerciales, se corresponden con nodos hoja a menor profundidad. Estas nuevas propuestas son más eficientes que las soluciones clásicas tradicionales, ya que decrementan tanto el tamaño de las pruebas de revocación como los tiempos de generación de las estructuras y comprobación de revocaciones.

En el capítulo 4 se presenta una serie de aplicaciones móviles que ponen en liza varias investigaciones producidas durante el transcurso del doctorado. En él se describe un nuevo algoritmo de confiabilidad que permite calcular la fiabilidad entre dos usuarios concretos de forma unidireccional y dinámica. Gracias a la teoría de los seis grados de separación y las comunicaciones y relaciones de las personas en las redes sociales, se ha diseñado un esquema que puede ser utilizado por cualquier tipo de aplicación social para darle un valor añadido a los usuarios, en cuanto a seguridad y confianza. En base a este algoritmo, se han diseñado una serie de aplicaciones que lo utilizan para demostrar sus eficacia y rendimiento. Se ha implementado una aplica-

ción para compartir vehículo, un entorno donde la mayoría de usuarios no poseen ese grado de confianza necesario para fiarse de otro usuario a la hora de compartir coche. Para dicha aplicación se han detallado las pruebas de campo realizadas, lanzando la aplicación al público en general, y obteniendo unas métricas realmente buenas. En este capítulo también incluye otro tipo de aplicaciones de diversa índole, realizadas para enfatizar que las investigaciones realizadas en la teoría son totalmente aplicables en la práctica. Por ejemplo, el capítulo incluye la descripción de una aplicación móvil propuesta para detectar y denunciar de forma anónima las infracciones de los conductores en los semáforos, de manera, que permite notificar a los usuarios de las zonas colindantes del peligro existente y generar métricas que sirvan para analizar los semáforos que son más propensos a ser infringidos, para actuar en consecuencia. El capítulo se completa con una serie de aplicaciones que tienen como base el algoritmo de confianza descrito en la primera sección para diversos objetivos como permitir hacer quedadas con amigos de forma inteligente, o buscar la playa ideal, el mejor restaurante o el mejor sendero para cada usuario en cada momento.

5.2. Trabajos Futuros

Las investigaciones realizadas han ido abriendo nuevos retos a los que enfrentarse. De esta manera, no sólo se han conseguido nuevas propuestas para afianzar la seguridad en entornos tan críticos como los vehiculares, sino que ha servido para detectar y descubrir nuevos trabajos futuros que se podrían llevar a cabo en estos escenarios.

Desde el punto de vista de la autenticación, las investigaciones llevadas a cabo en ZKP no interactivas han dejado varios problemas abiertos. Por un lado se hace necesario diseñar algún tipo de sistema de compresión de datos para que el tamaño requerido a la hora de almacenar los grafos a utilizar, sea mucho menor para grafos de más de 100 nodos. Otra de las investigaciones futuras a realizar en esa temática tiene que ver con la reducción del proceso de descifrado de los mensajes. Aunque el proceso de generación de los paquetes es óptimo, su descomposición posterior de ir descifrando retos por cada segmento se vuelve costoso para grafos de más de 100 nodos. Sería por tanto aconsejable mejorar estas funciones mediante la utilización de técnicas de procesamiento en paralelo y optimización de recursos. Por último, se podrían encontrar nuevos problemas matemáticos complejos para basar el sistema.

Por el lado de los esquemas de revocaciones de certificados y seudónimos, es necesario que puedan ser aplicados a sistemas vehiculares, y no sólo a dispositivos móviles. Las pruebas que se han realizado presuponen que las unidades de los vehículos y las infraestructuras auxiliares sean reemplazadas por teléfonos inteligentes. Si se pudiera implementar el esquema en VANETs

según la definición convencional, se podría adoptar en grandes proyectos gubernamentales de despliegue de redes vehiculares, para comprobar su eficacia en pruebas de campo en tiempo real. Otros de los trabajos abiertos es la necesidad de diseñar un nuevo método más seguro y eficiente para verificar que un determinado vehículo no ha sido revocado. Los esquemas propuestos maximizan y aseguran la prueba de revocación de un determinado vehículo, pero el sistema no posee un método optimizado para verificar que un vehículo no ha sido revocado. Los árboles de Huffman permiten segmentar el conjunto de los vehículos revocados por el tiempo que frecuentan las carreteras, otorgando una gran mejora frente a las soluciones tradicionales. Sería conveniente seguir explotando esta idea, y poder segmentar el conjunto de los vehículos por otro tipo de comportamientos, como por ejemplo por el número de conexiones realizadas con otros vehículos, o el tiempo real de rodaje en centros urbanos.

Por último, el algoritmo de confianza propuesto en el último capítulo deja la posibilidad abierta de que sea implantado en soluciones de terceros para ver si mejora las métricas de confianza de dichas soluciones. Sería bueno que una aplicación relevante ya existente y con una gran masa crítica de usuarios pudiera adoptar la propuesta y hacer tests para parametrizar si los usuarios se sienten más seguros utilizando el algoritmo diseñado. Además, el algoritmo se debe nutrir de nuevas redes sociales como Snapchat o Twitch. Ahora está implementado con Facebook, y preparado para poder admitir datos de Twitter y Google Plus. Otra mejora posible sería la inclusión de nuevos parámetros a considerar en el núcleo del algoritmo. Ahora se monitoriza el texto de los comentarios y las interacciones con 'me gusta'. Podría ser interesante que el algoritmo se nutriese de las fotos y sus etiquetas para detectar estados de ánimo que puedan añadir más datos cuantitativos acerca de la relación entre usuarios. Por parte del esquema de auto-denuncia anónima en infracciones en semáforos, sería aconsejable poder predecir la infracción del semáforo para intentar avisar con mayor antelación a los usuarios cercanos. Con técnicas de machine learning, mediante deep learning y big data, se podría estimar con cierto grado de probabilidad en base a comportamientos anteriores, y velocidades y distancias actuales, si un determinado vehículo se va a saltar un semáforo o no. De esta manera, los usuarios circundantes podrían actuar en consecuencia antes de que la infracción ocurriese, minimizando así la probabilidad de siniestro.

Apéndice A

Publicaciones

Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo

Benjamin Franklin

En este apéndice se listan todos los artículos publicados en revistas así como las presentaciones en congresos nacionales e internacionales, producto de las investigaciones realizadas durante el desarrollo de la Tesis en los últimos casi cuatro años.

Revistas Indexadas ISI JCR [1]

1. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Authentication Based on Non-Interactive Zero-Knowledge Proofs for the Internet of Things. Sensors, 16, 1. http://www.mdpi.com/1424-8220/16/1/75, 2016.

    - Impact Factor: **2.245**
    - Category:
        - Electrical and Electronic Engineering: **Q1**
        - Instrumentation: **Q1**

Revistas Internacionales [2]

1. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Revocation for Certificateless Authentication in VANETs. International Journal of Intelligent Computing Research. ISSN: 2042 4655, Vol. 5, Issue 3-4. 2014.

2. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Tree-Based Revocation for Certificateless Authentication in Vehicular Ad-Hoc Networks. Journal of Computer and Communications. Vol. 2, pp. 14-21. 2014.

Conferencias Indexadas CORE [11]

1. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. An Experimental Hybrid Wireless Platform for Vehicular Networks. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. Coimbra, Portugal. 2016.
   - CORE Score: **A**
2. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Revocation in VANETs Based on k-ary Huffman Trees. Workshop on experiences with the design and implementation of smart objects held in conjuction of MobiCom, pp. 25-26. París, Francia. 2015.
   - CORE Score: **A***
3. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. A Trustworthy Distributed Social Carpool Method. The Third Workshop on Large Scale Distributed Virtual Environments, held in conjunction of Euro-Par. Vol. 9523, pp. 324-335. Viena, Austria. 2015.
   - CORE Score: **A**
4. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Revocation Management in Vehicular Ad-hoc Networks. The 8th IEEE International Symposium on Security, Privacy and Anonymity in Internet of Things held in conjunction with IEEE TrustCom. Vol. 1, pp. 1210-1217. Helsinki, Finlandia. 2015.
   - CORE Score: **A**
5. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Lightweight Non-Interactive Authentication and Confidential Information Exchange for Mobile Enviroments. International Joint Conference, Advances in Intelligent Systems and Computing. Vol. 369, pp. 261-272. Burgos, España. 2015.
   - CORE Score: **B**
6. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Efficient Management of Revoked Pseudonyms in VANETs using ID-Based Cryptography. XII International Workshop on Security In Information Systems, pp. 701-708. Barcelona, España. 2015.

- CORE Score: **C**

7. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Tree-Based Revocation for Certificateless Authentication in Vehicular Ad-hoc Networks. International Conference on Computational Intelligence and Software Engineering, pp. 14-21. Beijing, China. 2014.

   - CORE Score: **C**

8. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Non-Interactive Authentication Scheme for Light Environments. International Conference on Information Technology and Applications. Sydney, Australia. 2014.

   - CORE Score: **C**

9. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil. Increasing Privacy and Trust in Cooperative Social Platforms for Vehicular Applications. XI International Workshop on Security In Information Systems. Vol. 1, pp. 3-13. Lisboa, Portugal. 2014.

   - CORE Score: **C**

10. F. Martín-Fernández, P. Caballero-Gil. Use of a Duplex Construction of SHA-3 for Certificate Revocation in VANETs. X International Workshop on Security In Information Systems. Vol. 1, pp. 3-11. Angers, Francia. 2013.

    - CORE Score: **C**

11. F. Martín-Fernández, P. Caballero-Gil. Version of the New SHA Standard Applied to Manage Certificate Revocation in VANETs. International Work Conference on Artificial Neuronal Networks. LNCS 7902, pp. 161-168. El Puerto de la Cruz. 2013.

    - CORE Score: **B**

Conferencias Internacionales Lecture Notes in Computer Science series [2]

1. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Managing Certificate Revocation in VANETs Using Hash Trees and Query Frequencies. XV International Conference On Computer Aided Systems Theory. LNCS 9520, pp. 57-63. Las Palmas de Gran Canaria, España. 2015.

2. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Analysis of the New Standard Hash Function. XIV International Conference On Computer Aided Systems Theory. LNCS 8111, pp. 142-149. Las Palmas de Gran Canaria, España. 2013.

Conferencias Internacionales IEEE [1]

1. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Using query frequencies in tree-based revocation for certificateless authentication in VANETs. 9th International Conference for Internet Technology and Secured Transactions. IEEE, pp. 268-273. Londres, Reino Unido. 2014.

Otras Conferencias Internacionales [10]

1. N. Álvarez-Díaz, P. Caballero-Gil, H. Reboso-Morales, F. Martín-Fernández. Optimizing Resource Allocation and Indoor Location Using Bluetooth Low Energy. 18th International Conference on Air Transport Management. Londres, Reino Unido. 2016.

2. F. Martín-Fernández. Ad-hoc Networks for Disasters. First Workshop on Mobile Tools for Emergencies and Critical Infrastructures. San Cristobal de La Laguna, España. 2015.

3. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Detection and Report of Traffic Lights Violation using Sensors and Smartphones. 9th International Conference on Ubiquitous Computing and Ambient Intelligence. Vol. 9454, pp. 237-248. Puerto Varas, Chile. 2015.

4. N. Álvarez-Díaz, P. Caballero-Gil, F. Martín-Fernández. Task Assignment Through Indoor Location with Bluetooth Low Energy Devices. 4th International Conference on Theory and Practice in Modern Computing. Las Palmas de Gran Canaria, España. 2015.

5. P. Caballero-Gil Pino, F. Martín-Fernández, C. Caballero-Gil. Ubiquitous Computing Technologies to Manage a Transport Monitoring System. IEICE Information and Communication Technology Forum 2015. Manchester, Reino Unido. 2015.

6. P. Caballero-Gil, F. Martín-Fernández, C. Caballero-Gil. Cooperative Social System based on Trust for Carpooling. IEICE Information and Communication Technology Forum 2015. Manchester, Reino Unido. 2015.

7. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Tree-Based Revocation for Authentication in Vehicular Ad-hoc Networks. 14th International Conference on Computer Systems and Technologies, pp. 71-78. Ruse, Bulgaria. 2014.

8. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Tree-Based Management of Revoked Certificates in Vehicular Ad-hoc Networks. World Congress on Engineering. Vol. II, pp. 1425-1431. Londres, Reino Unido. 2013.

9. F. Martín-Fernández. Some Uses of the New Standard Hash Function. II Workshop on Security in Internet of Things. San Cristobal de La Laguna, España. 2013.

10. C. Caballero-Gil, J. Molina-Gil, P. Caballero-Gil, F. Martín-Fernández, D. Yanes-García. VAiPho. Introducing Secure and Self-Organized Vehicular Ad-Hoc Networks. Proceedings CompSysTech: 12th International Conference on Computer Systems and Technologies. Viena, Austria. 2011.

Conferencias Nacionales [13]

1. F. Martín-Fernández, M. Pérez-García, R. Aguasca-Colomo, P. Caballero-Gil, E. Suarez-Curbelo. Los UAVs como relays de comunicaciones. Usos en emergencias. Madrid Drone. Madrid. 2016.

2. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. ¿Estamos preparados para la Internet de las Cosas?. Reunión Jóvenes Investigadores del Instituto de Desarrollo Regional de la Universidad de La Laguna. San Cristobal de La Laguna. 2015.

3. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Esquemas Criptográficos en Redes Móviles Autogestionadas. Congreso de Jóvenes Investigadores de la Real Sociedad Matemática Española. Murcia. 2015.

4. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Autenticación No Interactiva para Internet de las Cosas. I Conferencia de Jóvenes Investigadores en las Islas Canarias. San Cristobal de La Laguna. 2015.

5. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Algoritmos Criptográficos y Aplicaciones Seguras para Escenarios de Transporte. CyberCamp 2014, I PhD WorkShop en Seguridad. Madrid. 2014.

6. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil. Autenticación No Interactiva para Internet de las Cosas. XIII Reunión Española sobre Criptología y Seguridad Informática. Alicante. 2014.

7. F. Martín-Fernández, C. Caballero-Gil, J. Molina-Gil, P. Caballero-Gil. Plataforma móvil segura y confiable para carpooling. X Foro de Innovaciones tecnológicas para el transporte. Las Palmas de Gran Canaria. 2013.

8. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil. Conexión segura entre dispositivos móviles para la asistencia a la conducción. XII Reunión Española sobre Criptología y Seguridad Informática. San Sebastián. 2012.

9. F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil. Aplicaciones Seguras en Internet de las Cosas para Transporte. IX Foro de Innovaciones tecnológicas para el transporte. Puerto del Rosario. 2012.

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

11. F. Martín-Fernández. Riesgos en Smartphones e Internet de las Cosas. Primer Workshop en Seguridad en Internet de las Cosas. San Cristobal de La Laguna. 2012.

12. F. Martín-Fernández, D. Yanes-García, P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil. Detecta atascos de tráfico y aparcamientos libres en tu smartphone. Salón atlántico de logística y transporte. Las Palmas de Gran Canaria. 2011.

13. P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, D. Yanes-García, F. Martín-Fernández. VAiPho. Una herramienta para la asistencia en la conducción. VIII Foro de Innovaciones tecnológicas para el transporte. Las Palmas de Gran Canaria. 2011.

Trabajos Fin de Carrera Co-dirigidos [4]

1. P. Caballero-Gil Pino F. Martín-Fernández (Directores) Trabajo Fin de Grado en Ingeniería en Informática dirigido: Sistema Inteligente de Detección y Aviso de Infracciones en Semáforos Mediante Smartphones. Proyectando: R. Lakhani Lakhani. Universidad de La Laguna. ETSI Informática. 23 julio de 2015. Sobresaliente (9).

2. C. Hernández-Goya, F. Martín-Fernández (Directores) Trabajo Fin de Grado en Ingeniería en Informática dirigido: Localización de personas en entornos rurales mediante la combinación de sensores y teléfonos móviles usando tecnología BLE. Proyectando: S.J. Kapai Harpalani.

Universidad de La Laguna. ETSI de Informática. 22 julio de 2015 Sobresaliente (10) (por unanimidad).

3. J.L. Roda-García, F. Martín-Fernández (Directores) Trabajo Fin de Grado en Ingeniería en Informática dirigido: Sistema de decisión inteligente para la toma de pulsaciones cardíacas usando Android Wear. Proyectando: M. Lodeiro Santiago. Universidad de La Laguna. ETSI de Informática. 22 junio de 2015 Sobresaliente (10) (por unanimidad).

4. P. Caballero-Gil, F. Martín-Fernández (Directores) Trabajo Fin de Grado en Ingeniería en Informática dirigido: Localización interior y asignación de tareas con dispositivos Bluetooth Low Energy. Proyectando: N. Álvarez Díaz. Universidad de La Laguna. ETSI de Informática. 19 junio de 2015 Sobresaliente (10) (por unanimidad).

Apéndice B

English Overview

This is my destiny! Don't tell me what I can't do!

John Locke

This appendix provides an explanation, in English, of the main contents of the Thesis.

B.1. Abstract

The Internet of Things (IoT) is a trending concept that arises from the need to monitor and interconnect billions of information devices, typically equipped with sensors, actuators, microprocessors, communication interfaces and/or power sources. New challenges related to wireless security appear in IoT because the most common medium used for communication among hyper-connected devices is wireless. Thus, since most IoT objects have limited processing power, finite memory and low battery life, new lightweight cryptographic algorithms, in terms of processing and memory requirements, are necessary to obtain the efficiency of end-to-end communication and applicability to low-resource devices. With the advent of increasingly powerful technology that is reduced in size and weight, the security level of the cryptographic schemes that are used in wireless communications has been increasing due to the rise of new algorithms based on harder mathematical problems, such as elliptic curves. In particular, the evolution of networks towards IoT has implied that this process is now accelerated.

The hypothesis, and main motivation of this work, has been the consideration that in the new paradigm of the Internet of Things the current security does not fit to the needs of such networks. Thus, this Thesis focuses on one of the wireless networks that are having more impact on research in recent years: Vehicular Ad-hoc NETworks or VANETs. The main lines followed in

the research described in this memory have been to design new cryptographic lightweight algorithms to provide reliability and confidentiality to legitimate users and ensure an efficient authentication through managing the malicious users.

Therefore, this Thesis focuses on the conception, design and implementation of new cryptographic algorithms for secure applications in transport scenarios in order to solve several current problems in this field. Thus, among the main objectives of this work are the analysis of the state of the art on some relevant cryptographic algorithms for Internet of Things; the study of the most significant applications of the transport area; the proposal of new cryptographic algorithms; the adaptation of this proposals on wireless networks, the implementation of the algorithms designed on Android Open Source Project platform; the integration of the solutions in heterogeneous networks of mobile devices with different technologies; the security evaluation of the proposals against malicious attacks; the comparison of the obtained results with other relevant systems; and the development of the best solutions proposed in mobile applications to be accessible in real environments.

The contributions of this Thesis are aligned with a deep analysis and improvement of existing cryptographic algorithms for mobile and volatile environments generated with the emergence of the Internet of Things, with particular emphasis on the applicability to transport scenarios. Contributions can be briefly summarized in the following achievements: a detailed review of the state of the art security in wireless communications, a elaborated study of the most relevant authentication and revocation methods for users in vehicular environments, a authentication method based on non-interactive zero knowledge proofs, a revocation systems based on authenticated data structures and hash trees, an algorithm to obtain a quantitative reliability measure, between users, based on social networks and the theory of six degrees of separation, and a self-reporting anonymously system for traffic lights violations.

This appendix introduces a new authentication method based on non-interactive zero knowledge proofs. This method has been designed for transport environments. Also, in this section a new solution for managing the misbehaving users in vehicular ad-hoc networks is provided. Using authenticated data structures and hash trees, different schemes based on k-ary trees or Huffman codes are proposed to manage the revoked users. In addition, it is introduced some mobile applications designed and developed using previous research. Finally, this appendix ends with a summary of the most important contributions learned from research, and future works open.

This thesis has been developed under the national research project called TUERI [245] thanks to the FPI BES-2012-051817 grant associated to it, and in connection with several research projects funded by Ministerio de Ecnonomía y Competitividad: DEPHISIT [114], ATLAS [222] and CASUS

[242].

B.2. Authentication in VANETs

Due to the emergence of the new paradigm of interconnected objects, where the physical dimension mimics the logical dimension, it is necessary to encode more than 100,000 million objects, because each person is surrounded by approximately 3000 objects. As previously mentioned, a key aspect to consider is the way of communication between these objects. Thus, the mobile nature and small size of many of these devices that form the IoT usually imply that such a communication is wireless and that the resulting network is established as a mobile ad-hoc network (MANET), which is a network composed of mobile devices, wirelessly connected, and generally characterized by properties of auto-configuration.

Each device that is a member of a MANET can move freely, which involves that the link conditions between different devices can be continuously changing and requires that any node can act as a router for the communications of other nodes. Another important aspect of these networks is that, in general, they can operate independently or be connected to the Internet. When a member of the network is connected to the Internet, usually the network enables Internet access to the other devices that do not have a direct Internet connection.

Regarding security, in MANETs, many types of threats exist that can affect their use. Two of the most dangerous attacks are against the security of wireless communications through denial-of-service and man in the middle (MitM) attacks.

The proposal described in this work aims to be applied mainly to IoT devices used in mobile environments, such as vehicles, for example. In particular, this work proposes the design and implementation of a new lightweight cryptographic scheme to enable secure wireless communications in a MANET. It describes the design and analysis of a new scheme based on a non-interactive zero-knowledge proof for the authenticated exchange of confidential information in IoT, which allows a receiver node to authenticate the sender and to compute a shared secret key from the received message.

B.2.1. Related Works

Security in the Internet of Things is a primary factor. As a proof of that, the new operating system that Google has introduced for the Internet of Things, called Brillo, follows the secure by default paradigm. The major security threats to the Internet of Things, can be classified into three main categories:

- Related to the physical nature of smart objects: cloning of smart things

by untrusted manufacturers, malicious substitution of smart things during installation, firmware replacement attack and extraction of security parameters since smart things may be physically unprotected.

- Related to intercommunication between smart objects: eavesdropping attack if the communication channel is not adequately protected, man-in-the-middle attack during key exchange, routing attacks and denial-of-service attacks.

- Related to sensible data processed by the smart objects: privacy threats.

If secure communication is established over insecure channels through symmetric cryptography, the secure pre-distribution of secret keys is one of the most essential tasks. This topic has attracted much attention among researchers [68] throughout the years. However, the best-known system for the secure distribution of secret keys via insecure channels continues being the Diffie-Hellman scheme [67], which allows two users to compute a shared secret key from two secret numbers and a public exchange of information, thanks to the discrete logarithm problem. Such an algorithm does not include user authentication, which leads to the possibility that am MitM attack can be launched. In order to avoid it, asymmetric cryptography and public key certification can be used. The present work faces the problem of the secure distribution of secret keys based on the idea under Diffie-Hellman scheme, but including authentication and using graph theory problems instead of the discrete logarithm problem.

In the case of MANETs, different proposals exist to protect communications, which are based either on secret-key cryptography or on public-key cryptography.

On the one hand, the security level of many symmetric schemes is high, but their major drawback is the difficulty of the pre-distribution of shared secret keys. In an environment like that of MANETs in the IoT [9], the assumption regarding the existence of a fully-secure channel to transmit the symmetric keys is very difficult to achieve. In addition, if the MANET is large and based only on symmetric cryptography, the number of secret keys that would be required is very high. Thus, as previously mentioned, to solve the problem of secure secret key distribution, asymmetric cryptography can be used.

On the other hand, public-key cryptography offers the capability of a digital signature scheme [95] if the sender decrypts with its private key and the receiver encrypts with the public key of the sender. Through a digital signature, authentication of both the sender and message, e.g., identification and integrity, can be guaranteed. Interestingly, the ability to use digital signatures that public-key cryptography provides is exactly what solves the main

challenge it presents, which is the need to establish trust in public keys to prevent MitM attacks. In an MitM attack, the attacker makes independent connections with two nodes and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is being controlled by the attacker. The solution to this problem involves that each public key is certified so that the digital signature contained in the certificate guarantees the identification of the user responsible for the public key.

Several models to achieve certification of public keys exist [177]. The most common one is based on a public key infrastructure (PKI), where trust is placed in certificate authorities. Another interesting scheme is based on a decentralized trust model called the web of trust, where each user has a ring with a group of public keys that it trusts. The authors of [38] propose an efficient public key management scheme that is suitable for fully-self-organized MANETs, where all nodes play identical roles. Finally, a certificate-less alternative for PKIs is identity-based cryptography, where the public key of each user is some unique piece of information related to its public identity, so that public-key certificates are unnecessary.

The main disadvantage of asymmetric cryptography is the high computational complexity of the schemes, since most public-key cryptosystems are too heavy to be used in lightweight environments like MANETs in IoT. To solve this issue, this work proposes a combination of symmetric and asymmetric cryptography to allow the use of secret session keys in these environments. In particular, the proposal offers both strong authentication of legitimate nodes through open communications and the exchange of secret keys shared between pairs, which can be used as session keys.

Node authentication is here performed using an approach based on the idea of zero-knowledge proof (ZKP) [93], which defines a method to prove the knowledge of a certain piece of information without revealing anything about it. Typical ZKPs are based on several challenges and responses, involving a successive exchange of messages, which implies the need to have a stable and continuous connection between nodes [79].

However, this assumption is impossible in a volatile environment like IoT, where sometimes, devices move at a high speed, such as, for example, smart vehicles. In these cases, a massive exchange of messages to run a typical ZKP can be infeasible due to possible connection failures during the protocol. In order to deal with this problem, the idea of non-interactive ZKP (NIZKP) has emerged in the literature [23].

In an NIZKP, all of the challenges of a typical ZKP are condensed into a single package sent in a single message. This leads to the optimization of the time necessary for the exchange of messages, so that just a single message is necessary, and even this message can be sent as a beacon in broadcast mode. The work [82] shows a method that transforms an interactive protocol into a

non-interactive protocol, which can be applied to turn interactive ZKPs into NIZKPs thanks to the use of hash function. Besides, on the one hand, as a general theoretical result, the authors of [80] present the first NIZKP for NP whose construction is based on one-way permutations and certified trapdoor permutations [140]. On the other hand, as a specific practical result, the work [131] shows an NIZKP of the Hamiltonian cycle problem, which can be used with the proposal described here.

B.2.2. Preliminaries

If a prover P is trying to prove to a verifier V its knowledge of a solution to a difficult problem, it can use a zero-knowledge proof so that V is not able to trick P and discover the solution or any information that can help it to compute anything faster than before. Thus, what V can see thanks to the protocol should be something that it could have computed by itself. A ZKP must fulfil three main properties, usually called completeness, soundness and zero-knowledge. Completeness means that for any valid input, a prover P can always complete the proof successfully; soundness means that no malicious prover P can construct a valid proof system; and zero-knowledge means that no malicious verifier V is able to derive extra knowledge from the interaction.

This work is based on a non-interactive zero-knowledge proof, which can be formalized as follows. If $\{0,1\}^*$ denotes the set of all strings and R denotes a witness, for a language $L \subseteq \{0,1\}^*$, a pair of probabilistic Turing machines (P,V), in which P has probabilistic polynomial time power and V has deterministic polynomial time power, is said to be a non-interactive zero-knowledge proof system of the language L if it fulfills the following conditions related to correctness and security against malicious provers and verifiers:

- Completeness: for any polynomial $p(\cdot)$ and common input $x \in L$, x is accepted by V with a probability greater than $1-\epsilon$:

$$Pr[V(x,R,P(x,R))=1] \geq 1 - \frac{1}{p(|x|)} \tag{B.1}$$

- Soundness: for any interactive Turing machine P' representing a dishonest prover, any polynomial $p(\cdot)$ and any common input $x \in L$ provided by P' x is accepted by V with a probability at most ϵ:

$$Pr[V(x,R,P'(x,R))=1] < \frac{1}{P(|x|)} \tag{B.2}$$

- Zero-knowledge: for any $x \in L$ provided by P, no information is revealed from x to V that it could not compute alone before running the

protocol, which means that there is a probabilistic polynomial time algorithm M, such that:

$$V(x) = x, (R \in 0, 1^{c(|x|)}, P(x, R)) \approx_c M(x)_{x \in L} \tag{B.3}$$

One of the most important factors of any ZKP is the choice of the mathematical problem that forms its basis. The work [92] showed that, under certain complexity assumptions, on the one hand, any NP problem can be used to define a ZKP, and on the other hand, only problems in BPP (bounded-error probabilistic polynomial time) can be used to describe non-interactive zero-knowledge proofs.

In this work, the chosen problem is the graph isomorphism problem. An isomorphism between two graphs is a bijection that preserves the adjacency relationship, *i.e.*, any two vertices of a graph are adjacent if and only if so are their images in the other graph. The graph isomorphism problem consists of determining whether two graphs are isomorphic or not. This problem has been used in cryptography [94] because an efficient algorithm to solve it in general is yet unknown. In particular, the determination of whether two graphs with the same number v of vertices and the same number of edges are isomorphic or not involves a brute force attack, because it requires checking whether some of the $v!$ possible bijections preserve adjacency. In general, the graph isomorphism problem is one of a few problems in computational complexity theory belonging to NP, but not known to belong to a P or to an NP-complete subset [89]. Therefore, its resolution, depending on the size and the type of the involved graphs, can be very difficult. In particular, the graph isomorphism problem has been proven to be in BPP, which opens the door to the definition of NIZKPs based on it.

B.2.3. NIZKP-Based Authentication

The proposal here described is based on a variant of NIZKP that uses only a single message to verify the knowledge and can be adapted to different levels of security, so that the larger the number of challenges, the higher the level of safety for the verifier. In particular, the parameters used in the proposal are shown in Table B.1.

In [23], it was shown that the validity of NIZKPs relies on the computational assumption of an ideal cryptographic hash function. Thus, the proposed scheme uses a cryptographic hash function that fulfills such a requirement.

In the described scheme, each node broadcasts a message to identify itself as a legitimate network node. The message consists of a number of commitments defined by isomorphic graphs generated from an initial graph known by all legitimate nodes. For instance, that graph may represent the network, so that each node represents a network user.

Notation	Meaning
G	Graph known by all legitimate nodes, on which they know how to solve a hard problem
Sol_G	Solution to the hard problem in G
n	Number of challenges
Cha_i	i-th challenge proposed by the verifier
G_i	i-th isomorphic graph used as a commitment
Iso_i	Isomorphism between G and G_i
Res_i	i-th response corresponding to the challenge Cha_i on the graph G_i
$h(\cdot)$	Cryptographic hash function
$LSB(\cdot)$	Least significant bit of an input string
$E_{k_i}(\cdot)$	Symmetric encryption with key k_i
$Subkey$	Contribution of a node to the session key

Tabla B.1: Proposal Parameters.

Each commitment of the message, except the first one, is encrypted and can only be decrypted after verifying all of the previous ones.

In particular, the message is divided into $n + 1$ segments that are all encrypted with different keys, except the first segment, which is not encrypted (see Figure B.1). Thus, a legitimate network user can authenticate itself to join a communication session with another node if this latter node is able to decrypt all of the segments of the message broadcast by the first one, so that it can reach the last segment, which contains the contribution of the sender node to the session key.

The encryption key of each segment depends on the previous segment, so that although someone wants to decipher only the last segment, this is impossible, because it would require the decryption of all previous segments. The security level of the scheme depends on the number of segments of the message, which represent different challenges. The greater the number of segments, the more complex it is to reach the last segment and to obtain the information required for the establishment of the shared key.

After the two-way authentication using the same procedure, based on the idea of the Diffie-Hellman scheme, both nodes will know the shared session key, but with both exchanged subkeys.

Each segment, except the last one, contains an isomorphic copy of the original graph. A one-way hash function known by all legitimate network nodes is used to define the challenge that the receiver must solve on each isomorphic graph. Moreover, the same hash function is used to define the encryption key for each message segment.

The operations that the receiver must perform on the message are:

1. Process the first segment of the message, which is not encrypted.
2. Compute, using the hash function, the challenge that matches the information included in the segment.
3. Check whether the response corresponds to the challenge and the isomorphic graph.

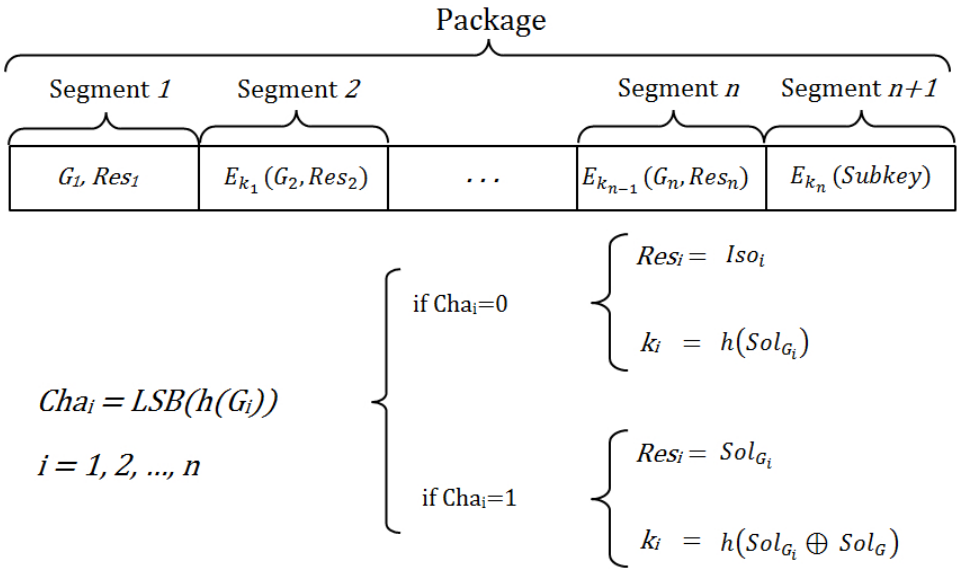

Figura B.1: Components of Sent Messages

4. From the challenge, compute the key it has to use to decrypt the next segment.

5. Apply Steps 2 through 4 until the last segment, which once deciphered contains the information needed to establish the shared secret key.

If the used hash function, problem and graph are adequate, the probability that $Cha_i = 0$ is 1/2. Thus, the probability that a legitimate node knows the key k_1 is 1/2, that it knows the two keys k_1 and k_2 is $1/2^2$,..., and that it knows the n keys k_1, k_2, ..., k_n is $1/2^n$.

The challenges have been chosen as the known ZKP is based on isomorphic graphs. However, in the NIZKP here proposed, the challenges are defined from the result of a Boolean output of a hash function defined through the least significant bit (LSB) applied on each committed isomorphic graph. Thus, for each challenge, the response is defined as follows:

- If $Challenge = 0$, the response is the isomorphism.
- If $Challenge = 1$, the response is the solution to the problem in the isomorphic graph.

The flowchart of the algorithm to be run by the receiver is shown in Figure B.2. The pseudocode of the proposed scheme is shown below.

Algorithm Message 1 Processing

//Params: beacon, encrypted message segments

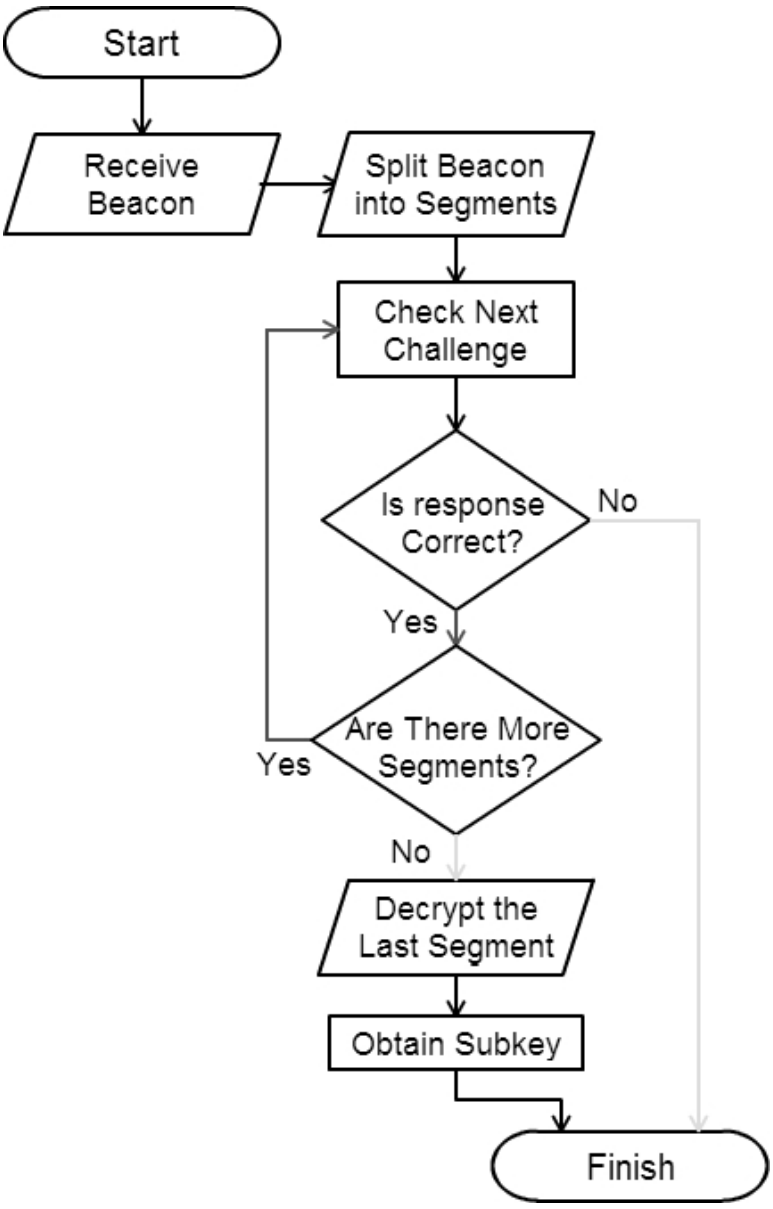

Figura B.2: Flowchart of the Proposed Algorithm

//Params: t_seg, dimension of beacon segments
//Params: solg, solution in the original graph
//Return: subkey, contribution to the key
function getData (char[] *beacon*, int tseg, char[] *solg*)
01: var *segs*[]; // Stores the message segments
02: // Message is divided into $tseg - size$ segments
03: $segs = beacon$.splitByTam($tseg$);

```
// Isomorphic graph and response
// First segment is not encrypted
var gi = getGi(segs[0]);
var res = getRes(segs[0]);
// The challenge is computed
var cha = LSB.hash(gi.getBytes());
// Check whether the response is correct
if (res != response(gi, cha))
  return; // If not correct, abort
endif
// The solution is obtained in gi
var sol = solve(gi);
// ki is the encryption key of the next segment
var ki = cha' * hash(sol) ⊕ cha * hash(sol ⊕ solg)
var decryption;
// The following steps are repeated
for (int i = 1; i < segs.size() − 1; i++){
  // The segment is decrypted with the key ki
  decryption = Crypto.decrypt(segs[i], ki);
  gi = getGi(decryption);
  res = getRes(decryption);
  cha = LSB.hash(gi.getBytes());
  if (res != response(gi, cha))
    return;
  endif
  sol = solve(gi);
  ki = cha' * hash(sol) ⊕ cha * hash(sol ⊕ solg)
}
// Decryption of the last segment provides the
// contribution to the shared key.
return Crypto.decrypt(segs[segs.size()], ki);
endfunction
```

The receiver can access the last message segment thanks to the decryption key obtained from the previous segment after running the algorithm. This last segment contains the contribution of the sending node to the potential session key shared with each receiver.

The initial graph and a secret key that is a solution to a difficult problem in such a graph are known by all legitimate network users. For example, this solution may be a Hamiltonian cycle because the Hamiltonian cycle problem for arbitrary graphs is considered a difficult problem. The Hamiltonian cycle problem consists of determining whether there is a path in the graph that vi-

sits each vertex exactly once. This problem is often considered NP-complete, but there are some particular graphs for which the problem is polynomial or even linear. Because of this, in this work, if the Hamiltonian cycle problem is used, the use of non-planar graphs is suggested. A graph is planar if it can be drawn in a plane without graph edges crossing. In order to check in linear time whether a given graph is non-planar, Theorem 1 can be used:

Theorem 1. For any simple, connected, planar graph with v vertices and e edges: if $v \geq 3$ then $e \leq 3v - 6$.

Thus, this theorem can be used to prove that a graph is not planar when the above relationship between v and e is not fulfilled. In fact, when generating the graph, if a planar graph is obtained, edges are added at random until the theorems condition is not satisfied, thereby ensuring that the graph is non-planar.

The hash function chosen for computing the challenges and encryption keys for each message segment in the implementation of the proposed scheme is the new standard hash function SHA-3 [17] [19].

The symmetric system to encrypt the message segments used in the implementation is the stream cipher of the fourth generation of mobile communications (LTE) [73] [76], known as SNOW3G [74]. This choice is based on the linear computational complexity both in encryption and decryption, which guarantees the efficiency and speed of the encryption and decryption processes.

B.2.4. Applications to the Internet of Things

All of the applications that have been implemented and analyzed in this subsection to prove the effectiveness of the scheme proposed in this work are for a decentralized environment where Wi-Fi Direct and/or Bluetooth Low Energy are used for wireless communications, because already, many of the interconnected objects have access to these technologies, and they are available for Android. In the future, similar applications will be possibly developed based on the prospective LTE Direct technology.

In particular, the cases in which the required degree of confidentiality is that of communications encrypted with a secret session key are the main use cases of the described system.

The distribution and management of credentials for secure communication can be a relatively simple task if only a restricted group of centralized application providers is considered. Instead, in a distributed architecture, like the one of the Internet of Things, many more problems emerge, as explained in [219], because any device can be connected to any device at any time, and devices might not have had any previous contact with each other in advance. Hence, in this scenario, the problem of key management becomes an important problem. A solution may be based on using a scheme like the

proposed one because it is quite flexible and adaptable to the needs of the devices that interact in IoT.

Regarding applications in Mobile Ad-hoc Networks, conducting business transactions in MANETs is an interesting use case of the proposal, because in that scenario, a legitimate network node might want to share its own resources with other legitimate nodes to carry out such transactions. Due to their mobile nature, often, the nodes of a MANET do not have Internet access in many places. Thus, those legitimate nodes that have an Internet connection may want to rent their connection to other legitimate nodes. For this task, both nodes can establish a shared secret session key, which can be done with the scheme proposed.

In general, two different scenarios for the use of variants of the proposed scheme are described below: to report authenticated information unidirectionally either without using secret keys or using a public key. On the one hand, a node may only want to transmit information from an authenticated way (see Figure B.3) so that other legitimate nodes hearing the messages rely on the information that the sending node transmits, thanks to its knowledge of the secret network key used to generate the isomorphic graphs and solutions involved in the described protocol. Two specific examples of use cases within a MANET in this new scenario are event notification or dissemination of advertising shops. The transmission of advertising using the scheme proposed implies that only legitimate nodes can send advertising, which avoids massive spam from nodes that do not belong to the network. On the other hand, a legitimate node may want to spread its public key in an authenticated way and only to other nodes that also belong to the network (see Figure B.3). This use case requires the application of another variant of the proposed scheme where sent beacons hide in their last segment the public key of the sender node. This implies that only legitimate network users can access the public key of the sender hidden in the last segment of the beacon, because the challenges and responses are based on a secret network key. This approach can be used whenever the spread of an event through a MANET requires the use of a digital signature scheme by legitimate nodes, which send their public keys to other legitimate nodes of the network in an authenticated way.

Regarding applications in Vehicular Ad-Hoc Networks, where a vehicular ad-hoc network (VANET) can be seen as a special type of MANET where the nodes are vehicles and the main objective is to prevent adverse circumstances on the roads and achieve more efficient traffic management, one of the most important aims in the design of such networks is to resist security attacks [137]. Thus, in the area of VANETs, the proposed scheme can be used to authenticate vehicles in isolated areas (mountain areas, tunnels, etc.), where no Internet connection is available. Each vehicle can send an authentication message and agree on a shared key to communicate thereafter, following the proposed scheme based on the Diffie-Hellman idea.

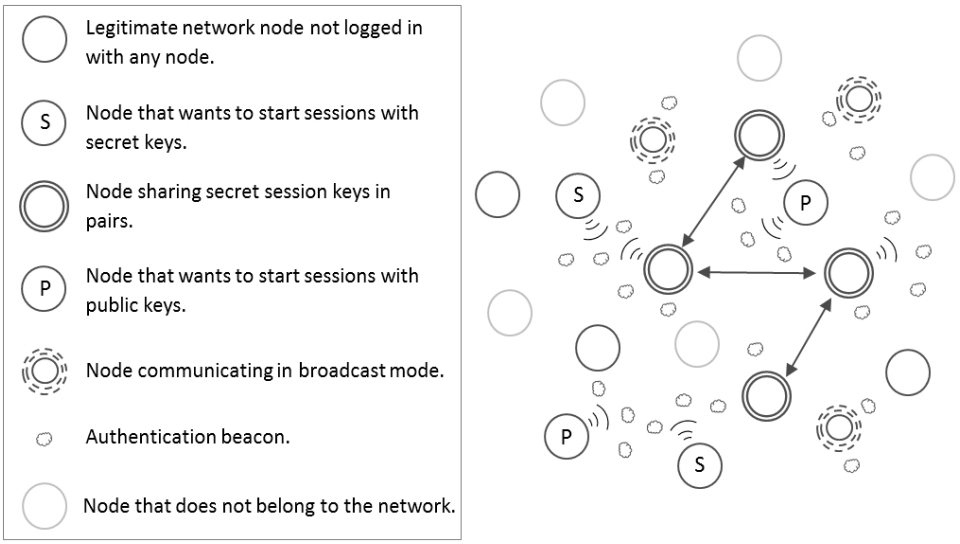

Figura B.3: Types of MANET Nodes

Furthermore, the proposal can be applied to solutions of other problems in VANETs. For example, it can be used to authenticate the information sent from the smart traffic lights to vehicles. Some critical environments require that this information is sent authenticated, so that only legitimate users of the network can decrypt and process it. Thus, for instance, there are proposals like [159] that describes a low-cost solution based on a smart traffic light using a light sensor that provides information in real time about the traffic light color. The solution uses a Bluetooth Low Energy (BLE) module that allows transmitting the state of the traffic light to nearby vehicles, as a beacon notification. This beacon notification can be sent authenticated through the proposal explained.

Regarding applications in sensor networks, the proposal presented here can be interesting, specifically for its application in wireless sensor networks (WSNs). WSNs have evolved significantly in the last few years, generating a promising research area about a fruitful and useful technology. This technology involves two types of entities: sensor nodes and base stations. In general, base stations are more powerful than sensor nodes, but this trend is changing thanks to advances in low-powered technology.

Nowadays, it is possible to create a WSN using platforms like Arduino, Rasperry Pi, Odroid, Intel Edison, *etc.*, which can have several sensors. Since some applications in WSNs only require the information of a specific sensor, the scheme proposed can be used for these platforms to send the sensor information independently and in an authenticated way by using a broadcast beacon to send the data of each sensor. Thus, sent data will only be accessible to legitimate entities.

The applicability of the proposed solution is more to sensor platforms, such as those coordinated through Arduino, than to individual sensors, because Android can be installed in them. Thus, the secure communications based on the proposed scheme can be used on any sensor platform based on Arduino, such as Odroid.

Besides, the proposed authentication scheme can be used to complete other solutions and to add a layer of security. For example, existing WSN models based on preloaded parameters, like broadcast keys for sensor nodes, can be improved by using the scheme presented to generate shared keys and using them as broadcast keys for the sensor nodes.

B.2.5. Security Proofs

In the following, several known attacks are analyzed in relation to the proposal.

On the one hand, the security of the scheme depends on the chosen hash function. A collision attack on a cryptographic hash tries to find two inputs producing the same hash value. Another possible weakness can be a preimage attack, which tries to find a message that has a specific hash value. A cryptographic hash function should resist both preimage and collision attacks. In this work, the implementation uses the hash function SHA-3 for several operations. The algorithm behind SHA-3 is the Keccak function. With Keccak, it is possible to target a given security strength level by choosing the appropriate capacity, *i.e.*, for a given capacity c, Keccak is claimed to resist any attack up to complexity $2^{\frac{c}{2}}$. This approach is similar to the one suggested to choose the security strength used in [13] by NIST (National Institute of Standards and Technology).

On the other hand, the security of the proposal depends on the symmetric cipher used to encrypt each segment. The encryption used in the implementation is the basis of the security in LTE communications, called SNOW 3G. The SNOW 3G core is supplied as portable Verilog, with a Verilog Hardware Description Language (VHDL) version available, thus allowing customers to carry out an internal code review to ensure its security.

Therefore, the security of cryptographic operations is guaranteed due to the used standards: SHA-3 and SNOW 3G. The SHA-3 algorithm is based on a permutation where collisions and preimages can be found for its cryptographic hash function in one query to the permutation. The sponge construction has been proven to be indifferentiable from a random oracle if the underlying permutation is assumed to be ideal [16], and this result applies to Keccak function. Optimal bounds have been obtained on collision resistance and on preimage and second preimage resistance for Keccak in the ideal permutation model [6]. Regarding SNOW 3G, it has linear time complexity, which guarantees efficiency during the encryption/decryption process.

Furthermore, security proofs of SNOW 3G are based on the assumption that this encryption system behaves like a perfect random function of the key [126].

The specific problem under the ZKP is fundamental for the security of the scheme. The implemented scheme uses the two graph problems of the graph isomorphism and of the Hamiltonian cycle.

Regarding possible attacks to the problem of the graph isomorphism, some efficient algorithms for some specific graphs have been proposed. For example, in [10], several probabilistic algorithms were discussed; and in [60], some algorithms for determining whether two graphs are isomorphic or not have been described. However, the proposed scheme can use graphs of different sizes, and depending on the size of these graphs, the security can be increased. The generated graphs are completely random, and therefore, none of the proposed algorithms are useful for finding the secret. Therefore, under the hypothesis of using well-chosen instances of the problem, the graph isomorphism problem can be considered NP-complete for the particular execution of the proposal.

With respect to the Hamiltonian cycle problem, under the condition of the defined scheme, the problem can be also considered NP-complete. This is achieved with the use of non-planar graphs, because through the theorem shown in a previous section, we can ensure that in the implementation, none of the graphs that are generated in the scheme are planar. Again, under the hypothesis of using well-chosen instances of the problem, the Hamiltonian cycle problem can be considered NP-complete for the particular execution of the proposal. Thus, a real attack based on the used graph problems is not possible because the complexity of the chosen problems can be considered NP-complete.

An MitM attack happens when an attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. This is one of the most used attack schemes in wireless networks.

In the case of the proposal, an MitM attack cannot be carried out because there is no way to gain information during the transaction. In particular, the scheme uses a single beacon to send some information from the sender to the receiver. Therefore, there is no communication between them, so that other users cannot intercept communication and, hence, cannot impersonate communications.

If the scheme is used to establish a secret key through the Diffie-Hellman algorithm, thanks to the fact that the protocol uses strong mutual authentication with secret keys, it is robust against an MitM attack. Mutual authentication refers to two parties authenticating each other at the same time.

Consequently, an MitM attack against the proposed scheme would not succeed. The attacker could intercept beacons, but without access to secret

parameters of the scheme, it cannot get any confidential information. The configuration parameters of the proposed scheme are only accessible for the legitimate users of the network and are provided by a trusted third party during the initialization of the scheme.

In MANETs, due to the lack of a centralized structure, denial of service (DoS) attacks can be frequent. In order to protect the proposed scheme from DoS attacks, although the communication produced with the application is through a non-secure channel, only legitimate nodes of the networks are able to send and decrypt valid messages.

The proposed scheme is resistant against a brute force attack. A brute force attack, also called exhaustive key search, is a trial-and-error method used to obtain relevant data through the generation of a large number of consecutive guesses about the desired data. In the proposal described, since the desired data are contained in the last segment of the sent package, a brute force attack on the key used to encrypt the last segment would be required. In order to make it possible that a brute force attack could be used to discover the key used to encrypt such a segment, it could be necessary to check all of the possible keys. The key used to encrypt each segment has a fixed size defined by the hash function used to obfuscate the response to the previous segment challenge. SHA-3 has been used as a hash function in the implementation, so since its smallest output is 224 bits, a brute force attack would involve checking 2^{224} combinations. If the scheme is used as a key agreement protocol, since in that case, interaction between parties exists, a maximum response time could be defined in order to prevent possible brute-force attacks.

Another dangerous attack in MANETs is the sibling attack, which occurs when a node illegitimately uses multiple identities. This problem is avoided in the proposed scheme thanks to the distributed nature of the used NIZKP.

Finally, the proposal is also resistant to identity theft because node access is controlled by an NIZKP.

In a nutshell, usual attacks have no harmful effect on the proposal, because its security is supported by NIZKP based on complex mathematical problems and current standard hashing and encryption.

B.2.6. Implementation

The described scheme is intended for its use in areas related to the Internet of Things. It has been implemented for Android and Android Wear platforms, which are two examples of the proliferation of devices in this new dimension of the Internet. Android is the most popular smartphone operating system, with over 80 % market share worldwide. Android Wear is the operating system based on Android for wearable devices of the same company and has more than 90 % of the market share in the devices of its kind.

Therefore, all of the results presented here are the result of the implementation of the scheme in these two platforms belonging to the Android Open Source Project. The source code is open source under a Git repository on the GitHub Platform [150].

Figure B.4 shows a screenshot of the Android application that was created to analyze the performance of the scheme in smartphones.

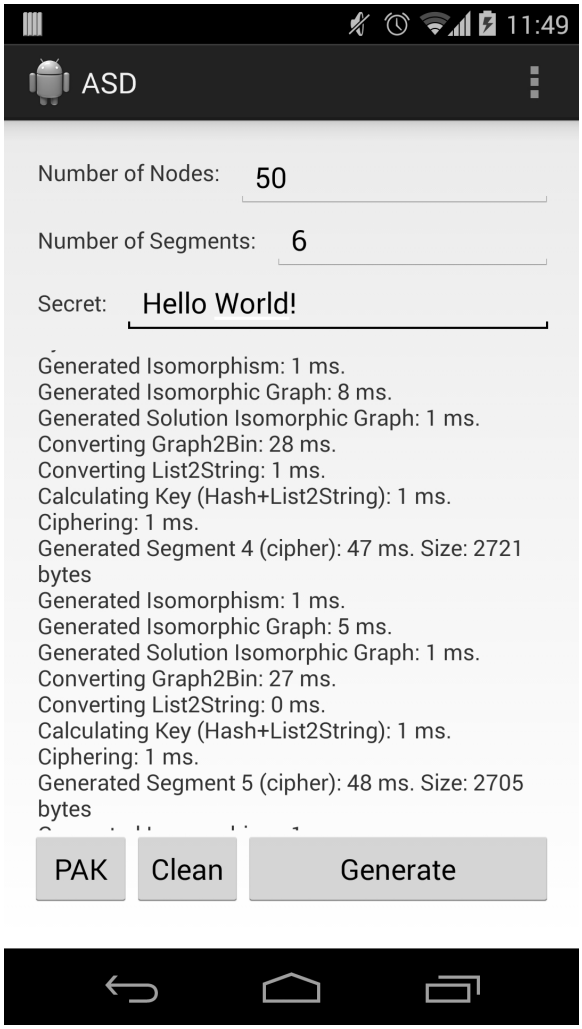

Figura B.4: Android Application Screenshot

Despite the implementation described in this section having been focused on IoT devices where it is possible to have an Android operating system, like current smartphones and smartwatches, this does not mean that the necessary devices have to be extremely powerful devices. For instance, there are sensor platforms, like Odroid, that, despite their small size, are able to run an operating system like Android, so consequently, they can be used

to run the proposed system. Besides, Google has recently introduced a new operating system for IoT, called Brillo, which is a fork of Android, so the implementation described is easily adaptable to future devices with the Brillo operating system.

One of the premises that the scheme meets is that it is lightweight in terms of size and the speed of computation, so it fulfills the requirements of the devices of the IoT. Therefore, the size of the message or package has been reduced in the implementation to be as small as possible in order to use the smallest possible space in memory and in order to have fast communication between devices. Because of this, also the storage format of each of the elements that make up the message has been optimized. Thus, the following measures have been taken to serialize these elements:

- Graphs: Graphs are serialized by denoting their adjacency matrices into integers, because this is their fastest implementation. This serialization has been also improved by using hexadecimal rather than integer numbers. In this case, the storage size is smaller, but the speed serialization is quite slow. Thus, a faster serialization has been chosen compared to a slight improvement in the package size, so graphs have been serialized through their adjacency matrices.

For instance, given the adjacency matrix:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

it can be represented in one dimension as:

$$01101101001100100000110110$$

This one-dimensional representation is serialized by converting it into an integer of eight digits, formed of five integers separated by the character comma.

$$13, 20, 25, 1, 22$$

- Solutions: Solutions have been serialized by using lists of integers separated by the character comma because both types of solutions, the Hamiltonian cycle and the isomorphism between graphs, can be represented as lists. For example, if the isomorphism between two graphs is $1 \to 2$, $2 \to 5$, $3 \to 1$, $4 \to 3$, $5 \to 4$, using array indexes as original nodes and the values of the array as the values of the new nodes, the isomorphism is represented as:

$$2, 5, 1, 3, 4$$

- Segments: Each segment of the message is serialized in hexadecimal. Besides, all of the segments, but the first one, are encrypted as explained in previous sections. The last segment contains a secret encrypted by the key that can be obtained with the solution to the challenge of the previous segment. For instance, the aspect of a segment containing an isomorphic graph and challenge and response corresponding to the graph is the following, expressed with hexadecimal characters:

$$A324D0E3F19$$

- Message or package: The complete message is the concatenation of all of the generated segments. To separate the segments, the character | is used. An example of a package with three segments is:

$$A324D0E3F19|F9223B3EE34|DC34F212ACB$$

Given the format that has been used to represent the elements that are part of the proposed scheme, optimal package sizes have been achieved.

The message size is given by the dimension of the graph used to represent the network. The more nodes a graph has, the more space is needed. This involves the segments being larger, resulting in larger package sizes.

Therefore, we have analyzed which sizes per segment are recommended depending on the number of nodes in the graph, which are shown in Figure B.5.

This clearly reveals a polynomial tendency of order two that represents the relationship between the size of the segments and the number of nodes of the graph that represents the network. Thus, the following polynomial function (see Equation B.4) has been calculated that defines the data, so that segment sizes for graphs with more nodes can be estimated.

$$y = 0{,}9765x^2 + 5{,}8046x - 1{,}1513 \tag{B.4}$$

Thus, for instance, a scheme in which the graph of the network is defined by 50 nodes and the number of challenges that defines the scheme package is six would require a package size of 16 kilobytes without including the size of the shared secret. Thus, in order to send a message in a single package securely by using this scheme, we have used six challenges for 50 nodes, and the sent message has an overhead of only 16 kilobytes.

The computational time required for the devices to create the packages to send depends on the number of segments that are defined in the package. Thus, the larger the number of segments, the greater its security and the more time that the microprocessor of the device requires for generating the package.

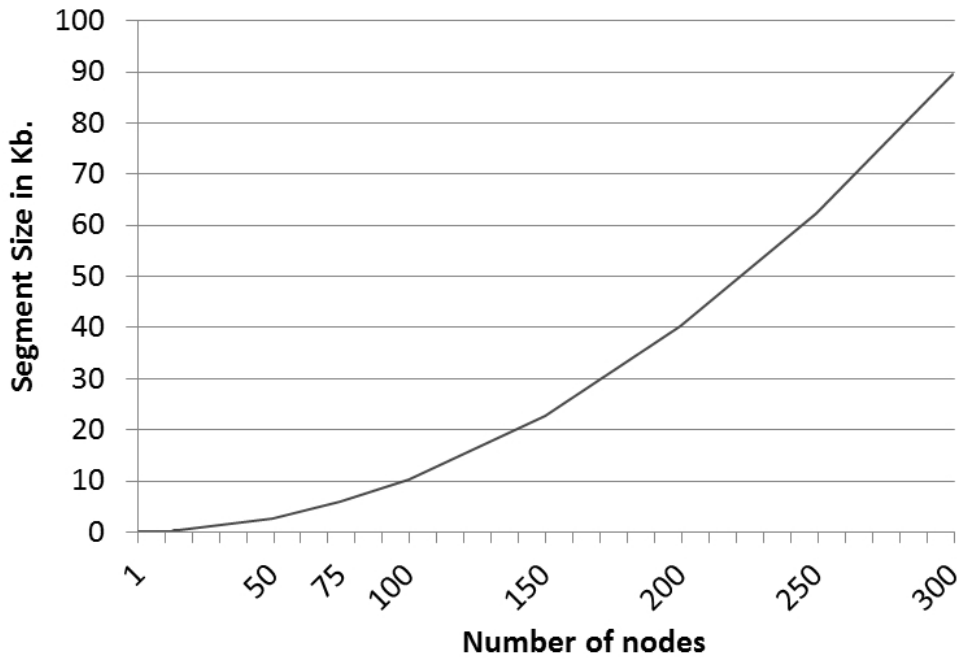

Figura B.5: Segment Size Trend

For the tests, smartphones have been chosen from three possible ranges, low-cost, mid-cost and high-cost, and being several years old. The selected smartphones models are: Motorola Moto G, Samsung Galaxy S3 and LG Nexus 5. This selection has been taken to verify the effectiveness of the scheme on devices with limited computing capacity.

Furthermore, Android Wear smartwatches have been also used for testing, specifically the LG G Watch and Samsung Gear Live models. Since the smartwatches depend on a smartphone, which must be constantly connected via Bluetooth Low Energy, package generation is in fact done from the smartphone.

Considering all of these characteristics, the programming language that has been used is Java, because it is the most widely-used platform for programming in the Android Open Source Project. After making dozens of experiments, the average of the results is as shown below. The time for segment generation based on the number of nodes in the graph is shown in Figure B.6.

However, the results shown in Figure B.6 are strongly conditioned by the process of the serialization of graphs. In fact, this process requires much more computation time than the sum of all of the remaining steps of the scheme. Not surprisingly, the serialization process for large graphs requires more than 98 % of the time required to build the package. The first steps

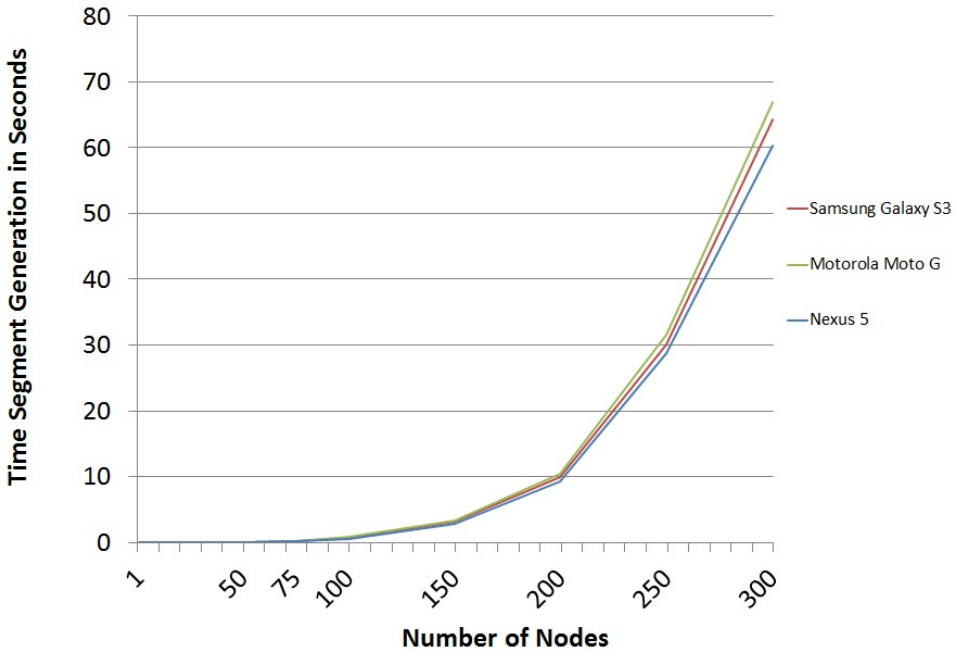

Figura B.6: Segment Generation Time Trend

of the serialization process were much slower, more than 20 times, before the adjacency matrix was converted into hexadecimal characters. However, after optimizing the process, it was concluded that it is more efficient to convert it into integers using the characteristics of Java classes. The problem is that now, the occupied space is greater than in the previous process. The computational speed has improved by a factor of 20, but still, the process remains very slow compared to other operations of the scheme, due to the limitations of the Java Virtual Machine. For example, for a network of 300 nodes, a graph serialization takes about 59,138 ms. The complete generation of a segment, including the serialization graph, takes about 60,345 ms. These results can be substantially improved by implementing the scheme in a low-level language that Android Open Source Project has, such as the programming language C, which is a compiled, not interpreted, language like Java.

From the results of Figure B.6, the trend has been obtained, and the associated polynomial of order four has been calculated (see Equation (B.5)), which represents the number of nodes and the segment generation time. This equation can be seen as an estimation of how long it would take to create packages for testing, depending on the number of nodes of the graph.

$$y = (5e-6)x^4 + (23e-4)x^3 - 0{,}537x^2 + 30{,}548x - 227{,}48 \qquad \text{(B.5)}$$

For example, in a scheme where the graph is defined by 50 nodes and

the number of challenges that defines the scheme package is six, the time to build the package is 300 ms. Thus, a mobile device with limited capabilities could easily generate packages for the proposed scheme, as demonstrated in these implementations.

Finally, we have analyzed the time it would take mobile devices to decipher the package that comes from other mobile devices. These package-processing times are really short, as evidenced by the processing time per segment shown in Figure B.7.

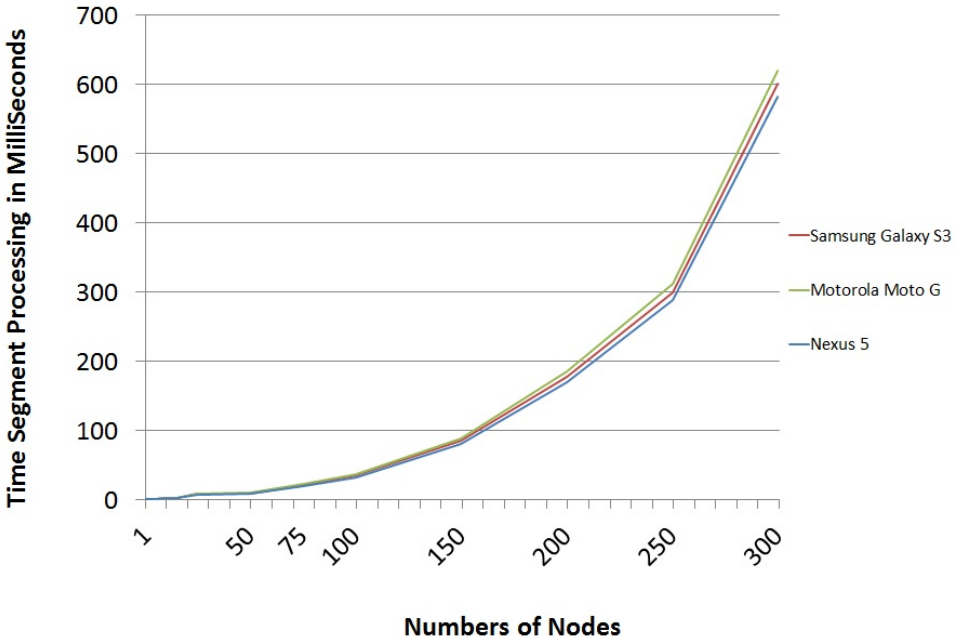

Figura B.7: Segment Processing Time Trend

After analyzing the above results, it was concluded that the trend of the data follows a polynomial function of order four (see Equation (B.6)). This equation can be used to estimate the processing time per segment depending on the nodes of the graph representing the network.

$$y = (2e-9)x^5(-1e-6)x^4 + (2e-4)x^3 - 0{,}0184x^2 + 0{,}673x - 1{,}7392 \text{(B.6)}$$

This means that, for instance, a scheme in which the graph of the network is defined by 50 nodes and the number of challenges that defines the scheme package is six, the time a mobile device takes to process a received package to get the secret is only 48 ms.

B.2.7. Comparative Analysis

In the recent literature, it is hard to find novel schemes proposed for the authentication of mobile devices, specifically in the Internet of Things.

		Conf1. [101]	Conf2. [101]	Conf3. [101]	Conf4. [101]	Our Scheme
10 Challenges	Time	469	1302	484	1522	454
	Size	4045	4045	4045	4045	17,826
100 Challenges	Time	3422	8070	3703	9824	5665
	Size	39,595	39,595	39,595	39,595	187,132

Tabla B.2: Comparative Data: Time (ms) and Size (bytes).

Many existing research works propose lightweight authentication schemes based on challenges and responses, but none of the proposed schemes have been implemented completely. Many authors omit one of the most important pieces of information that can characterize a scheme of this kind: the time required for authentication. Just a few authors have analyzed the transmission bit rate, the percentage of packages that are lost in interactive schemes and/or the total real time that schemes require. In this work, those parameters are studied, but the percentage of lost packages is not analyzed, because the scheme is non-interactive and just one single message is sent, so no loss can happen.

We have analyzed several lightweight authentication schemes applied to the Internet of Things. For instance, [101] proposes a zero-knowledge proof authentication algorithm based on isomorphic graphs and describes its evaluation and implementation. The proposed mechanism allows authentication with varying confidence and security levels. Such a work describes an implementation on conventional computers (with different configurations), so that the starting conditions are more powerful than when using mobile devices, as in this proposal, mainly because of the efficiency of the used programming language. On the one hand, the scheme proposed uses Java as a programming language, so that the implementation is as cross-platform as possible and, thus, can be applied to many current mobile devices. On the other hand, the compared scheme uses random graphs of 41 nodes. In order to allow the comparison, the scheme proposed has been also implemented with graphs of 41 nodes. The comparison results are shown in Table B.2, where the four different hardware configurations described are shown. Another consideration is the fact that the compared scheme uses an interactive ZKP with six exchanges of messages, while the proposal is based on an NIZKP in which only one single message is necessary.

From Table B.2, it can be concluded that the scheme that has been designed here is computationally faster considering the average results and characteristics of both implementations. On the other hand, the compared scheme uses fewer memory bytes. However, the relationship between memory and time is better in the scheme proposed here than in the others. This is due to the fact that the difference in memory space of a few kilobytes is acceptable for todays mobile devices, both when being sent with wireless technologies and when being stored and processed.

Other schemes with an objective similar to that of the one proposed here

have been also compared. One of them was chosen for this study due to its popularity, frequent application and use of Diffie-Hellman key exchange [67]. Such a method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communication channel, so that this key can then be used to encrypt subsequent communications using a symmetric key cipher. However, if no additional security measure is used, the Diffie-Hellman key exchange is vulnerable to an MitM attack. This vulnerability is possible when Diffie-Hellman key exchange does not include the authentication of participants. However, feasible solutions for authenticated Diffie-Hellman schemes, such as the one proposed here, allow preventing such attacks.

After reviewing the relevant bibliography, a scheme was chosen that implements an authenticated Diffie-Hellman protocol for the comparison. The scheme is called password authenticated key (PAK) Diffie-Hellman exchange [28] and proposes to add mutual authentication based on a memorizable password, to the basic, unauthenticated Diffie-Hellman key exchange. PAK allows two parties to authenticate themselves while performing the Diffie-Hellman exchange. It also provides a secure and authenticated key-exchange protocol, which ensures forward secrecy and is secure against offline dictionary attacks when passwords are used.

For the comparison, the PAK scheme was implemented for Android and Android Wear devices. Till now, the PAK scheme had not been implemented on any Android platform, so this work provides the first implementation of the PAK scheme in the most popular platform of mobile devices. Both the source code of the scheme proposed and the source code for the PAK scheme have been released as open source and are available in a hosted repository on GitHub [150]. In the tests, the shared secret information in both schemes has been the typical Hello World! used in computer programming environments.

In particular, in order to compare the proposed scheme with the PAK scheme, the following configuration was used:

- Graphs of 41 nodes, which are large enough to be considered secure in the scheme; thus, an attack on the graph isomorphism would involve 41! iterations, which would be unfeasible.
- Packages with 3, 4, 5, 6, 7 and 8 challenges.
- The time for these comparisons includes both package generation by the sender and unpacking the package by the user who receives it.

The results of the comparison are shown in Table B.3. The conclusion is that in general, the scheme proposed has a similar performance as the PAK scheme, and even in some cases, the proposed scheme slightly improves the results of the PAK scheme.

PAK Scheme	Our Scheme	
Time (ms)	Challenges	Time (ms)
197	3	86
	4	112
	5	153
	6	176
	7	195
	8	221

Tabla B.3: PAK scheme *vs.* the Proposal.

B.3. Revocation in VANETs

This section describes a solution for the efficient and methodical management of revocation in vehicular ad-hoc networks, for both certificate-based and identity-based authentication. It proposes the use of an authenticated data structure based on a dynamic hash tree, which is a k-ary tree and Huffman Codes, together with a new version of the SHA-3 hash function. This combination allows optimizing search and insertion operations in the tree. Consequently, the proposal is very useful both when vehicular networks are widely used, and in urban environments. Simulation results, obtained through the combination of NS-2 and SUMO performs, are promising and confirm this hypothesis.

B.3.1. Introduction

Authentication is a crucial requirement for any communication network. On the one hand, an efficient way to authenticate legitimate and honest nodes is necessary. On the other hand, being able to exclude compromised nodes is fundamental to guarantee trustworthiness of network services.

When communication security is based on public-key cryptography, a central problem is to guarantee that a particular public key is authentic and valid. The traditional approach to this problem is through public-key certificates emitted by a Public-Key Infrastructure (PKI), in which a Certificate Authority (CA) certifies ownership and validity of public-key certificates. This solution presents many difficulties because the issues associated with certificate management are quite complicated and expensive. A different approach is the so-called Identity-Based Cryptography (IBC), where each users public key is his/her public IDentity (ID) so that the need for public-key certificates is eliminated.

In order to use any public-key cryptosystem in practice, an efficient revocation mechanism is necessary because private keys may become compromised. Traditionally, this problem has been solved through a centralized

approach based on the existence of a Trusted Third Party (TTP), which is usually a CA distributing the so-called Certificate Revocation Lists (CRLs) that can be seen as blacklists of revoked certificates. Alternatively, some authors have proposed an approach based on hash trees as Authenticated Data Structures (ADSs) for a more efficient management of certificate revocation.

Vehicular Ad-hoc NETworks (VANETs) are self-organizing networks built up from moving vehicles that communicate with each other mainly to prevent adverse circumstances on the roads, but also to achieve more efficient traffic management. Security in VANETs faces many challenges due to the open broadcasting of wireless communications and the high-speed mobility of vehicles. In these networks, any malicious misbehaving user that can inject false information, or modify/replay any previously disseminated message, could be fatal to the others. VANETs are considered a promising research area of mobile communications because they offer a wide variety of possible applications, ranging from the aforementioned road safety and transport efficiency, to commercial services, passenger comfort, and infotainment delivery. Furthermore, VANETs can be seen as an extension of mobile ad-hoc networks where there are not only mobile nodes, named On-Board Units (OBUs), but also static nodes, named Road-Side Units (RSUs). The so-called Intelligent Transportation System (ITS) includes two types of communications:

- between OBUs: Vehicle TO Vehicle (V2V)
- between OBUs and RSUs: Vehicle TO Infrastructure (V2I) and Infrastructure TO Vehicle (I2V).

It is crucial to have a good system for revocation of malicious vehicles to protect the safety of other users. Some of the most useful applications in VANETs require efficient revocation systems. Various applications based on information from other vehicles obtained through V2V, V2I and I2V communications to exchange event-driven or periodic messages are highlighted in [181]. Among them, three remarkable applications are based on:

- *Cooperative forward collision warning.* This system accomplishes the goals necessary to assist a vehicle in avoiding becoming involved in an accident with the vehicle travelling ahead of it. The system uses V2V communication with multi hop relaying in order to send warning messages.
- *Vehicle-based road condition warning.* Vehicles collect information about road conditions via their sensors, and after collecting sufficient information at their OBUs, they process these data to determine the road situation in order to send warning messages to other vehicles through V2V communication.

- *Cooperative collision warning.* The main goal of this application is to warn the driver about any predicted accident, by using V2V communication.

All these applications require an accurate and reliable system to exchange messages between vehicles. The receiver first must verify the reliability of the sender in order to know whether the received information is trustable or not. Therefore, an effective system of verification of revoked users is necessary.

Both the European standard for ITS, named ITS-G5, and its American counterpart, named Wireless Access in Vehicular Environment (WAVE), are based on the IEEE 802.11p amendment to the IEEE 802.11 standard. The WAVE standard on the so-called Dedicated Short-Range Communications (DSRC) channels is defined as the only wireless technology that can potentially meet the extremely short latency requirement for road safety messaging and control under any condition because DSRC was specifically designed for automotive use. According to DSRC/WAVE, vehicles periodically exchange with nearby vehicles beacons containing sender information such as location and speed because many VANET applications, such as cooperative collision warning, rely on the information embedded in these beacons.

Within the family of standards for vehicular communications IEEE 1609 based on the IEEE 802.11p, the standard 1609.2 deals in particular with the issues related to security services for applications and management messages. This standard describes the use of PKIs, CAs and CRLs, and implies that in order to revoke a vehicle, a CRL has to be issued by the CA to the RSUs, who are in charge of sending this information to the OBUs. Each vehicle is assumed to have a pair of keys: a private signing key and a public verification key certified by the CA; and any VANET message must contain: a timestamp with the creation time, the sender signature, and the sender public-key certificate. In particular, the IEEE 1609.2 standard proposes both broadcast authentication and non-repudiation through the use of the elliptic curve digital signature algorithm.

In order to protect privacy in VANETs, each OBU can obtain multiple certified key pairs and use different public keys each time. These public keys are linked to pseudonyms that allow preventing location tracking by eavesdroppers. Therefore, once VANETs are implemented in practice on a large scale, the size of CRLs will grow rapidly due to the increasing number of OBUs and to the use of such multiple pseudonyms. Thus, it is foreseeable that if CRLs are used, they will become extremely large and unmanageable. Moreover, this context can bring a phenomenon known as implosion request, consisting of several nodes who synchronously want to download the CRL at the time of its updating, producing serious congestion and overload of the network, which could ultimately lead to a longer latency in the process of validating a certificate.

The proposal described proposes the use of hash trees to achieve coopera-

tive revocation of malicious users both in certificate-based and in IBC-based authentication in VANETs. In particular, it uses a k-ary hash tree as an ADS for revocation management. By using this ADS, the process of query on the validity of public certificates/pseudonyms will be more efficient because OBUs will send queries to RSUs, who will answer them on behalf of the TTP. In this way, at the same time this TTP will no longer be a bottleneck, and OBUs will not have to download any entire CRL. Instead of that, they will have to manage hash trees where the leaf nodes contain revoked certificates/pseudonyms. In particular, the used k-ary trees are based on the application of a duplex construction of the Secure Hash Algorithm SHA-3 recently chosen as standard, because the combination of both structures allows improving efficiency of updating and querying revoked certificates/pseudonyms.

B.3.2. Related Works

In many applications, the use of public-key cryptography is essential for information security [21]. In those cases, the revocation problem is one of the most difficult to solve.

Usual revocation procedures are based on a CA that manages revoked public-key certificates by including the corresponding certificate serial numbers in a CRL and distributing this CRL within the network in order to let users know which nodes are no longer trustworthy [194]. Under these circumstances, it is very important that the distribution of the CRL is done efficiently in order to allow that the knowledge about untrustworthy nodes can be spread quickly to the entire network.

As aforementioned, the family of standards IEEE 1609 describes the use of PKI in VANETs. In particular, the work [109] defines a proposal for the use of a PKI to protect messages and mutually authenticate entities in VANETs.

Also based on a PKI, a well-known solution for strong authentication in VANETs is based on the signature of each message [113]. However, the use of a traditional approach to PKIs may fail to satisfy the real time requirement in vehicular communications because according to the DSRC protocol, each OBU will periodically transmit beacons so even in a normal traffic scenario, it is a very rigorous requirement to deploy an authentication scheme that allows at the same time efficient revocation of invalid public keys, and efficient use of valid public keys, which is exactly the main goal of this work.

For revocation in VANETs, previous works assume that the entire CRL may be delivered by broadcasting it directly from RSUs to OBUs [118], and then distributed among OBUs cooperatively [188]. However, the large size of VANETs, and consequent large size of the CRLs, makes this approach infeasible due to the overhead it would cause to network communications. This issue is further increased with the use of multiple pseudonyms for the

nodes, what has been suggested to protect privacy and anonymity of OBUs [200].

Since there are almost one thousand million cars in the world [180], considering the use of pseudonyms, a direct conclusion is that the number of revoked certificates might reach soon the same amount, one thousand million. On the other hand, assuming that each certificate takes at least 224 bits, in such a case the CRL size would be 224 Gbits, what means that its management following the traditional approach would not be efficient. Even though regional CAs were used and the CRLs could be reduced to 1 Gbit, by using the 802.11a protocol to communicate with RSUs in range, the maximum download speed of OBUs would be between 6 and 54 Mbit/s depending on vehicle speed and road congestion, so on average an OBU would need more than 30 seconds to download a regional CRL from an RSU. A straight consequence of this size problem is that a new CRL cannot be issued very often, what would affect the freshness of revocation data. On the other hand, if a known technique for large data transfers were used for CRL distribution as solution for the size problem, it would result in higher latencies, what would also impact in the revocation data validity. Consequently, a solution not requiring the distribution of the full CRL from RSUs to OBUs, like the one proposed in this work, would be very helpful for the secure and efficient operation of VANETs.

A general revocation method not based on CRLs, called Online Certificate Status Protocol (OCSP) [193], involves a multitude of validation agents that respond to client queries with signed replies indicating the current status of a target certificate. This explicit revocation method has an unpleasant side effect because it divulges too much information. Since validation agents constitute a global service, they must involve enough replication to handle the load of all validation queries, what means that the signature key must be replicated across many servers, which is either insecure or expensive.

Another general solution not based on CRLs, called Certificate Revocation Tree (CRT), was proposed in [127] as an improvement of OCSP involving a single highly secure entity that periodically posts a signed CRL like data structure to many insecure validation agents so that users query these agents. In CRTs, the leaf nodes are statements concerning revoked certificates, and the CA signs the root. By using CRTs, the responder can prove the status of any certificate by showing the path from the root to the leaf node without signing the response, because the signatures of any leaf node are identical, and given by the signature contained in the root. Thus, no trust in the responder is necessary. The proposal here described is based on this idea.

In general, a hash tree is a tree structure whose nodes contain digests that can be used to verify larger pieces of data [183]. The leaf nodes in a hash tree are hashes of data blocks while nodes further up in the tree

are the hashes of their respective children so that the root of the tree is the digest representing the whole structure. Hash trees usually require the use of a cryptographic hash function in order to prevent collisions. Most implementations of hash trees are binary, but this work proposes the use of the more general structure of k-ary trees because when combining it with a particular choice of cryptographic hash function, it is possible to optimize the update of the hash tree.

The basic ADS proposed in [127] is a Merkle hash tree [182] where the leaf nodes represent revoked certificates sorted by serial number. A client sends a query to the nearest agent, which produces a short proof that the target certificate is (or not) on the CRT. The work[117] introduces several methods to traverse Merkle trees allowing time space trade-offs. Other ADSs based on multi-dimensional tree structures are studied in [184] to support efficient search queries, allowing the retrieval of authenticated certificates from an untrusted repository used for dissemination by various credential issuers. Besides, many tree-balancing algorithms have been proposed in the bibliography for hash trees [59]. For instance, AVL trees are balanced by applying rotation, B-trees are balanced by manipulating the degrees of the nodes, and 2-3 trees contain only nodes with at least 2 and at most 3 children. However, in the particular application of public-key revocation, balancing trees does not necessary minimize the overall communication.

Another interesting problem with CRTs appears each time a certificate is revoked as the whole tree must be recomputed and restructured. Skip-lists proposed in [96] [97] can be seen as a natural and efficient structure to reduce communication by balancing the CRT. However, they are not good solutions for other problems such as insertion of new leaf nodes.

Hash trees are usually based on widely used hash functions. The scheme proposes the use of a new version of Keccak as cryptographic hash function in the hash tree. Keccak is the cryptographic hash function used in the new SHA-3 standard [19]. The requirements set by NIST for SHA-3 candidates included typical security properties of hash functions, such as collision resistance, preimage resistance and second preimage resistance [7]. Different types of implementations of SHA-3 finalists have been evaluated in several works [102] [8], obtaining in most cases positive conclusions. The original SHA-3 uses a sponge construction [17], which in a cryptographic context is an operating mode on the base of a fixed length transformation and a padding rule. Instead of it, this scheme proposes a duplex construction [18]. The main advantage of the duplex construction is that it provides digests on the input blocks received so far, what is applied in the proposal here described so that the hash tree is constructed efficiently.

Previous proposals that can be considered close to this work are [192] [88] because they use hash trees for revocation in VANETs. However, they use neither perfect k-ary trees nor a specific hash to optimize tree operations.

B.3.3. Tree-Based Proposal

The scheme described is based on using as ADS a k-ary tree (See Figure B.8), which is a rooted tree where each node has no more than k children. The use of k-ary hash trees instead of binary trees allows increasing efficiency of the construction and update of hash trees. Thus, one of the major drawbacks of ordered tree structures, which is the necessary reconstruction when there are changes in the tree, only occurs when the k-ary tree requires a new level of depth, because otherwise the nodes simply are inserted from left to right to complete the level of depth corresponding to their query frequency. In this way, our proposal is based on a dynamic tree-based data structure that varies depending on the number of revocations.

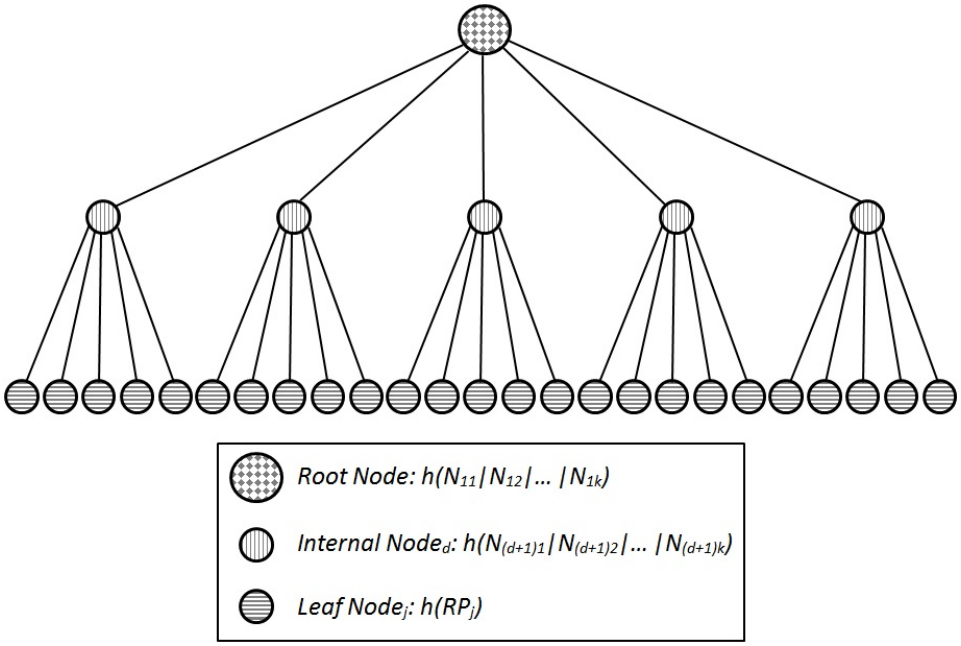

Figura B.8: Hash Tree Based on a 5-ary Tree

The proposed model is based on the following notation:

- h: Cryptographic hash function used in the hash tree.
- D (≥ 1): maximum Depth of the hash tree.
- d ($< D$): Depth of an internal node in the hash tree.
- s: Number of revoked certificates/pseudonyms.
- R_j ($j = 1, 2, ..., s$): Serial number of the $j-th$ Revoked certificate/pseudonym.

- N_{ij} ($i = D - d$ and $j = 0, 1...$): Internal Node of the hash tree obtained by hashing the concatenation of all the digests contained in its children.
- N_{0j} ($j = 0, 1...$): Leaf node of the hash tree containing $h(R_j)$, ordered according to revocation.
- k: Maximum number of children for each internal node in the hash tree.
- f: Keccak function used in SHA-3.
- n: Bit size of the digest of h, which is here assumed to be the lowest possible size of SHA-3 digest, 224.
- b: Bit size of the input to f, which is here assumed to be one of the possible values of Keccak, 800.
- r: Bit size of input blocks after padding for h, which is here assumed to be 352.
- c: Difference between b and r, which is here assumed to be as in SHA-3, $2n$, that is 448.
- l: Bit size of output blocks for building the digest of h, which is here assumed to be lower than r.

On the one hand, in order to build the tree, the first parameter to consider is the maximum number of children per node. This parameter defines the k of the k-ary tree to be built. If k equals 2, the resulting tree is the typical binary tree, but the proposal allows different values for k, such as 3, 4, 5, etc. For instance, a 5-ary tree is shown in Figure B.8.

On the other hand, in order to find a node in the tree, a hash table is used to map each revoked certificate/pseudonym with the exact path that defines the tree.

In the proposed scheme, the bandwidth cost of sending the new versions of the revocation tree from the TTP to the OBUs is significantly reduced compared with proposals, existing because in general, only nodes in the tree that have changed need to be updated. This implies a significant improvement with respect to previous tree-based schemes for revocation management because one of their main problems is the necessary update of the entire hash tree every time a new leaf node is added or an existing leaf node is deleted.

The authenticity of the used hash tree structure is guaranteed thanks to the TTP signature of the root. The procedure to follow when the part of the revocation tree that is necessary for authenticity verification is pushed from an RSU to an OBU after this latter queries the first one about a certificate/pseudonym, is as follows. If the RSU finds the digest of the queried

certificate/pseudonym among the leaf nodes of the tree because it is revoked, then the RSU sends to the OBU the route from the root to the corresponding leaf node, along with all the siblings of the nodes on this path. After checking all the digests corresponding to the received path and the TTP signature of the root, the OBU gets convinced of the validity of the evidence on the revoked certificate/pseudonym received from the RSU.

Regarding the cryptographic hash function h used in the hash tree, this proposal is based on the use of a new version of the Secure Hash Algorithm SHA-3. In SHA-3, the basic cryptographic hash function f called Keccak contains 24 rounds of a basic transformation and its input is represented by a 5×5 matrix of 64-bit lanes. In contrast, our proposal is based on 32-bit lanes. Another proposed variation of SHA-3 is the use of a duplex version of the sponge structure of SHA-3. On the one hand, like the sponge construction of SHA-3, the proposal based on a duplex construction also uses Keccak as fixed-length transformation f, the same padding rule and data bit rate r. On the other hand, unlike a sponge function, the duplex construction output corresponding to an input string might be obtained through the concatenation of the outputs resulting from successive input blocks (see Figure B.9).

The use of the duplex construction in the proposed hash tree allows the insertion of a new revoked certificate/pseudonym as new leaf of the tree by running a new iteration of the duplex construction only on the new revoked certificate/pseudonym. In particular, the RSU can take advantage of all the digests corresponding to the sibling nodes of the new node, which were computed in previous iterations, by simply discarding the same minimum number of the last bits of each one of those digests so that the total size of the resulting digest of all the children remains the same, n. This proposed procedure makes the hash tree construction more efficient than previous tree-based schemes when a new leaf corresponding to a new revoked certificate/pseudonym has to be inserted in the tree. While the maximum number of children of an internal node has not been reached, the RSU has to store not only all the digests of the tree structure but also the state resulting from the application of Keccak hash function f in the last iteration corresponding to such internal node, in order to use it as input in a next iteration.

Periodic deletions of certificates/pseudonyms that are in the tree and reach their expiration date, require reconstruction of the part of the tree involving the path from those nodes to the root. Thus, in order to maximize the proposal, such tree reconstruction is linked to the moment when all the sibling nodes of some internal node expire because this avoids unnecessary reductions of the system efficiency by having to reconstruct the tree very often.

The TTP is responsible for periodically updating the tree by deleting the expired certificates/pseudonyms, and for reconstructing the tree when neces-

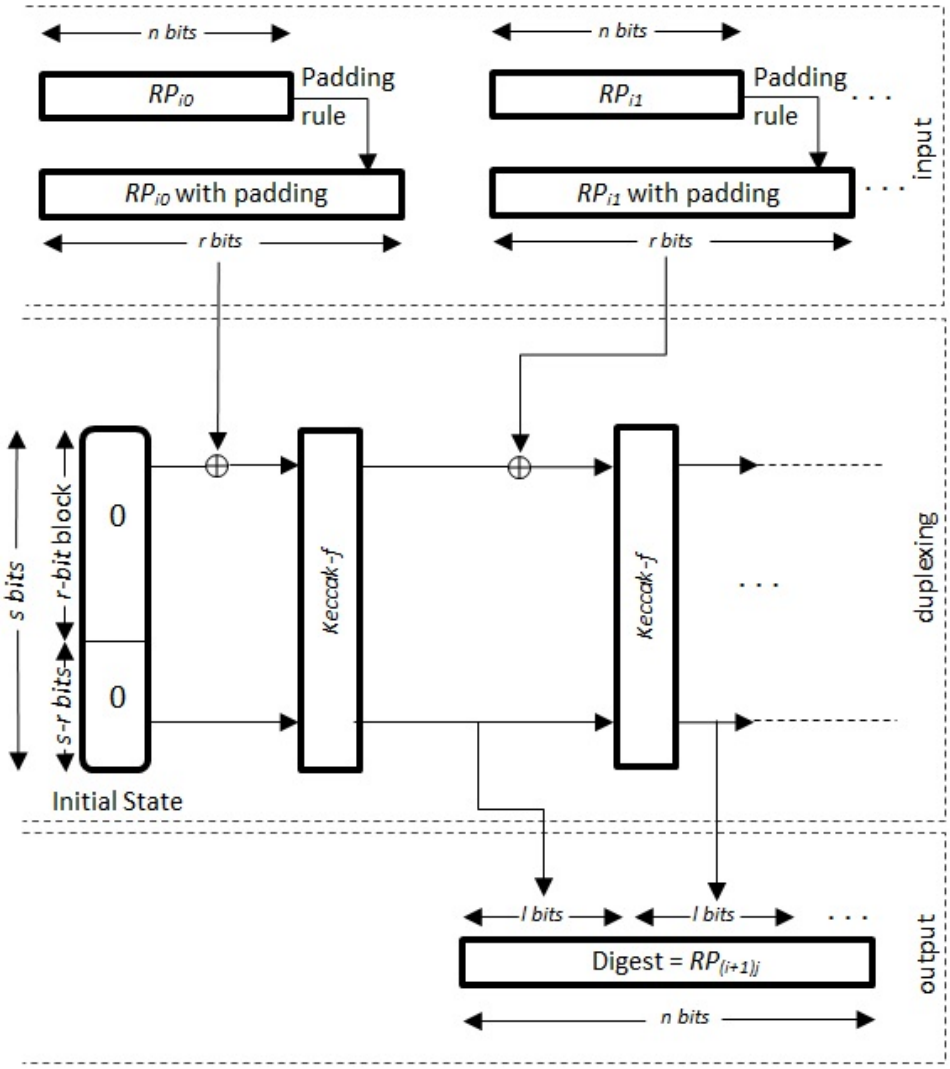

Figura B.9: Proposed Duplex Construction

sary. After each update, the TTP sends the corresponding modifications of the updated tree to all RSUs. The RSU has to search vehicle certificates/pseudonyms in the revocation tree each time an OBU requests it. The RSU must provide the requesting OBU either with a verifiable revocation proof of any revoked certificate/pseudonym or with a signed message indicating that the queried certificate/pseudonym has not been revoked and is labelled as OK. In the first case, by using the answer data, the OBU can verify the TTP signature of the received signed root, recompute the root of the revocation tree, and check it by comparing it with the received signed root.

When using ID-based cryptography, the proposal is based on the use of

a group of pseudonyms set for each OBU, so that for each pseudonym the TTP provides the OBU with a corresponding private key. If any of these pseudonyms is revoked by the TTP, it inserts all the pseudonyms corresponding to the same OBU in the revocation tree.

The proposal scheme has been object of different implementations and simulations, in order to evaluate its performance.

The proposal can be considered computationally efficient because it obviates the need to sign each RSU reply. In general, the proposal does not require trusting all RSUs. Indeed, the only case when trust on RSU is necessary is when it provides an OK answer because this could be a fraud.

When an OBU receives an OK message signed by a cheating RSU, it trusts it momentarily. However, when it contacts another RSU, it asks it again about the same certificate/pseudonym. If this RSU provides the OBU with a proof of revocation whose timestamp contradicts the OK answer signed by the questioned RSU, the OBU sends to the latter RSU an impeachment on the questioned RSU, so that the honest RSU can send it to the TTP, who will revoke its public key by deleting it directly from the revoked RSU. Otherwise, if the second RSU also sends a signed OK message, the OBU goes on asking about the same certificate/pseudonym until it reaches either a contradiction or a prefixed trust threshold.

Each OBU stores locally in two separate and complementary structures, the pseudonyms of those OBUs that it has previously checked as unreliable, and of those OBUs that have been reliable till then. Therefore, in the future, if it reconnects with any of these vehicles, it can use such information to decide how to proceed. If there is no RSU nearby, it uses these data to decide whether to establish the communication or not. Otherwise, even if there is an RSU nearby, there is no need to re-ask it about a checked revoked certificate/pseudonym.

B.3.4. Tree Algorithms

In order to optimize operations in the tree, we have developed specific algorithms that work on perfect k-ary hash trees. First, an algorithm to find a revoked pseudonym in the tree (see Algorithm 11) has been designed and implemented. The idea of this algorithm is to return the full path from the root node to the node searched, including all sibling nodes found on the route. This set of nodes is the proof of verification that a node is revoked. So, a user can recalculate the root node to check if the signature of the certificate authority is valid.

Besides, a specific algorithm to insert a new revoked pseudonym in the tree has been defined. Thus, only a single and simple iteration is required to recalculate the hash function, according to which the new node is inserted (see Algorithm 12). This algorithm improves the efficiency of the traditio-

Algorithm 11: Search Algorithm

Input: *rpSearch*, Tree-ID of leaf node to be searched.
Output: *retPath*, Path from root to leaf node.

```
1 Add Root Node to retPath;
2 for i ← (D − 1) to 0 do
3 |   Compute rpSearch branch to i height;
4 |   Add Siblings Nodes of the computed top level branch to retPath;
5 Compute rpSearch Leaf Node;
6 Add rpSearch Leaf Node and Sibling Nodes to retPath;
7 return retPath;
```

nal insertion algorithms that use hash structures. This is because it takes advantage of the characteristics of the duplex construction used.

Algorithm 12: Insertion Algorithm

Input: *rpNew*, Tree-ID of Leaf Node to be inserted.

```
1 Go to Last Depth Level where the rpNew Node must be inserted;
2 if It is necessary to Create a New Depth Level then
3 |   Generate New Depth Level;
4 |   Insert rpNew Node;
5 |   Reconstruct Hash Tree;
6 else
7 |   Insert rpNew Node;
```

An algorithm for removing a revoked pseudonym from the tree is also proposed (see Algorithm 13), where leaf nodes of the tree are deleted only when they expire. From time to time the system run a process that is responsible for removing tree expired certificates. An expired certificate is invalid for any kind of communication, so that vehicles do not ask about such certificates because they do not trust directly.

Finally, a fast and efficient algorithm for the restructuring of the tree is proposed in Algorithm 14. A tree is to be restructured only when the deletion or insertion of a node causes the removal or creation of a depth level.

B.3.5. Huffman Version

Specifically, a version of the proposal using Huffman Codes has been designed, where revoked keys are represented by leaf nodes, which are at different depths depending on the different times that the corresponding vehicles normally spend on the road.

The Authenticated Data Structure based on a tree model proposed in

Algorithm 13: Deletion Algorithm

Input: *rpDelete*, Tree-ID of Leaf Node to be deleted.

```
Define nodeToDelete;
Assign rpDelete to nodeToDelete;
while nodeToDelete has no Leaf Node do
    Compute nodeToDelete Parent Node;
    Delete nodeToDelete Node;
    Assign nodeToDelete Parent Node to nodeToDelete;
Reconstruct Hash Tree;
```

Algorithm 14: Reconstruction Algorithm

Input: *Tree*, Input Hash Tree.

Output: *Tree*, Reconstructed Hash Tree.

```
for i ← 1 to D do
    Compute the branch of i on the Tree;
    Relocate the nodes of the branch on the Tree;
for i ← (D − 1) to 0 do
    for j ← 0 to Number_of_Nodes_at_Level_i do
        Recompute Hash Value of the Node j from its Children at
         Level i+1;
return Tree;
```

this proposal is defined based on several components. The hash function used to define the revocation tree is denoted by $h(...)$. The digest obtained with the hash function h applied on the concatenation of the inputs N_i, where $i = 0, 1, ...$ is denoted by $h(N_0|N_1|...)$. The Depth of the hash tree, which indicates the number of different types of vehicles used in the scheme, is denoted by D, and its minimum possible value is 1. The depth of a node x in the tree is denoted by d_x and its value is less than D. The total number of revoked pseudonyms is denoted by t. The Revoked Pseudonym is denoted by RP_j, where $j = 1, 2, ..., t$. A Node in the hash tree in denoted by N_{ij}, where $i = D - d_{N_{ij}}$ and $j = 0, 1, ...$ The maximum number of children for each internal node is denoted by k.

The proposal described here is based on k-ary Huffman trees, which are used as hash trees for revocation management in VANETs (see Figure B.10).

In the hash tree proposed in the new version, if possible, every new revoked node is always inserted in the tree as a new sibling to the right. In order to optimize this operation, the hash function that is used is a sponge construction of SHA-3 [17] because this allows to leverage the calculation of the hash of all previous siblings to avoid recalculating the entire digest.

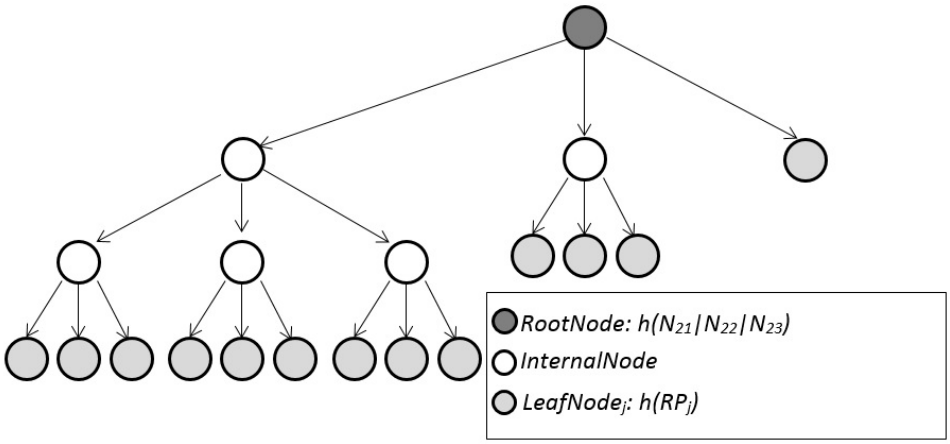

Figura B.10: Example of 3-ary Hash Tree with 3 Levels

In addition, the proposal uses the idea behind Huffman codes to build trees where the shortest paths correspond to the most queried certificates. Therefore, different levels of the tree are defined according to the estimated probability or frequency of queries on each vehicle. It can be assumed that this frequency will be closely related to the time each vehicle spends usually on the road. Thus, the leaf nodes closer to the root node will correspond to revoked vehicles that spend more time on the road, such as public transport vehicles.

B.3.6. Simulations of k-ary Tree

The following subsection contain respectively a few comments regarding the choice of parameters, some results obtained from different implementations and simulations of the proposal, and brief algorithmic descriptions of the tree operations.

The choice of adequate values for the different parameters in our proposal must be done carefully, taking into account the relationships among them. In particular, since the maximum tree size:

$$n(1 + k + k^2 + k^3 + \cdots + k^D) = \frac{n(k^{(D+1)}-1)}{k-1}$$

is upperbounded by the size of available memory in the RSU, and the maximum number of leaf nodes of the k-ary tree k^D is lowerbounded by the number of revoked certificates/pseudonyms s, both conditions can be used to deduce the optimal value for k.

In order to achieve a realistic evaluation, real data of vehicular environments and detailed studies have been used. Depending on the number of nodes in vehicular environments, the number of revoked certificates has been

estimated using the proposal described in [14]. In particular, this statistical research of NIST estimated that 10 % of the certificates need to be revoked.

The data used in the comparisons have been chosen according to the study presented in [65]. Such a research focuses on Madrid city and estimated its fleet in 1.7 million vehicles. Using these data and the NIST study about revocation rate in VANETs.

A real scenario was simulated with the characteristics shown in Table B.4.

Parameter	**Value**
Scenario	Madrid City (Spain, 2014)
Size scenario	$25Km^2$
Simulation time	1000 s
Number of vehicles	1349
% of Vehicles with OBU	10, 20, ..., 90, 100
MAC	IEEE 802.11p
Propagation model	DSDV
Transport protocol	UDP
Size package	1 Kb
Link Layer	LL

Tabla B.4: Parameter Values for the Simulation Scenario

Vehicle features in the simulations are defined in Table B.5.

Parameter	**Value**
Speed	[0-33] m/s
Transmission distance	55 m
Antenna	OmniAntenna
txPower	1.4 mW
rxPower	0.9 mW
sensePower	0.00000175 mW
idlePower	0 mW
Initial Energy	75 J

Tabla B.5: Vehicle and OBU Profile

For the traffic generation, SUMO software tool [236] has been used. SUMO is a free and open traffic simulation suite that allows modelling of intermodal traffic systems, including road vehicles, public transport and pedestrians.

In order to simulate the architecture and communications of a VANET, a tool called NS-2 [197] was used. NS-2 is a discrete event simulator targeted at networking research that provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and

satellite) networks. NS-2 was used for the simulation of the network. The data generated with NS-2 were subsequently transferred to SUMO for their display on the map. Sent messages and interactions between nodes in NS-2 can be seen in Figure B.11.

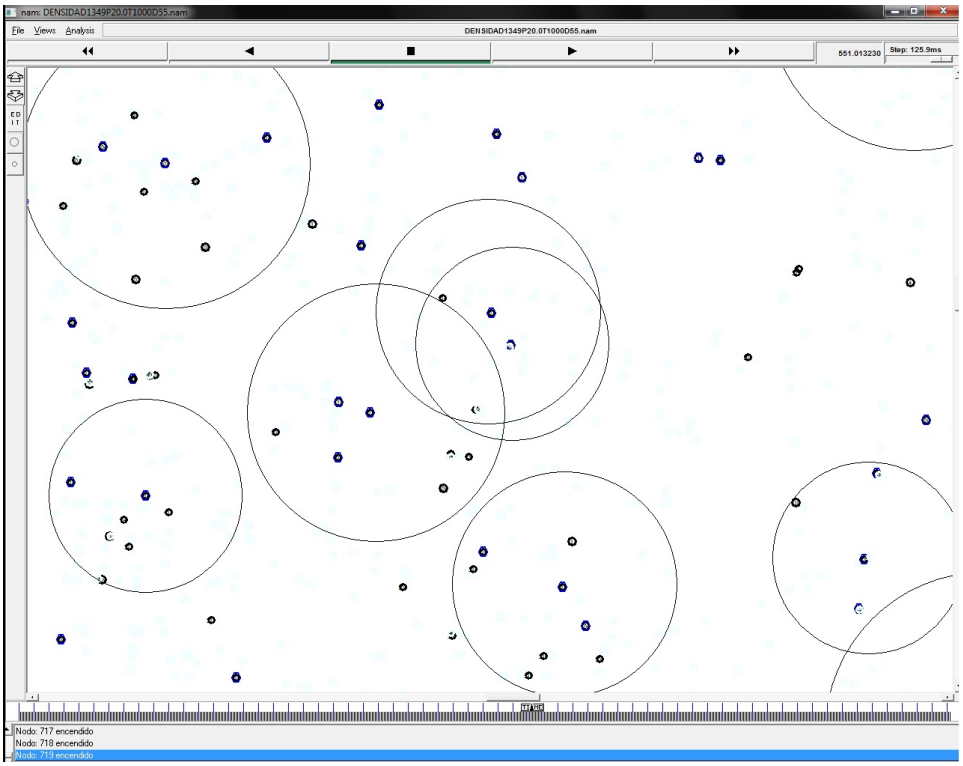

Figura B.11: Architecture and Communications of VANET Simulation

The interaction between the traffic generated with SUMO (Figure B.12) and the network simulated with NS-2 is generated using MOVE.

MOVE allows users to rapidly generate realistic mobility models for VANET simulations. MOVE is built on top of an open source micro-traffic simulator SUMO. Its output is a realistic mobility model that can be immediately used with popular network simulators, such as NS-2.

With the obtained simulation data and settings shown in previous tables, the visual aspect of SUMO software corresponding to the selected scenario is shown in Figure B.12.

From the comparison between the size of the typical revocation list and the size of the proposed revocation tree, the obtained results can be seen in Figure B.13.

It shows that although the proposed revocation structure uses more space than the usual revocation list, the revocation proof requires much less space. This is because when using typical revocation lists each vehicle obtains the

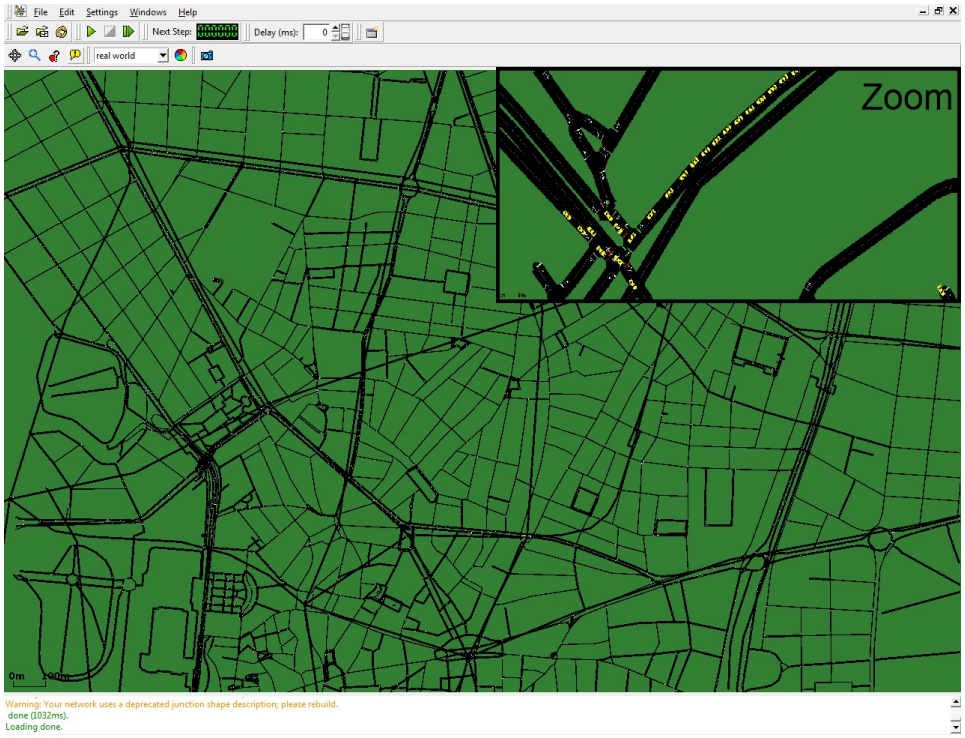

Figura B.12: Example of SUMO Traffic Simulation

full list each time it requires it, while in our scheme it only gets a path in the hash tree when it is necessary. As RSUs are assumed to have sufficient memory, their need to store the complete hash tree is not a problem. The key issue to optimize is the shipping of the revocation proof because vehicles move at high speeds and typical revocation lists are too heavy to be sent in these environments.

In fact, vehicle ad-hoc networks use a specific standard for wireless communications between vehicles and infrastructures, called WAVE. Wireless Access for Vehicular Environments (WAVE) is an approved amendment to the IEEE 802.11 standard, also known as IEEE 802.11p. This standard is required to support the Intelligent Transportation Systems applications in the short-range communications. The communication between vehicles or between the vehicles and the roadside infrastructure is relied on the band of 5.9 GHz and it is optimized to operate in very high-speed environments.

With respect to the authentication process, the number of authentications and queries, together with the revocation tree designed and implemented are represented in Figure B.14, depending on the percentage of vehicles with OBUs in the described scenario.

Traffic connections and authentications generated in the VANET have been estimated at random in order to check node revocation. This estima-

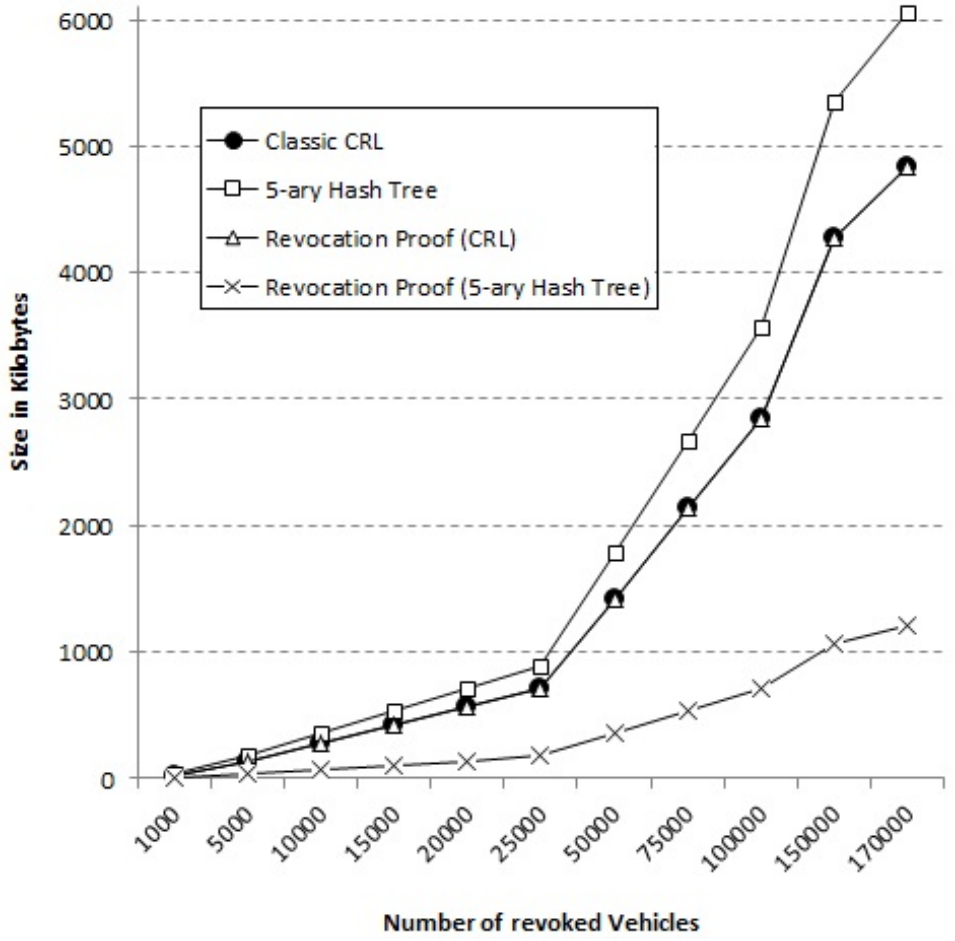

Figura B.13: Comparison with the Typical Revocation Lists

tion was drawn up on the simulation scenario and compared with the traffic generated by conventional revocation lists on the same scenario. The comparison results, which demonstrate the efficiency of our proposal can be seen in Figure B.15.

B.3.7. Simulations of Huffman Version

According to the scenario selected in the previous subsection, there are 1,7 million vehicles in Madrid city. Of these, around 80 percent are private cars. There are 15646 taxi licenses and 2022 buses of the public transport company in Madrid. Deliveries account for 14 percent of total commutes.

After analyzing these data, we propose to divide vehicles into 4 different types: Type A are Public Buses, Type B are Taxis and Emergency Vehicles, Type C are Private Transport Buses and Delivery Vehicles, and Type D are

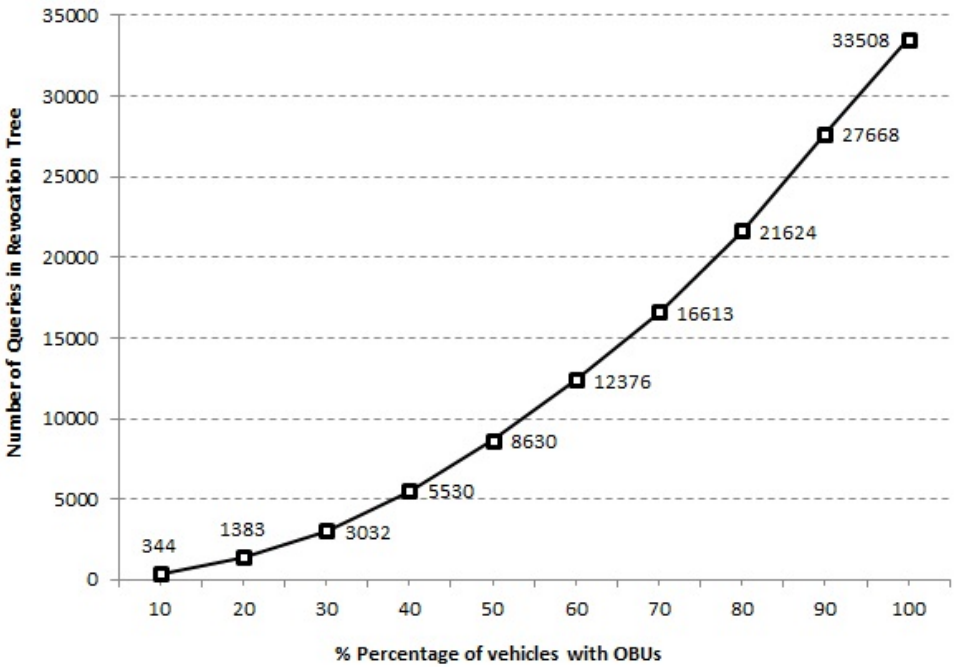

Figura B.14: Queries in the Simulated Scenario

Private Vehicles and Motorcycles.

Knowing that the total number of revoked nodes may be around 17000, we can estimate that at most, the tree structure revocation will contain the following revoked nodes by type:

- Estimated number of revoked vehicles of Type A: 20.
- Estimated number of revoked vehicles of Type B: 390.
- Estimated number of revoked vehicles of Type C: 2990.
- Estimated number of revoked vehicles of Type D: 13600.

With these data, starting from an initial value of k=21, and running an iterative construction procedure from lower levels to higher levels, the resulting range for optimal values of k is $(20, 49]$ and the final result is k=35, so the proposed revocation tree contains 35 nodes in the first level, 525 in the second one, 4725 in the third level, and 60725 potential revoked nodes in the fourth level. With this value of k=21, an oversize exists at the deepest level, so if additional leaf nodes are needed in upper levels, the procedure explained in previous sections to change internal nodes into leaf nodes is possible.

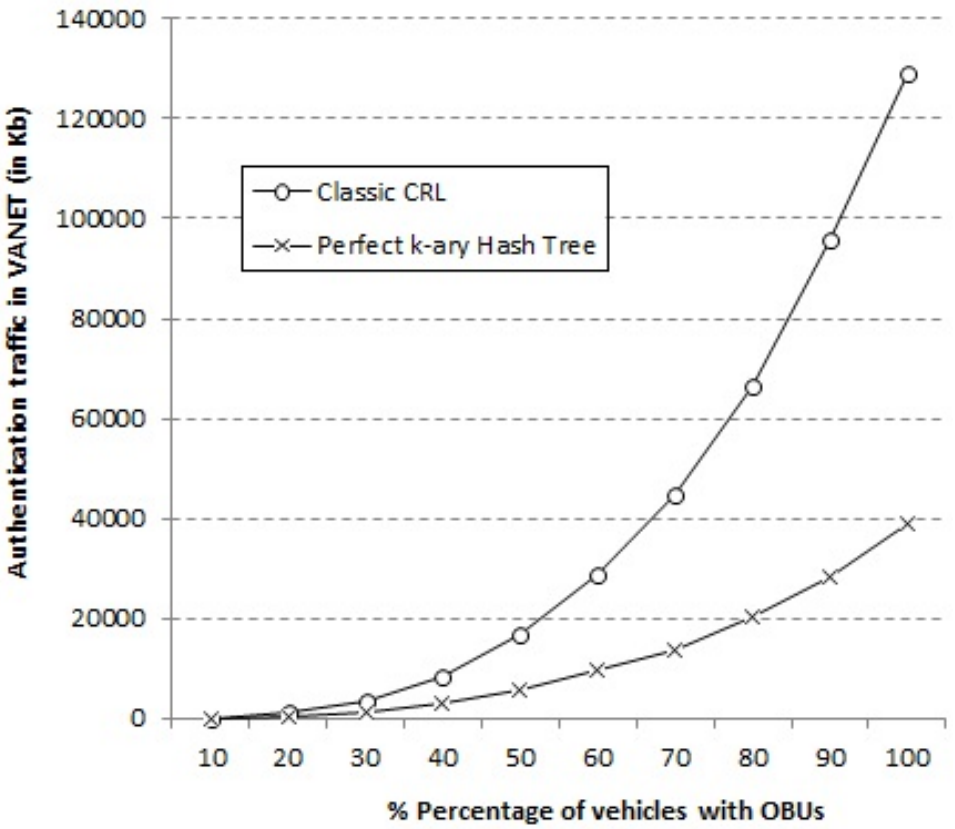

Figura B.15: Traffic Generated in the Verification Process

The size of the revocation test in this real example, considering 228 bits, increases with the depth at which is the revoked node in the tree ($228 \cdot 35 \cdot d_x$), so that the maximum size of the test is:

- Maximum size of test for Type A nodes: 8 Kilobits.
- Maximum size of test for Type B nodes: 16 Kilobits.
- Maximum size of test for Type C nodes: 32 Kilobits.
- Maximum size of test for Type D nodes: 64 Kilobits.

When using classical CRLs, the proof that a node is revoked involves sending the whole CRL, so in this example that would mean sending $17000 * 228 = 3876000$ bits (3876 Kilobits) in all cases. Thus, for this real example, the proposed method involves an impressive improvement in all cases (see Figure B.16). For the comparisons shown in that image, the proposal explained in the previous subsection, using 5-ary hash trees as revocation structure, is included.

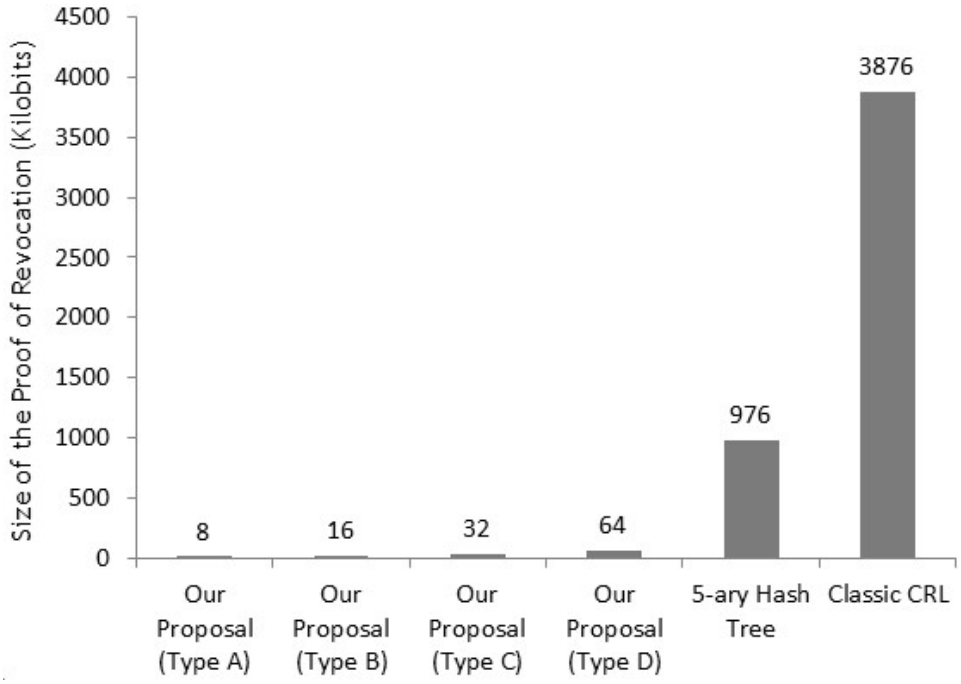

Figura B.16: Comparison Between Sizes of Revocation Proofs

As can be seen in Figure B.16, the system proposed improves the overall size of requests that are made to the structure of revocation in all cases. This is because different sizes are used depending on the type of vehicle that is revoked, resulting in a smaller size for the most common vehicles on the roads (see Figure B.17).

As deduced from Figure B.17, the query of the most frequent vehicles on the roads, which are the most likely to be consulted, is optimized.

B.4. Mobile Applications

During the Thesis, it has always desired to take the researches from theory to practice. Therefore, from the outset, it has implemented all proposed algorithms in different mobile applications, or apps, that would be used to demonstrate the usefulness of the proposals. In current times, the more portable and popular technology is the mobile phone. In addition, it has evolved a rapid pace, combining the user experience with computing capacity. For this reason, the researches have been publish to the general public, through developing apps for everyday use, mostly centered in vehicular environments.

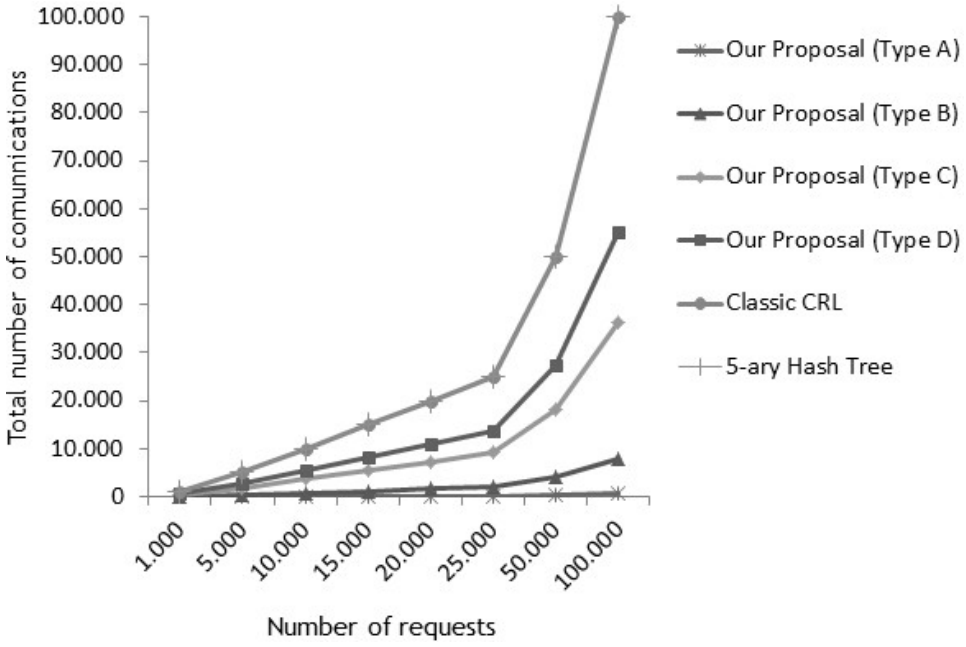

Figura B.17: Number of Requests by Vehicle Type in Different Proposals

B.4.1. Carpooling

The rapid increase in the number of vehicles has raised air pollution, which is one of the main factors of the global warming. This fact has led governments to take steps to try to decrease pollution in urban centres, through measures such as encouraging the use of public transport. However, public transport is not always the most convenient and/or affordable solution for everybody.

Another practical solution to the problem would be to avoid the low occupancy of most vehicles in cities, through sharing cars so that empty seats are used in most trips. This modality is known by the term carpooling and has been proposed as an effective way to reduce pollution and spending. A related but different approach, known as carsharing, is based on collective fleets of cars with multiple users, but such a solution does not solve as many problems as carpooling. Moreover, ridesharing is the general term used to refer to solutions for sharing the use of a car with other people in order to travel to a given destination. Besides carsharing and carpooling (also known as real-time, instant, dynamic or ad-hoc ridesharing), ridesharing also includes other versions known as slugging, lift sharing and covoiturage.

The main difference between slugging and other versions of ridesharing is that slugging involves the creation of free and unofficial ad-hoc carpool networks. On the other hand, the term lift-sharing refers to interurban sharing car travels, while the term carpooling refers to urban journeys within

the city. Finally, covoiturage concept is of French origin and has a social implication because it involves not only car sharing but also making new friends.

This work does not address other ridesharing solutions different from carpooling.

These types of collaborative solutions are increasingly used since the beginning of the economic crisis, thanks to technology 2.0. They are applicable in almost any environment, but are especially useful in situations like universities, holidays, long journeys and urban centres because in these situations both the owners and passengers of vehicles have the same motivations to consider the carpooling solution. Usually, their main goal is to share fuel cost, but there may be other reasons such as try to avoid parking problems, want to talk and meet new people or to make a contribution to the environmental protection, etc.

The main problem of carpooling is reliability because the service requires users to be confident that the driver will drive them to their destination, and drivers must trust users to allow them to enter their cars. An improvement to increase reliability of existing carpooling solutions is proposed here, based on the use of the latest technological advances of smartphones and social networks. The described solution allows the establishment of trust and reputation accountability between drivers and passengers, while protecting the privacy of all the users.

B.4.1.1. State of the Art

The first carpooling projects emerged in the late 1980s [240]. However, in those days, without the technology available today, users had to face many difficult obstacles such as the need to develop a user network and convenient means of communication. Gradually, the media used to organize the trips were changing from corded telephone to other more flexible technologies such as the Internet, email and smartphones. Thus, the system started to grow so quickly that in 2009, carpooling represented 43.5 % [78] of all trips in the United States, where 60 % of these trips are fam-pools with family members. Also in Europe, carpooling has become increasingly popular over the past years, thanks to Germany Carpooling and France Blablacar platforms.

Nowadays, many different carpooling platforms and services similar to the aforementioned exist, but even today, they may be considered in their early stages because none of them has reached a critical mass of users. Table B.6 shows several features of a few existing carpooling systems, including the most relevant security-related ones. In particular, we have chosen for this comparative analysis the representative systems: Amovens [233], Blablacar [20], CarPooling [91], Compartir.org [54], and ZimRide [257].

BlaBlaCar [20] is the world leading ridesharing service. It is focused on

Platform	Social Network	Privacy	Reputation System	Phone Cert.	Trust Alg.
Amovens [233]	yes	yes	yes	no	no
Blablacar [20]	yes	yes	yes	yes	no
Carpooling [91]	yes	no	yes	no	no
Compartir [54]	no	yes	no	no	no
ZimRide [257]	yes	yes	no	no	no
Proposed System	**yes**	**yes**	**yes**	**no**	**yes**

Tabla B.6: Representative Carpooling Platforms

long distance trips, and uses social networks for registration and feedbacks as a guarantee and an enabler of real connections between users of the service. On the other hand, the biggest carpooling service in the United States is ZimRide [257], where payments are made through credit cards account and PayPal. The main trust enforcing system in all these platforms is based on points given by users. However, the bypass of this security system is quite easy because users who obtain a negative score, can create a new profile with new credentials and no points.

Apart from these practical platforms already in operation, there are several papers that propose different solutions. The work [47] shows an integrated system for the organization of carpooling service by using different technologies such as web, GIS and SMS. The authors of [217] propose a web platform to carpool. The paper [81] presents a carpooling architecture that uses a credit mechanism to encourage cooperation between users.

A more recent work is [52], where an algorithm to encourage carpooling is proposed based on assigning priority to users with positive feedbacks through a fuzzy logic scheme. Another paper, [86], defines a push service to promote carpooling through instant processing. Finally, another interesting proposal is [26], based on a secure multi-agent platform that focuses on the security services allowing both the mutual authentication between the users and the application components with the system.

Our work differs from all the aforementioned because it mainly deals with the trust aspect of carpooling services through a combination of reputation measurement with privacy protection.

B.4.1.2. Platform

The main objective of the proposed design is the increase of both usability and security because its key factors are user-friendliness and privacy.

One of the main features of the proposal is that those users who publish their trips have their privacy fully protected. Unlike other carpooling platforms, in the described system, no user is allowed to access data such as email, phone or full name of others, unless it is authenticated on the platform and the algorithm for checking mutual trust returns a valid permission. In this case, the interested user can see all the data in full detail. Otherwise, it

can only send a request so that the receiver can decide whether the applicant is to be trusted or not.

The algorithm is based on trust relationships so that people who want to use the platform first need to authenticate in the platform through social networks such as Facebook, Twitter or Google+. In this way, the algorithm checks the existence of some chain of trust between the applicant and other users, based on the so-called rule of six degrees of separation [250], which is the theory that everyone is six or fewer steps away from each other in the world so that a chain of a friend of a friend statements can be made to connect any two people in a maximum of six steps.

Besides, the reputation gained through the use of the application is an influent factor, which is considered in the decision on whether carpooling with another person. To do this, at the end of every shared travel, the application asks both drivers and passengers to score the other users. Such scores are used in future trips so that seats offered by car drivers with good scores appear in better positions than others with lower scores. Also well-scored passengers have higher probability to have access to more details of drivers.

The proposed system architecture uses an application model known as client-server (see Figure B.18). Its different elements are the following:

- **Client**: Mobile device used for the system.
- **Server**: Hosted in the cloud, and divided into two parts. On the one hand, the GCM server is the Google Cloud Messaging server that handles all the notifications and is responsible for sending the notification when the receiver clients are alive. On the other hand, the DB dedicated server is the server that stores in its DataBase all the data related to the users and system. It also serves as a gateway for sending notifications between the client and the GCM server.

The proposed scheme protects user privacy through limited and controlled access to user data, according to the trust level stated for the relationship between each pair of users. This trust level is got through the combination of direct scores and trust networks so that it provides the system with enough data to deduce whether people can trust each other or not. In this way, privacy is dealt with as one of the most important aspects of the proposed carpooling system.

A first approach to the development of a trust measurement algorithm that provides a value to each pair of users is based on the use of the PageRank algorithm to predict whether two people can rely each other. However, since this algorithm does not conform totally to the morphology of the specific problem, a second approach is also being used to complement it, based on Bayesian networks to know whether people can trust others. The refined algorithm for trust measurement is available in the Android application.

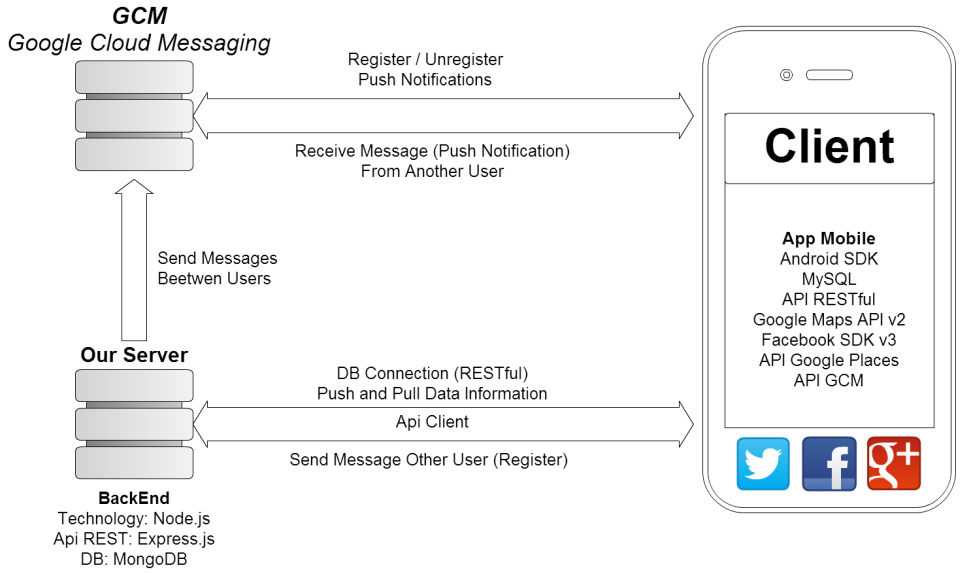

Figura B.18: Carpooling System Architecture

B.4.1.3. The Reputation Algorithm

No existing carpooling system offers the user a quantitative method that can be used to decide whether another user is trustful to share a vehicle or not. The proposal offers this feature based on the theory of six degrees of separation. Some of the proposals even do not allow users to decide who may or may not apply your route. There are some proposed that the quantitative method based on the similarities among users [50] [205]. Their main problem is that they have to collect information about the attributes and characteristics of each user. Our proposal aims to be simple for the user, so that he/she has not to fill out any information about his/her attributes. With a simple click to enable social login on the network after a previous registration, the user may enter the system and start using the platform.

The main feature of our scheme is the proposed reputation algorithm because thanks to it, the user can use a quantitative measure to decide whether trusting another user. The algorithm is based on the theory of six degrees of separation and individual scores within our platform. This number of steps may be reduced significantly by introducing the concept of social networks. Our application uses social networks when logging into the application to create network users to be used to interconnect with each other and provide a reliable measure of confidence. Through the use of social networks we can ensure that the six degrees of separation are reduced to only four. In particular, according to several research studies on Facebook [63] [248] [11], the obtained average distance was 3.9, corresponding to intermediaries or degrees of separation, what shows that the world is even smaller than

expected.

The reputation measurement is a value between 0 and 1, but is shown to the user as a character (S, A, B, C, D), and is computed from the values given by each pair of users to inform about the reliability in each other. This measure is calculated by taking into account the two parameters mentioned above: degree of friendship and appreciation of other users of the platform.

The social network is here represented by a directed weighted graph $G = (V, E)$ where $|V| = n$, $V = \{v_1, v_2, ..., v_n\}$, $|E| = m$, $e_{ij} \in E$, $e_{ij} = v_i \rightarrow v_j$ if v_i is adjacent to v_j for $1 \leq i \leq n$ and $1 \leq j \leq n$, $v_i \in V$, $v_j \in V$. On the one hand, $A = [a_{ij}]_{n \times n}$ refers to the adjacency matrix where $a_{ij} = 1$ if $e_{ij} \in E$, and $a_{ij} = 0$ otherwise. On the other hand, $P = [p_{ij}]_{n \times n}$ refers to a matrix representing degrees of friendship, as defined by equation B.7.

$$p_{ij} = \begin{cases} \frac{\sum_k (nA_{ij}^k \cdot wA_{ij}^k)}{MFC}, & if e_{ij} \in E \\ \\ 0, & otherwise \end{cases} \tag{B.7}$$

where:

- nA_{ij}^k is the number of repetitions of the k Friendship Action, which may be a comment, a like, or any other action between two users of a social network.
- wA_{ij}^k is the weight of the k Friendship Action. It represents the importance of the Action over others. Its value is between -0.25 and 1. For example, with the Facebook data, the weight of a comment is 0,727, and the weight of a like is 0,273. These values are obtained from the data shown in [15], where it is concluded that for every comment there are 2.667 likes. Thus, a comment is more important than a like. The system is able to detect positive, negative and neutral comments, using machine learning algorithms [247].
- MFC is called the Maximal Friendship Coefficient and represents the maximum weight of possible positive actions between any pair of users. Its value is: $max(\sum_k (nA_{ij}^k \cdot wA_{ij}^k))$.

Users compute scores from assessments after sharing routes. When a route is completed, the users who participated in it, can vote between 1 and 5 stars. Each passenger individually assesses the driver, and the driver individually assesses passengers. The weight is higher from driver to passengers than from passengers to driver, as the driver puts his/her vehicle available to the users. In order to account for the different ratings on a user, a simple arithmetic average is used. The metric taken into account for these ratings is as shown in Table B.7. It has been estimated as neutral value 3 stars rating. 4 or 5 stars is considered positive rating and 1 or 2 is negative rating. The

drivers have a higher impact factor in the neutral rating, causing less impact on the negative rating and higher impact in the positive rating. The equivalence between star rating and point rating is based on the study published in [252]. The main basis of such a system is a definition about good and bad behaviours. For example, 2 stars is a bad score, while 3 stars is an acceptable score. Thus, the difference between 2 and 3 stars is more relevant than the difference between 3 and 4 stars, as it involves the transition from a bad score to an acceptable score. If a user scores with 2 stars, it means that the journey was bad. However, if a user scores with 3 or 4 stars, it is a sign that the ride was not bad, so the point score also reflects that fact.

Star Rating	**Driver Rating Impact**	**Passenger Rating Impact**
1 star	-0.5 points	-0.75 points
2 stars	-0.25 point	-0.5 points
3 stars	0.5 points	0 points
4 stars	0.75 points	0.5 points
5 stars	1 points	0.75 points

Tabla B.7: Impact of Ratings

The final metric rating of the assessments is given by equation B.8.

$$MeanR_j = \frac{\sum_i (PR_{ij} + NR_{ij})}{nR_j} \tag{B.8}$$

where:

- PR_{ij} is the total Positive Rating received by user j from user i. It corresponds to the ratings with 3 stars or more.
- NR_{ij} is the total Negative Rating received by user j from user i. It corresponds to the ratings with 1 or 2 stars.
- nR_j is the total number of Ratings received by user j.

The metric that indicates the degree of reliability of a user i over another user j is called Trust Rate, and described in equation B.9.

$$TR_{ij} = \begin{cases} DT_{ij}, & \text{if both users are direct friends.} \\ IT_{ij}, & \text{if a chain of friendship between both users exists.} \\ NT_{ij}, & \text{if no chain of friendship between both users exists.} \end{cases} \tag{B.9}$$

where:

- If the user j is a direct friend of the user i, Direct Trust obtained using the equation B.10 is used.

$$DT_{ij} = p_{ij} \cdot wP + MeanR_j \cdot wR \tag{B.10}$$

- If the user j shares a chain of friends with user i, Indirect Trust obtained using the equation B.11 is used:

$$IT_{ij} = (\prod_{(a,b)\in chain} p_{ab}) \cdot wP + MeanR_j \cdot wR \tag{B.11}$$

- If no chain of friendship between user i and user j, the equation B.12 applies.

$$NT_{ij} = MeanR_j \cdot wR \tag{B.12}$$

where:

- wP is the weight of Friendship. Its value (0,625) is given by the average happiness of people with their friends in Facebook social network [252].
- wR is the weight of Rating. Its value is $(1 - wP)$, i.e. 0,375.
- $\prod_{(a,b)\in chain} p_{ab}$ represents the degree of friendship between two users joined by a chain.

The relationship between the Trust Rate (TR) metric and the value provided to the user, is shown in Table B.8.

Trust Rate	**Character Shown to Users**
[-0.25, 0)	D
[0, 0.25)	C
[0.25, 0.5)	B
[0.5, 0.75)	A
[0.75, 1]	S

Tabla B.8: Equivalence between Trust Rate and Shown Character

The maximum reputation metric (TR) that a user can receive is 1 (represented as S score for the user), which corresponds to the situation when the users direct friends have rated it with the highest scores.

A user can have total null valuation in the following situations: it is starting to be known, it does not have any degree of friendship, and/or it has negative reviews.

This valuation is dynamically calculated as a function of the friendship degree that a user has got. It helps users to have a reliability measure about

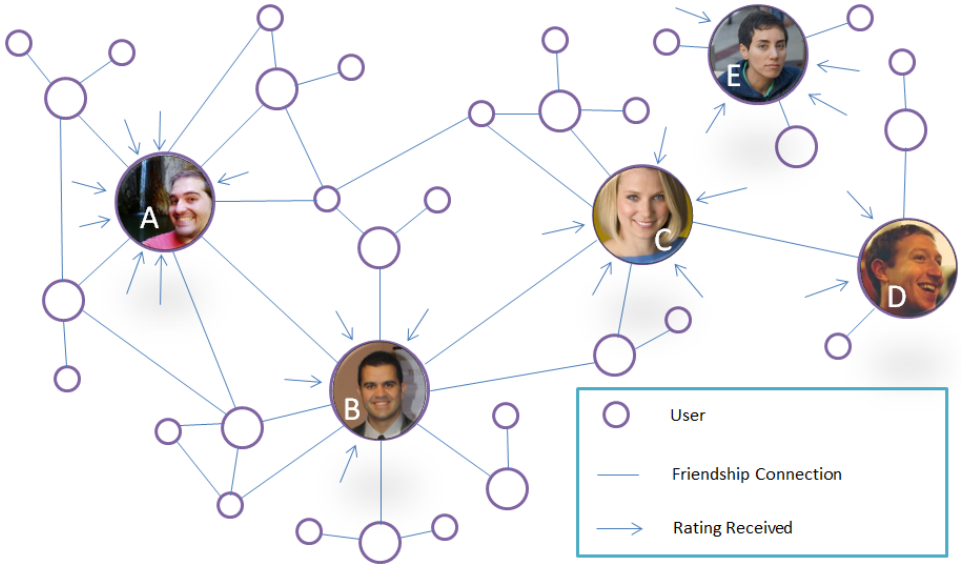

Figura B.19: Example of Network (Picture from Facebook Public Profiles)

whether to trust another user of the platform. Besides, only users who have a valuation higher than B and/or users who have been accepted by the driver to make the route can see certain route data, such as the phone number or any other confidential data.

The system threshold, defined as reliability value, is set to a value higher than B (i.e. A or S). This choice is because this is the minimum value required to represent a direct friendship.

B.4.1.4. Implementation Analysis

The algorithm is executed in parallel using different threads to optimize efficiency. For the calculation of IT value, where two users are not direct friends but they have a chain of friendship, each partial DT value, is calculated in a different thread. A typical network of the proposed system is shown in Figure B.19.

In order to discern whether an action taken on social network is positive or negative, the proposed scheme uses AlchemyAPI, which by Machine Learning algorithms provides the social sentiment of a text. The proposal only takes into account the latest 100 interactions between users on social networks.

Each TR value is variable or dynamic for each pair of users. It depends on the degree of friendship between the pair of users and their interactions in social networks. Furthermore, this value is continuously updated because social actions among users constantly change in social networks, and as time

goes by there is more feedback and ratings of the user within the system.

In order to analyse the operation of the proposed reputation algorithm, Table B.9 provides a sample of parameters generated using the application, considering the network of Figure B.19.

RelationShip	**Data about RelationShip**	
User A - User B	Likes in Facebook	78
	Comments in Facebook (positives - negatives)	19
	Friendship Rate	*0.860*
User B - User A	Likes in Facebook	82
	Comments in Facebook (positives - negatives)	12
	Friendship Rate	*0.761*
User B - User C	Likes in Facebook	53
	Comments in Facebook (positives - negatives)	8
	Friendship Rate	*0.497*
User C - User B	Likes in Facebook	4
	Comments in Facebook (positives - negatives)	2
	Friendship Rate	*0.061*
User C - User D	Likes in Facebook	76
	Comments in Facebook (positives - negatives)	21
	Friendship Rate	*0.882*
User D - User C	Likes in Facebook	42
	Comments in Facebook (positives - negatives)	7
	Friendship Rate	*0.405*

Tabla B.9: Sample Friendship Data with a Global $MFC = 40{,}831$

The next Table shows the ratings received by users within the application through the completed routes.

Through the data of Figure B.19 and the previous Table, and considering 40,831 as MFC value for every user, trust rates have been obtained in the next Table using the described algorithm:

For example, to calculate the trust rate to the user A in the system, about his/her relationship with the user B, the calculations shown in expression B.13 are carried out.

$$DT_{AB} = \frac{(19 \cdot 0{,}727) + (78 \cdot 0{,}273)}{40{,}831} \cdot 0{,}625 +$$

	User A	User B	User C	User D	User E
Received Ratings as Driver	4 stars 5 stars	3 stars 4 stars 4 stars	5 stars	-	2 stars 3 stars
Received Ratings as Passenger	4 stars 4 stars 5 stars 4 stars	5 stars 4 stars 2 stars	3 stars 4 stars 3 stars	2 stars 3 stars 2 stars	3 stars 1 star 2 stars

Tabla B.10: Sample Ratings

	User A	User B	User C	User D	User E
User A	-	0.704 (A Score)	0.392 (B Score)	0.152 (C Score)	-0.075 (D Score)
User B	0.690 (A Score)	-	0.435 (B Score)	0.191 (C Score)	-0.075 (D Score)
User C	0.242 (C Score)	0.211 (C Score)	-	0.468 (B Score)	-0.075 (D Score)
User D	0.224 (C Score)	0.187 (C Score)	0.378 (B Score)	-	-0,075 (D Score)
User E	0.213 (C Score)	0,172 (C Score)	0.125 (C Score)	-0,083 (D Score)	-

Tabla B.11: Trust Rates Sample

$$
\begin{aligned}
&+\frac{0{,}5+0{,}75+0{,}666+0{,}333-0{,}333}{6}\cdot 0{,}375 = \\
&= \frac{35{,}107}{40{,}831}\cdot 0{,}625+\frac{2{,}66}{6}\cdot 0{,}375 = \\
&= 0{,}537+0{,}167 = 0{,}704
\end{aligned}
\tag{B.13}
$$

In order to calculate the trust rate to the user D in the system, about his/her relationship with the user A, the calculations shown in expression B.14 are carried out.

$$
\begin{aligned}
IT_{DA} &= p_{DC}\cdot p_{CB}\cdot p_{BA}\cdot wP + \\
&+MeanR_A\cdot wR = \\
&= (0{,}405\cdot 0{,}061\cdot 0{,}761)\cdot 0{,}625 + \\
&+\frac{0{,}75+1+0{,}33+0{,}33+0{,}66+0{,}33}{6}\cdot 0{,}375 = \\
&= 0{,}019\cdot 0{,}625+\frac{3{,}41}{6}\cdot 0{,}375 =
\end{aligned}
$$

$$= 0{,}012 + 0{,}212 = 0{,}224 \tag{B.14}$$

Therefore, the proposed system can be considered fully dynamic and depending on the relationships between each pair of users. Figure B.20 shows a comparison of the trust rates in each pair of users and the normal rating taking other proposals based only on the rating from users. In particular, Figure B.20 shows the level of trust of the Axis Y User on the Axis X User.

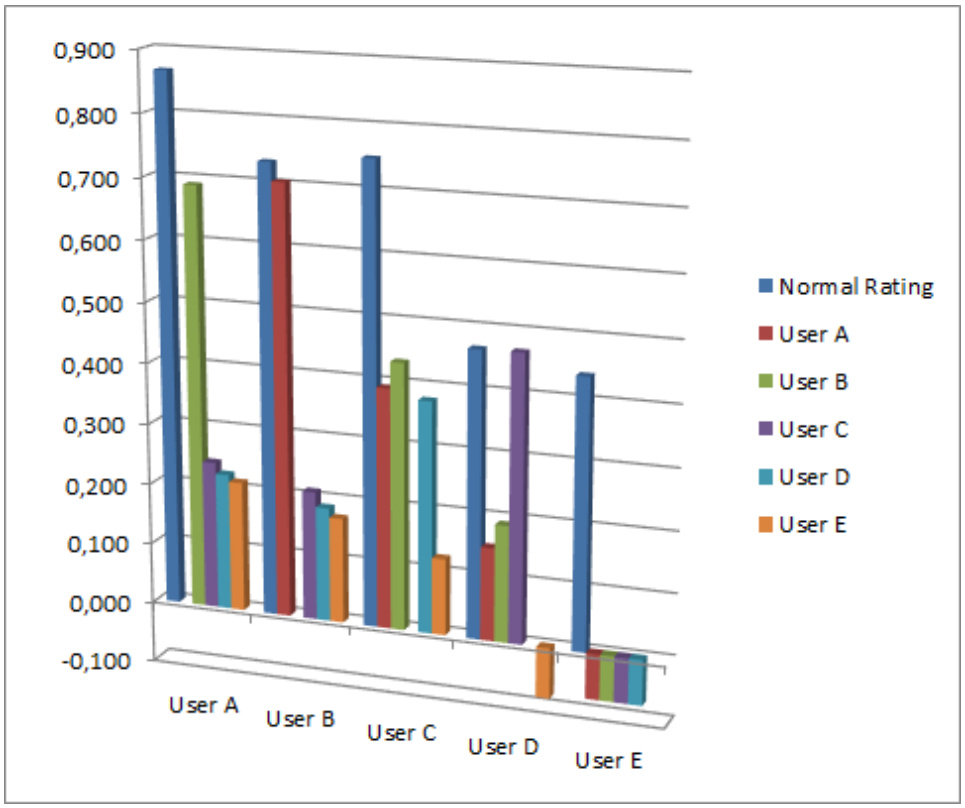

Figura B.20: Proposed Dynamic Rating System Versus Classical Rating

B.4.1.5. Security of the Scheme

Regarding the safety of the platform, the motivations of a malicious user may be different. On the one hand, the main reason for a regular attacker can be the satisfaction of breaking a system that is considered safe. Another motivation may be to access sensitive information or getting data privacy for fraudulent use. In addition, for this particular scheme, a user may want to increase their social status in the system increasing his/her score. Finally, one of the reasons that an attacker may have is to denigrate a user modifying his/her score with low ratings. In this way, a user can become a bad user to share route, and his/her participation gets unfeasible in many routes.

The Sybil attack is a notorious attack in traditional carpooling systems. These types of attacks are hacking attacks on peer-to-peer networks where a malicious device illegitimately takes multiple identities by forging them. Due to the privacy-preserving environment of carpooling schemes, a Sybil vulnerability is generally hard to defend against.

In a Sybil attack, the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous identities, using them to gain a disproportionately large influence. A reputation system vulnerability to a Sybil attack depends on how cheaply identities can be generated, the degree to which the reputation system accepts inputs from entities that do not have a chain of trust linking them to a trusted entity, and whether the reputation system treats all entities identically.

An entity on the analysed peer-to-peer network is a piece of software that has access to local resources. It advertises itself on the peer-to-peer network by presenting an identity. More than one identity can correspond to a single entity. In other words, the mapping of identities to entities is many to one. Entities in peer-to-peer networks use multiple identities for purposes of redundancy, resource sharing, reliability and integrity. In peer-to-peer networks, the identity is used as an abstraction so that a remote entity can be aware of identities without necessarily knowing the correspondence of identities to local entities. By default, each distinct identity is usually assumed to correspond to a distinct local entity. In fact, many identities may correspond to the same local entity.

A dishonest member or an adversary node may present multiple identities to a peer-to-peer network in order to appear and function as multiple distinct nodes. After becoming part of the peer-to-peer network, the adversary may then overhear communications or act maliciously. By masquerading and presenting multiple identities, the adversary can control the network substantially.

Let us consider, for example, the following scenario. A bad user (B0) sets up several bogus accounts in social media and the proposed system (B1, B2, B3, etc.). He/she then advertises a possible trip from $X \mapsto Y \mapsto Z$. For the first leg ($X \mapsto Y$) he/she uses the bogus accounts to claim that his/her vehicle is full and all these passengers state that they get off at Y. He/she can then get excellent scores and increase his/her reputation. At some point, this will be so high that other normal users will be able trust him/her from leg $Y \mapsto Z$.

The method of reputation is based on trust between friends, so that if a user is able to be friend of another user by the Social Network System, it is very difficult to get fooled because he/she must get relevant comments and likes on his/her publications, from the user to which he/she wants to cheat. Thus, although an adversary tries to cheat on Social Network System, a deep bond of friendship is necessary because the system feeds on the comments

and likes between two users to calculate his/her rating

The proposed algorithm reduces, significantly, the vulnerability to the attack described above because most of the score of our algorithm is preceded by confidence in degrees of friendship that binds each user to another user. Thus, if a user does not know (at all) another user, very high ratings of the latter in the system are not reliable enough for the former. It is remarkable that in our system the weight of the user rating is 37.5 % of the total trust rate, while the reliability of friendship is 62.5 %.

B.4.1.6. The Android Application

Although the system described in this work can be combined with existing carpooling platforms, the proposed design has been also embodied in a new Android application that has been already published in the Google Play Store under the name Carpoolap (see Figure B.21).

The Android application Carpoolap has been developed for the versions 3.0 or higher of the operating system. APIs like Google Maps v3.0, Google Places, Google Cloud Messaging, etc., and Facebook SDKs 3.0 and libraries like Action Bar Sherlock were used to add the novel functionalities of the new versions of Android on older versions. Thus, autocomplete in address searches, Google Maps 3D Technology, design based on the latest versions of Android, push notifications with requests or responses of passengers or drivers, etc. are among the main features of the Android Application Carpoolap.

Each user can see the routes he/she proposes as driver, and whether potential passengers exist for those routes. Besides, with colour codes, he/she can know the routes that each user has already made and the routes that have been confirmed by users. For the assessment of users participating in a route, after finishing it, each one can give a score. In order to deploy the carpooling platform, a server is also needed, so we developed one using JavaScript technologies by frameworks like node.js and express.js. As a database for all the data centralized on this server, we decided to adopt a No SQL database, such as MongoDB [190]. We deployed our server on a micro instance of Amazon Web Services, specifically under Ubuntu machine with Amazon EC2 account.

The application has been installed by more than 5000 users worldwide, and generated more than 1000 different journeys in its lifetime. The feedback received from users has been positive with an average score of more than 4 stars out of 5 on Google Play Store. Integration with other social networks would be positive for the ratings and integration with other platforms including carpooling apps. It would be ideal to have greater access to the Facebook API to improve the reputation algorithm with other internal parameters generated in this social network, but is currently closed to outside developers.

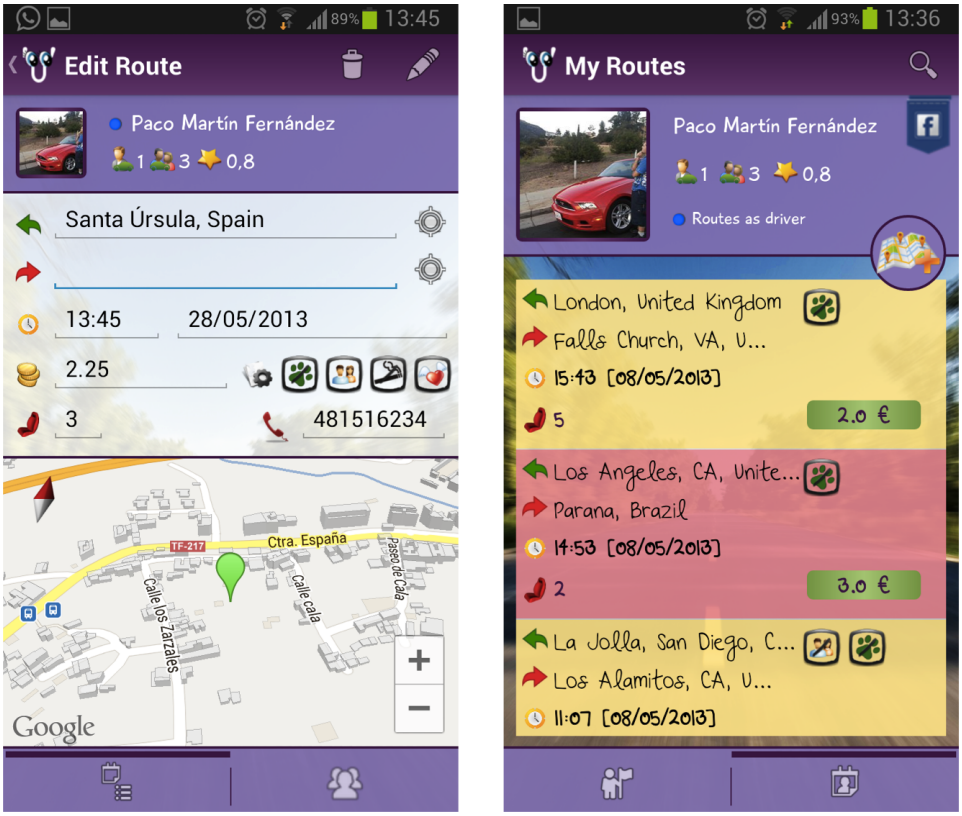

Figura B.21: Carpoolap Screens: Route Edition & Routes List

B.4.2. Traffic Light Violations

Problems related to road traffic have increased worldwide as a result of population growth, both in big urbanized zones and dense populated areas. These adverse traffic circumstances can reduce the efficiency of transport infrastructure and increase travel time, fuel consumption and pollution. One of the consequences of the widespread of Information and Communication Technologies (ICT) is the existence of countless applications that help to drive.

Currently, the decisions of drivers depend on what they see and/or hear. However, a system that includes interactive and cooperative driving and effective traffic control would provide a third channel for additional data that cannot be seen or heard by drivers directly, but that could be very helpful for decision-making.

The so-called Intelligent Transport System (ITS) is a set of technological solutions designed to optimize different modes of transport. ITs main goals are to improve passenger comfort, increase travel safety, mitigate traffic congestion and reduce fuel consumption. In order to achieve it, vehicles and

infrastructure have to cooperate. Since cooperation is only possible if the involved entities can communicate, wireless communications are required. Besides, ITS is based on different information technologies, sensors and the Internet.

A particularly difficult problem in road safety is the detection of users who do not stop at red traffic lights. There may be several causes of traffic light violations. One of them is the duration of the traffic lights because traffic lights with a very short duration can cause that users ignore the red light, what can produce a ripple effect that can cause many accidents. In order to try to address the problem of red light running, different solutions have been proposed such as new traffic signal mechanisms, red-light speed cameras to detect offenders, etc., which reduce traffic jams in urban centres around the world. These solutions are effective but very expensive to be widespread.

A study [214] found that urban crashes involving drivers who ran red lights, stop signs and other traffic controls were the most common type of crash. In particular, according to a report of the National Highway Traffic Safety Administration (NHTSA) [198], there were more than 2.3 million reported intersection-related crashes, resulting in more than 7,770 fatalities and approximately 733,000 injury crashes in the USA. The Fatality Analysis Reporting System of the NHTSA states that red light running crashes alone caused 762 annual deaths, and that 165,000 people are injured annually by red light runners. Besides, the Insurance Institute for Highway Safety (IIHS) [83] reports that half of the people killed in red light running crashes are not the signal violators, but drivers and pedestrians hit by red light runners.

B.4.2.1. Related Works

The need of improving road traffic management is evident worldwide. Governments are worried about the growing number of vehicles on roads and of traffic-related deaths. For this reason, they are trying to improve traffic safety by exploring the potential of the ITS through numerous research projects. The current ITS state-of-the-art is based primarily on a series of initiatives from both academia and industry, addressed mainly to try to enable the future development of VANETs.

Regarding countermeasures implemented in practice, different solutions to try to reduce red light running have been tested. For instance, there have been proposed [213] that signalized intersections are replaced by roundabouts, that more adequate yellow signal time is provided, and/or that a brief phase when all signals are red is added. However, results show that none of these measures eliminate the need for novel solutions to the problem of red light running.

One of the possible applications of ITS to provide a solution to the red light running problem are intelligent traffic lights, which are responsive to

pedestrians needs so that, for instance, green light duration changes for blind pedestrians. Besides, intelligent traffic lights can also be self-controllable to maximize flow in the route. They can be also used to punish those users that violate traffic lights [64], [51]. This solution, called red light camera, has been operating for several years in many different regions around the world [24]. A red light camera is a type of traffic enforcement camera that automatically captures an image of any vehicle that enters an intersection after jumping a red traffic light and sends this photo to the traffic signal control system so that the photo can be used as evidence that assists authorities in their enforcement of traffic laws. Generally, the camera is triggered when a vehicle enters the intersection (passes the stop-bar) after the traffic signal has turned red.

In [99], the authors present an adaptive traffic light system based on wireless communication between vehicles and fixed controller nodes deployed in intersections. Such traffic light system is based on short-range wireless communication between vehicles, which uses a controller wireless node placed in the intersection that determines optimum values for the traffic lights phases.

Google [77] presents several methods for automatically mapping the three-dimensional positions of traffic lights and robustly detecting traffic light state on board equipment in cars with cameras. They used these methods to map more than four thousand traffic lights, and to perform on board traffic light detection for thousands of drivers through intersections.

The work [2] proposes the use of RFID for dynamic traffic light sequences to avoid problems that usually arise with systems that use image processing and beam interruption techniques. RFID technology was applied to a multi-vehicle, multi-lane and multi-road junction area to provide an efficient time management scheme. A dynamic time schedule was worked out for the passage of each column. The conclusion was that the system could emulate the judgment of a traffic police officer on duty.

A modern traffic light for six roads and four junctions has been implemented by programming in the PIC16F877A microcontroller [124]. The system works efficiently over the present traffic controlling system with respect to less waiting time, efficient operation during emergency mode and suggestions of alternate route.

The main goal of this work is to propose a system to detect traffic light offences in order to warn nearby drivers and pedestrians, and prevent possible accidents. To the best of our knowledge, there is no proposal to notify the vehicles in an area where there is a nearby vehicle that has jumped a traffic light. Nor is there a solution allowing a vehicle to report that it has broken the law at a traffic light, anonymously. Anonymity is necessary in our system to encourage using it. The authorities can also benefit by analysing data generated by the system, in order to detect whether a traffic light is more

likely to be violated than another one and to optimize the timing of traffic lights.

B.4.2.2. System Operation

The proposal works as explained below. When a vehicle approaches a traffic light, it receives a beacon with the colour of the traffic light, and if it is in red, the driver is warned about that. Then, if it does not stop, the driver smartphone detects it and sends a warning to other nearby vehicles so that the other drivers receive a warning about the dangerous situation through their smartphones.

The implemented system uses sensors, smartphones and cloud servers to automatically detect and anonymously report that a driver has failed to respect a traffic light. Figure B.22 shows an overview of the system operation.

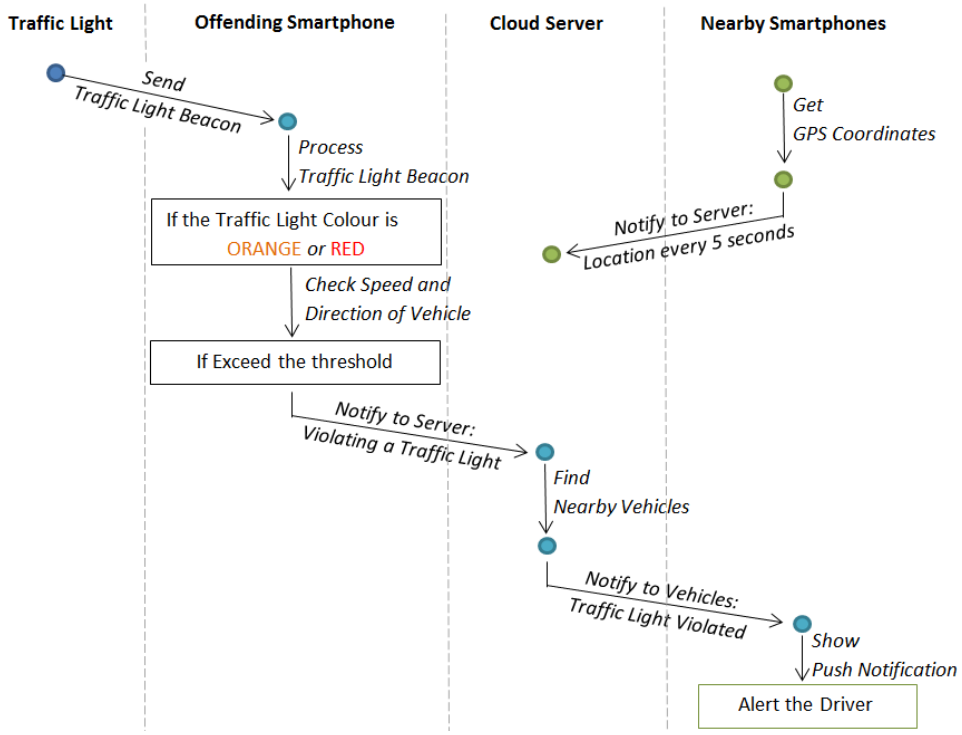

Figura B.22: Overview of the System Operation

As it can be seen in the scheme, each traffic light is continuously sending beacons with its colour. If it is orange or red, the receiver smartphone inside each car checks speed and direction and if it detects a red light running, it notifies it to server. The server then finds nearby vehicles and sends a push notification to the smartphones inside them in order to alert their drivers.

The main technology tools used in the proposed system are:

- Light sensors, to provide information in real time about traffic light colour.
- Bluetooth Low Energy (BLE) modules, to allow transmitting continuously the state of the traffic lights to nearby vehicles, as beacon notifications.
- Arduino-compatible devices, to allow sending and receiving data via BLE.

Thus, the proposed system is based on I2V and V2I communications, and is mainly formed by:

- A sensor platform: located in traffic lights, communicates with nearby smartphones. smartphones: used to identify the vehicles
- A cloud server: responsible to notify the other nearby vehicles about danger warnings.

As aforementioned, the system can be also used by road authorities to detect traffic lights that are less respected. Thus, an action plan can be established to investigate the causes for searching solutions (longer duration of light colour, etc.).

B.4.2.3. Security Scheme

The proposed system protects user anonymity, and integrity and authenticity of information, in order to promote the application to be used. The aim is not to find the users who skip the traffic lights, but to warn above that a user has jumped a traffic light, without being able to trace his/her identity. Therefore, a reliable and secure anonymity scheme is needed to inspire confidence to all users. The proposal uses a cloud server, a sensor platform and smartphones to achieve this aim. The smartphones are used to identify the vehicles. The sensor platform is located in traffic lights and communicates with the smartphones. The cloud server is responsible to notify to the other nearby vehicles.

In order to maintain this level of security, OpenSSL was used in the implementation. OpenSSL is an open-source implementation of the SSL and TLS protocols. OpenSSL supports a number of different cryptographic algorithms. In particular, this work uses the version 1.0.2 released in January 2016.

For the establishment of a secure communication channel, a Certificate Authority (CA) has been implemented in the cloud server. A certificate authority is an entity that issues digital certificates to certify the ownership of a public key. This allows others to rely upon signatures or on assertions made by the private key that corresponds to the certified public key. In this

model of trust relationships, a CA is a trusted third party, trusted both by the subject (owner) of the certificate and by the party relying upon the certificate.

The integrity of the message and the authenticity of the sender are protected through the use of a digital signature scheme. Thus, the vehicle uses its private key during the process of digital signature of the message sent to the server, and the server uses the user public key to verify the digital signature of the message. Specifically, the scheme is based on the Elliptic Curve Digital Signature Algorithm (ECDSA) [119] that offers a variant of the Digital Signature Algorithm (DSA), which uses elliptic curve cryptography.

The implementation is based on a digital signature scheme with the following parameters, where $\times$ denotes elliptic curve point multiplication:

- *Curve*: Equation defining an elliptic curve field.
- G: Elliptic curve base point, generator of the *Curve* with prime order n.
- n: Integer order of G, so that $n \times G = O$.
- d_A: Private key integer randomly selected in the interval $[1, n-1]$.
- Q_A: Public key curve point denoted by $Q_A = d_A \times G$.
- m: Message to sign.

On the one hand, Algorithm 1 is used to sign a message m.

Algorithm 15: Signature Algorithm

1 Calculate $e = h(m)$, where $h(\cdots)$ is the SHA-3 cryptographic hash function;
2 Let z be the L_n leftmost bits of e, where L_n is the bit length of n;
3 Select a cryptographically secure random integer k from $[1, n-1]$;
4 Calculate the curve point $(x_1, y_1) = k \times G$;
5 Calculate $r = x_1 \bmod n$. If $r = 0$, go back to step 3;
6 Calculate $s = k^{-1}(z + rd_A) \bmod n$. If $s = 0$, go back to step 3;
7 The signature is the pair (r, s);

On the other hand, Algorithm 2 allows the server to verify each signature.

In order to protect user anonymity, k-anonymity is used for the digital signature. The concept of k-anonymity was first formulated in [238] as an attempt to solve the problem that given person-specific field-structured data, produce a data release with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remain practically useful.

Algorithm 16: Verification Algorithm

1 Check that Q_A is not equal to the identity element O;
2 Check that Q_A lies on the curve;
3 Check that $n \times Q_A = O$;
4 Verify that r and s are integers in $[1, n-1]$. Otherwise, the signature is invalid;
5 Calculate $e = h(m)$, where $h(\cdots)$ is the same function used in the signature generation, SHA-3;
6 Let z be the L_n leftmost bits of e;
7 Calculate $w = s^{-1}$ mód n;
8 Calculate $u_1 = zw$ mód n and $u_2 = rw$ mód n;
9 Calculate the curve point $(x_1, y_1) = u_1 \times G + u_2 \times Q_A$;
10 The signature is valid if $r \equiv x_1$ mód n. Otherwise it is invalid;

In particular, a release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least $k-1$ individuals whose information also appear in the release.

The implemented scheme guarantees k-anonymity through the application of the ideas in [34], according to vehicle every user is randomly associated to a group that share cryptographic material such as a par of privates pubic keys and a group certificate so that this data are used to sign. In this way, users do not reveal their particular identities but only their group identifier.

B.4.2.4. Sensor Platform

Sensing systems for ITS are based on networked system vehicles and infrastructures, i.e. on smart vehicle technologies. Infrastructure sensors are in general tough devices that are installed in the road. These sensors may be disseminated during road construction or by sensor injection machinery for rapid deployment. There are many types of sensors: vehicle counters, weather stations, cameras to detect traffic jams, radars to detect high speeds, etc. These sensors can be ranged from very simple (such as sensors to detect the number of vehicles on a road section) to highly advanced (such as cameras to detect vehicles with a special software). Usually, the more complex sensors are the most expensive. A camera with visual detection of vehicles is a very expensive system, and it is used to avoid the violations of traffic lights.

In order to add intelligence to traffic lights, the proposed system uses a light sensor that provides information in real time about the traffic light colour. This, together with a Bluetooth Low Energy (BLE) module, allows transmitting the state of the traffic light to nearby vehicles, as a beacon notification.

Bluetooth road sensors are able to detect Bluetooth MAC addresses from Bluetooth devices in passing vehicles. If these sensors are interconnected, they are able to provide interesting data. Compared to other traffic measurement technologies, Bluetooth measurement has some differences:

- High Accuracy and the devices are quick to set up easily.
- Limited to a number of Bluetooth devices that can be broadcasting in a vehicle so counting and other applications are limited.
- Non-intrusive measurements what can lead to lower-cost installations for both permanent and temporary sites.

The sensor platform that is used consists of several electronic modules for composing a small, fully integrated system in any type of traffic light.

In this work, RFDuino [215], which is an Arduino shrunk to the size of a fingertip and made it wireless, is used as the board, exactly the 2216 model, with a Dual AAA Battery Shield. The shield has a step-up switching regulator that allows the batteries to be drained down to low voltages while still providing a stable 3.3V to the RFduino.

The Bluetooth Low Energy module used for the RFDuino is the RFD22102 RFduino DIP. This module has the technical specifications shown in Table B.12.

The format of a BLE message include a 1 byte preamble, 4 byte access codes correlated with the RF channel number used, a Packet Data Unit (PDU) that can be between 2 to 39 bytes and 3 bytes of CRC. Thus, the shortest packet would have 10 bytes and the longest packet would have 47 bytes. The transmission times of these packages range from 8 microseconds to the smallest package up to 300 milliseconds for the largest. The PDU for the advertising channel consists of the 16-bit PDU header, and depending on the type of advertising, the device address and up to 31 bytes of information. Also, the active scanner may request up to 31 bytes of additional information from the advertiser if the advertising mode allows such an operation. It means that a sizeable portion of data can be received from the advertising device even without establishing a connection. Advertising intervals can be set in a range of 20 ms to 10 s. It specifies the interval between consecutive advertising packets.

The sensor, which is connected to the traffic light, captures its colour and state emitted by a beacon, and constantly sends this information to all vehicles near the traffic lights. To ensure the integrity of each beacon, a digital signature scheme is used.

ISO/IEC 9796-2 [116] scheme 1 based on SHA-1 hash and RSA is applied for the digital signature, because its length is only 22 bytes, so it fulfils the storage requirements of BLE beacons. ISO/IEC 9796-2 is a standard signa-

Specification	Value
Part Number	RFD22102
Category	Bluetooth LE RF Module
Type	Transceiver / Controller
Band	2.4 GHz
CPU	16MHz ARM Cortex-M0
Flash	128kb
Ram	8kb
Multi Frequency	Yes
Package-Case	DIP - RFduino Footprint
Packaging	Bulk Clamshell
RoHS Compliant	Yes
Low Supply Voltage	1.9V
Typical Supply Voltage	3V
High Supply Voltage	3.6V
Transmit Current	18mA, 4uA ULP
Receive Current	18mA, 4uA ULP
FCC Approved	Yes
IC Approved	Yes
ETSI - CE Tested	Yes
Transmit Power	4dbm

Tabla B.12: RFD22102 BLE Technical Specs

ture scheme widely used in the smart card industry for public key certificates and message authentication because it quite simple to implement.

All traffic lights have a generic certificate to sign beacons, given by the CA of the Directorate General of Traffic.

The beacon is formatted as shown in Figure B.23, where:

- idTrafficLight: Unique identifier for the traffic light.
- bearing: Compass direction used to describe the direction of the traffic light (represented in degrees (0-360)).
- state: State of the traffic light (green, red, etc.)
- signature: Digital signature of the message.

The beacon is received by the smartphone, which is responsible for processing information and report anonymously if the traffic signal is not respected.

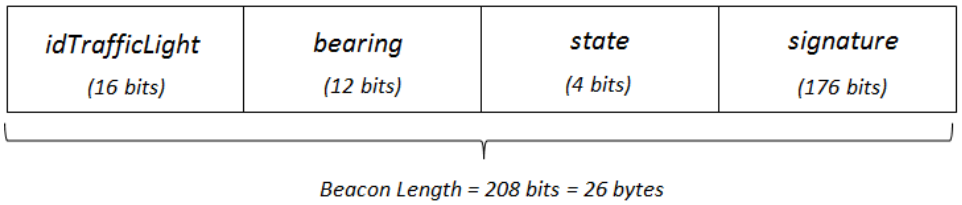

Figura B.23: Format of the Beacon Transmitted by the Traffic Light

B.4.2.5. User Application

A mobile application has been implemented to read the BLE beacon that the traffic light emits, and to process the information (See Figure B.24).

Figura B.24: User Interface of the Mobile Application

In order to monitor all system users and to establish communications, the use of a server, which is responsible for the control and monitoring, is proposed. The used system technologies are shown in Figure B.25.

Depending on the data of the beacon, and the speed that the vehicle has at that time, the application detects in background if the vehicle did not respect the traffic lights. If the vehicle driver violated the traffic light, the smartphone sends a message to a server that controls and manages such events. The server is responsible for searching its database to find nearby

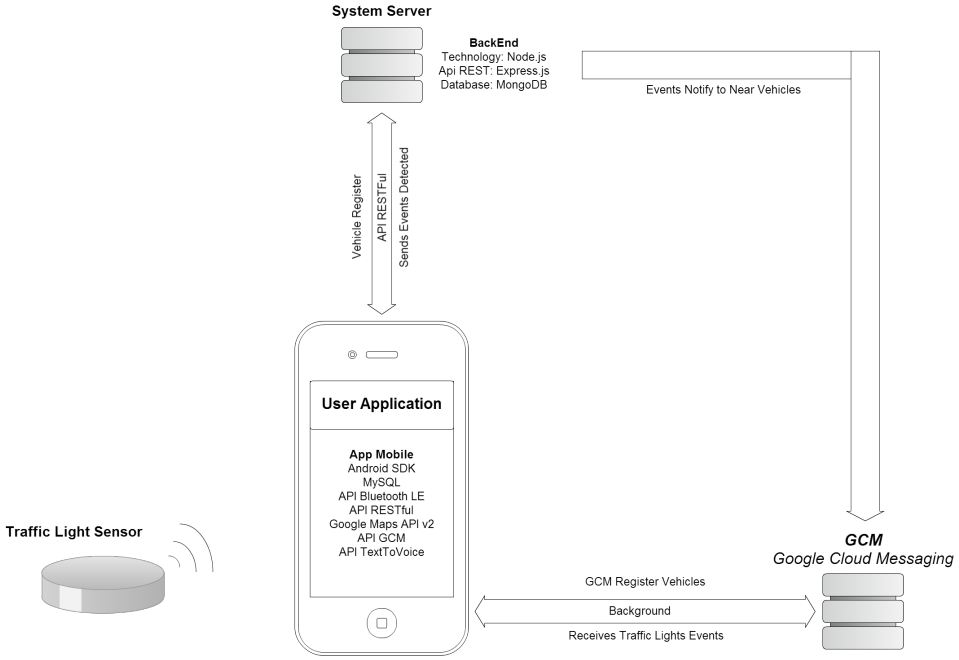

Figura B.25: Use Flow and Technologies used in the System

vehicles at that time. This is possible because all vehicles send every 5 seconds their current positions to the server. Once the server has located all vehicles near the traffic lights, a push notification is sent to all smartphones of nearby vehicles.

The application then informs the driver via voice that there is another driver who has skipped a traffic light in its area, so it recommends driving with special caution. The application also displays on a map, the position of the traffic light.

The server is a full-stack JavaScript implementation. As cross-platform runtime environment for server-side and networking applications Node.js is used. In order to connect the mobile application with the server through REST Web Services, the work uses Express.js. To store the vast amount of data on the server, a NoSQL database called MongoDB is used.

B.4.2.6. Implementation Analysis

The implemented system uses several processes that send various data packets. In Table B.13, the size of the different data packets used in the proposed system is shown.

Different batteries of tests were used to check the time separately for each scheme component and the total time. The simulations were done using multiple software packages to add credibility and develop a realistic simulation.

Packet	Size (in bits)
Beacon from Traffic Light to Smartphone (via BLE)	208
Data Event from Smartphone to Server (via WiFi/GPRS/3G/LTE)	248
Push Notification from Cloud Server to Smartphones	272

Tabla B.13: Size of Sent Packets

Thus, the scenario that has been used for simulations comprises a real traffic situation in the city of Madrid (Spain) in 2014 [65].

As a result, the times represented by the averages of all tests, shown in Table B.14, prove that efficiency has been achieved.

Component	Time (in ms)
Sending the beacon from the Traffic Light to the Smartphone	0,17
Smartphone data processing	51
Sending event from Smartphone to Cloud Server	116
Cloud Server data processing	104
Notification Push from Cloud Server to Smartphones	142
Total	**413,17**

Tabla B.14: Average Time Required to Send Different Data

After having got promising results with the described beta system, an improved system is now being developed so that red light runners do not only notifies the cloud but also directly other nearby vehicles through V2V communications using Wi-Fi direct technology. The improved system requires more care with security of communications so when an offence is automatically broadcast, not only authenticity of sender and integrity of data is guaranteed, but also freshness of data. Besides, in order to promote the use of the system, drivers privacy is preserved through reversible anonymity so that only fraudulent drivers who cause accidents lose their privacy. Finally, in case of accident, non-repudiation is introduced to be able to provide evidence to prove liability.

In order to guarantee freshness, a timestamp has being added to the signatures. On the other hand, in order to provide reversible anonymity and non-repudiation, the new proposal uses a group signature, which is a digital signature based on a group public key and individual private keys defined by a group manager so that it can be used to sign on behalf of the group and any signature can be verified with the group public key, but only the group manager can identify which member issued a given signature. In our case, the group is the group of vehicles, and the group manager is an official entity like the General Directorate of Traffic. An additional advantage of using group signature is the efficiency of verification because verification of

group signatures is performed with respect to the public key of the entire group.

B.5. Conclusions and Future Works

This Thesis work has been oriented to the proposal of innovative solutions for cryptographic algorithms and their applications in real situations applied to transport scenarios.

It has proposed a new authentication method based on non-interactive zero knowledge proofs. This method has been designed for its use in environments where devices move constantly and at high speeds, like the transport scenarios. The implementation and development of the scheme and the results and comparisons with similar methods are detailed. Thanks to this proposed scheme, it describes a new method to share authenticated information without requiring an interactive exchange of messages in distributed wireless networks. Therefore, the proposed protocol has a great applicability in vehicular environments where network nodes move at high speeds in different directions, with a short time to communicate with other nodes.

It has also provided new solutions to manage revoked users in vehicular ad-hoc networks, through authenticated data structures and hash trees. It has proposed a management revoked certificates scheme using k-ary trees, with fully configurable parameters to adapt the structure to different vehicle environments, from urban scenarios to rural areas. In addition, an alternative based on Huffman codes has been proposed. This new management structure has as main goal to model the tree according to revoked vehicles types. After analyzing the vehicles behavior, it has been concluded that there are certain vehicles that spend more time on the road, so they are more likely to communicate with other vehicles. In this way, the tree has been designed in order to that the revocation proof is optimized, in size and processing time, for most queried vehicles. Thanks to the design of a tree whose leaf nodes are at different depths, the proposal has been developed according to transit vehicles over roads, such as public transport or delivery vehicles. These new proposals improve the performance of traditional solutions. Therefore, the revocation proof size, structure generation time and revocation process are all optimized.

Finally, it has presented several mobile applications based on researches produced during the Thesis. It describes a new algorithm that calculates the reliability between two specific users. Through the theory of six degrees of separation and the relationships in social networks, it has inicluded a scheme that can be used by any social application to increase its security. Several applications have been developed based on this algorithm to demonstrate its performance. A carpooling application has been implemented. This Thesis has detailed the implementation development from zero to an application

with really good metrics. Also it has explained other applications to prove that the theory researches can be applied in real environments. A mobile application to detect and report violations traffic light anonymously has been implemented. In this way, a user can notify about the hazard to other nearby users and generate some metrics to analyze the violations in different areas.

Regarding future works, the research has opened new challenges. Thus, not only new proposals to strengthen security in critical vehicular environments have been achieved, but they have also helped to detect and discover new future work to be carried out in these scenarios.

With respect to authentication issues, the research has generated several open problems. On the one hand, it is necessary to design some system data compression that would improve the size required to store the graphs used in the scheme. This will allow the system to be more efficient for networks over 100 nodes, avoiding the use of clusters algorithms. Other future research is related to reduce the decryption process of the messages. The generation packages process is really optimal, but its decryption becomes expensive for networks of more than 100 nodes. It would be advisable to improve these functions using parallel processing and optimization techniques. Lastly, it could be interesting to design new complex mathematical problems to provide the system with several types of challenges so that the structure of the messages are more unpredictable for attackers.

On the side of the certificate revocation schemes, it is necessary that they can be applied to vehicle systems, and not only to mobile devices. The tests assume that on board units and road side units are replaced by smartphones. It would be interesting to implement them in conventional VANETs, by using them on large projects for the deployment of vehicular ad-hoc networks. Other open work is the need to design a new, safer and efficient method to verify that a particular vehicle has not been revoked. The proposed schemes maximize and ensure the revocation of a particular vehicle, but the system does not have an optimized method to verify that a vehicle has not been revoked. Huffman trees allow segmenting revoked vehicles by their road time, providing a great improvement over traditional solutions. It would be desirable to continue to evolve this idea to segment vehicles by other behaviors, such as the number of connections performed with other vehicles, or the time drive in urban centers.

To conclude, the proposed reliability algorithm opens the possibility to be implemented in third party solutions. It would be good that a popular application with a large number of users could adopt the proposal and make A/B test to check if users feel safer using the algorithm designed. In addition, the algorithm must adopt new social networks such as Snapchat or Twitch. The algorithm has been implemented with Facebook, and it is ready to get data from Twitter and Google Plus. Another possible improvement would be

the addition of new parameters to feed the data system. Now comments and Like interactions are monitored. It might be interesting that the algorithm gets photos and their hashtag and mentions to detect moods that can add more quantitative data about the relationship between users. Regarding the scheme for detecting and reporting violations traffic light anonymously, it would be advisable to predict violations of traffic lights to provide advance notifications to nearby users. Machine learning by deep learning techniques and big data could be used to estimate future violations, based on past behavior and current speeds and distances. In this way, nearby users could take precautions in advance, and minimize the risk of accidents.

Bibliografía

El tiempo es el mejor autor; siempre encuentra un final perfecto.

Charles Chaplin

[1] M. Al-kahtani. Survey on security attacks in vehicular ad hoc networks (vanets). In *6th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–9, 2012.

[2] K. Al-Khateeb, J. Johari, and W. Al-Khateeb. Dynamic traffic light sequence algorithm using rfid. In *Journal of Computer Science*, volume 4, pages 517–524, 2008.

[3] N. Álvarez-Díaz. Trabajo fin de grado en ingeniería en informática: Localización interior y asignación de tareas con dispositivos bluetooth low energy. In *Dirigido por: Caballero-Gil, P. y Martín-Fernández F. Universidad de La Laguna. ETSI de Informática. 19 junio de 2015. Sobresaliente (10) (por unanimidad)*, 2015.

[4] N. Álvarez-Díaz, P. Caballero-Gil, and F. Martín-Fernández. Task assignment through indoor location with bluetooth low energy devices. In *4th International Conference on Theory and Practice in Modern Computing*, 2015.

[5] N. Álvarez-Díaz, P. Caballero-Gil, H. Reboso-Morales, and F. Martín-Fernández. Optimizing resource allocation and indoor location using bluetooth low energy. In *International Journal of Aerospace and Mechanical Engineering. 18th International Conference on Air Transport Management*, volume 3(2), page 440, 2016.

[6] E. Andreeva, B. Mennink, and B. Preneel. Security reductions of the second round sha-3 candidates. In *ISC, Lecture Notes in Computer Science*, volume 6531, pages 39–53, 2010.

[7] E. Andreeva, B. Mennink, B. Preneel, and M. Skrobot. Security analysis and comparison of the sha-3 finalists blake, grostl, jh, keccak, and skein. In *AFRICACRYPT*, pages 287–305, 2012.

[8] K. Aoki, G. Matusiewicz, K. an Roland, Y. Sasaki, and M. Schlaffer. Byte slicing grostl: Improved intel aes-ni and vector-permute implementations of the sha-3 finalist grostl. In *International Conference on E-Business and Telecommunications*, pages 281–295, 2012.

[9] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. In *Computer Networks*, volume 54(15), pages 2787–2805, 2010.

[10] L. Babai, P. Erdos, and S. S.M. Random graph isomorphism. In *SIAM Journal on Computing*, volume 9(3), pages 628–635, 1980.

[11] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. In *Cornell University Library*, 2011.

[12] D. Balfanz, D. Smetters, P. Stewart, and H. Chi-Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed System Security Symposium*, 2002.

[13] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Computer security. In *NIST Special Publication 800-57, Recommendation for Key Management. Part 1: General*, 2012.

[14] S. Berkovits, S. Chokhani, J. Furlong, J. Geiter, and J. Guild. Public key infrastructure study: final report. In *Technical Report, MITRE Corporation for NIST*, 1995.

[15] M. Bernstein, E. Bahsky, M. Burke, and B. Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30, 2013.

[16] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. On the indifferentiability of the sponge construction. In *EUROCRYPT Lecture Notes in Computer Science*, volume 4965, pages 181–197, 2008.

[17] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Keccak sponge function family main document version 2.1. In *Updated submission to NIST*, volume 2, 2010.

[18] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In *Selected Areas in Cryptography*, pages 320–337, 2011.

[19] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. The keccak sha-3 submission. `keccak.noekeon.org/Keccak-submission-3.pdf`, 2011. [Online; accedido 05-Mayo-2016].

[20] E. Blablacar. Blablacar. `https://www.blablacar.es`, 2014. [Online; accedido 05-Mayo-2016].

[21] S. Blake-Wilson. Information security, mathematics, and public-key cryptography. In *Designs, Codes and Cryptography*, volume 19(2-3), pages 77–99, 2000.

[22] J. Blum and A. Eskandarian. The threat of intelligent collisions. In *IT Prof*, volume 6(1), pages 24–29, 2004.

[23] M. Blum, Feldman, and S. P., Micali. Non-interactive zero-knowledge and its applications. In *ACM Symposium on Theory of Computing*, pages 103–112, 1988.

[24] B. Bochner and T. Walden. Effectiveness of red light cameras. In *Transportation Engineers Journal*, volume 80(5), page 18, 2010.

[25] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Crypto. LNCS 2139*, pages 213–229, 2001.

[26] C. Bonhomme, G. Arnould, and D. Khadraoui. Dynamic carpooling mobility services based on secure multi-agent platform. In *IEEE Global Information Infrastructure and Networking Symposium*, pages 1–6, 2012.

[27] D. Bornstein. Dalvik vm internals. `http://sites.google.com/site/io/dalvik-vm-internals`, 2008. [Online; accedido 05-Mayo-2016].

[28] A. Brusilovsky, I. Faynberg, Z. Zeltsan, and S. Patel. Password-authenticated key (pak) diffie-hellman exchange. In *RFC 5683*, 2010.

[29] A. Burg. Ad hoc network specific attacks, in: Seminar ad hoc networking: Concepts, applications, and security. In *Technische Universitat Munchen*, 2003.

[30] M. Butler. Android: Changing the mobile landscape. In *Pervasive Computing*, volume 10(1), pages 4–7, 2011.

[31] L. Buttyan and J.-P. Hubaux. Security and cooperation in wireless networks: Thwarting malicious and selfish behavior in the age of ubiquitous computing. In *Cambridge University Press*, 2008.

[32] C2C-CC. Car2car project. `https://www.car-2-car.org/index.php?id=5`, 2016. [Online; accedido 05-Mayo-2016].

[33] C. Caballero-Gil, J. Molina-Gil, P. Caballero-Gil, F. Martín-Fernández, and D. Yanes-García. Introducing secure and self-organized vehicular ad-hoc networks. In *Proceedings of the 12th International Conference on Computer Systems and Technologies*, pages 454–459, 2011.

[34] C. Caballero-Gil, J. Molina-Gil, J. Hernandez-Serrano, O. Leon, and M. Soriano. On the revocation of malicious users in anonymous and non-traceable vanets. In *XIII Reunion Española sobre Criptología y Seguridad de la Información*, pages 87–91, 2014.

[35] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Knowledge management using clusters in vanets-description, simulation and analysis. In *International Conference on Knowledge Management and Information Sharing*, pages 170–175, 2010.

[36] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Design and implementation of an application for deploying vehicular networks with smartphones. *International Journal of Distributed Sensor Networks*, pages 1–10, 2013.

[37] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, D. Yanes-García, and F. Martín-Fernández. Vaipho. un herramienta para asistencia en la conducción. In *TRANSNOVA: VIII Foro de Innovaciones tecnológicas para el transporte*, 2011.

[38] P. Caballero-Gil and C. Hernández-Goya. Efficient public key certificate management for mobile ad hoc networks. In *EURASIP Journal on Wireless Communications and Networking*, volume 18, 2011.

[39] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Tree-based management of revoked certificates in vehicular ad-hoc networks. In *Proceedings of the World Congress on Engineering 2013*, volume 2, pages 1425–1431, 2013.

[40] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Non-interactive authentication scheme for light environments. In *International Conference on Information Technology and Applications*, 2014.

[41] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Revocation for certificateless authentication in vanets. In *International Journal of Intelligent Computing Research*, volume 5, 2014.

[42] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Tree-based revocation for certificateless authentication in vehicular ad-hoc networks. In *Journal of Computer and Communications*, volume 2, 2014.

[43] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Tree-based revocation for certificateless authentication in vehicular ad-hoc networks. In *Proceedings of International Conference on Computational Intelligence and Software Engineering*, pages 14–21, 2014.

[44] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Using query frequencies in tree-based revocation for certificateless authentication in vanets. In *Proceedings of 9th International Conference for Internet Technology and Secured Transactions*, pages 268–273, 2014.

[45] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Cooperative social system based on trust for carpooling. In *IEICE Information and Communication Technology Forum*, 2015.

[46] P. Caballero-Gil, F. Martín-Fernández, and C. Caballero-Gil. Ubiquitous computing technologies to manage a transport monitoring system. In *IEICE Information and Communication Technology Forum*, 2015.

[47] R. Calvo, F. De Luigi, P. Haastrup, and V. Maniezzoi. A distributed geographic system for the daily car pooling problem. In *Computers and Operations Research 31*, volume 13, 2004.

[48] W. Castro, M. Schilgen, S. Meyer, M. Weber, C. Peuker, and K. Wortler. Do whiplash injuries occur in low-speed rear impacts? In *European Spine Journal*, volume 6(6), pages 366–375, 1997.

[49] C. Chin-Ling and L. Chen-Ta. Dynamic session-key generation for wireless sensor networks. In *EURASIP Journal on Wireless Communications and Networking*, pages 91–101, 2008.

[50] S. Cho, A. Yasar, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. A conceptual design of an agent-based interaction model for the carpooling application. In *1st International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications*, pages 801–807, 2011.

[51] C. Choi and Y. Park. Enhanced traffic light detection method using geometry information. In *International Journal of Computer, Control, Quantum and Information Engineering*, volume 8(8), pages 1264–1268, 2014.

[52] M. Collotta, G. Pau, V. Salerno, and G. Scata. A novel trust based algorithm for carpooling transportation systems. In *IEEE International Conference on Energy Conference and Exhibition*, pages 1077–1082, 2012.

[53] O. Community. Owasp internet of things project. `https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project`, 2015. [Online; accedido 05-Mayo-2016].

[54] Compartir.org. Compartir.org. `http://compartir.org/`, 2015. [Online; accedido 05-Mayo-2016].

[55] B. Consortium. Butler project. http://www.iot-butler.eu/, 2015. [Online; accedido 05-Mayo-2016].

[56] F. Consortium. Eu-china fire project. http://www.euchina-fire.eu/, 2015. [Online; accedido 05-Mayo-2016].

[57] F. Consortium. Smartfire project. http://eukorea-fire.eu/, 2015. [Online; accedido 05-Mayo-2016].

[58] S. S. Consortium. Smart santander project. http://www.smartsantander.eu/, 2015. [Online; accedido 05-Mayo-2016].

[59] T. Cormen, C. Leiserson, and R. Rivest. Introduction to algorithms. In *MIT Press*, 1990.

[60] D. Corneil and C. Gotlieb. An efficient algorithm for graph isomorphism. In *Journal of the ACM*, pages 51–64, 1970.

[61] D. Cuff, M. Hanse, and J. Kang. Urban sensing: Out of the woods. In *Communications of the ACM*, volume 51(3), pages 24–33, 2008.

[62] F. D. da Cunha, A. Boukerche, L. Villas, A. Carneiro Viana, and A. A. F. Loureiro. Data communication in vanets: A survey, challenges and applications. *Research Report: RR-8498, INRIA Saclay*, (1):30, 2015.

[63] E. Daraghmi and S. Yuan. We are so close, less than 4 degrees separating you and me! In *Computers in Human Behavior*, pages 273–285, 2014.

[64] R. De Charette and F. Nashashibi. Traffic light recognition using image processing compared to learning processes. In *ICEE/RSI International Conference on Intelligent Robots and Systems*, pages 333–338, 2009.

[65] D. de Tráfico. Portal estadistico: parque de vehiculos. http://www.dgt.es/, 2015. [Online; accedido 05-Mayo-2016].

[66] A. Dhamgaye and N. Chavhan. Survey on security challenges in vanet. *International Journal of Computer Sciences*, 2:88–96, 2013.

[67] W. Diffie and M. Hellman. New directions in cryptography. In *IEEE Transactions on Information Theory 22*, pages 644–654, 1976.

[68] W. Diffie, V. Oorschot, P.C., and M. Wiener. Authentication and authenticated key exchanges. In *Designs, Codes and cryptography*, volume 2(2), pages 107–125, 1992.

[69] D. Djenouri, L. Khelladi, and N. Badache. A survey of security issues in mobile ad hoc networks. In *IEEE communications surveys*, volume 7(4), pages 2–28, 2007.

[70] C. Doctorow. *All Complex Ecosystems Have Parasites.* 2005.

[71] J. Douceur. The sybil attack. In *Peer-to-Peer Systems, Springer*, pages 251–260, 2002.

[72] B. Ducourthial and F. El Ali. Architecture pour communication vehicules infrastructure. In *CFIP 2009*, 2009.

[73] P. Ekdahl and T. Johansson. Snow - a new stream cipher. In *NESSIE Workshop*, pages 167–168, 2000.

[74] P. Ekdahl and T. Johansson. A new version of the stream cipher snow. In *Selected Areas in Cryptography, Lecture Notes in Computer Science*, volume 2595, pages 37–46, 2003.

[75] ETSI. Intelligent transport systems (its), security, threat, vulnerability and risk analysis (tvra). In *Technical Report ETSI TR 102 893*, volume 1.1, 2010.

[76] ETSI/SAGE. Specification of the 3gpp confidentiality and integrity algorithms uea2 and uia2. In *SNOW 3G Specification*, volume 2, 2005.

[77] N. Fairfield and C. Urmson. Traffic light mapping and detection. In *IEEC International Conference on Robotics and Automation (ICRA)*, pages 5421–5426, 2011.

[78] U. Federal Highway Administration. U.s. department of transportation, federal highway administration. In *National Household Travel Survey*, 2009.

[79] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. In *Journal of Cryptology*, volume 1(2), pages 77–94, 1988.

[80] U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero knowledge proofs randomstring. In *Symposium on Foundations of Computer Science*, pages 308–317, 1990.

[81] J. Ferreira, P. Trigo, and P. Filipe. Collaborative car pooling system. In *International Conference on Sustainable Urban Transport and Environment*, 2009.

[82] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology-CRYPTO-86 Lecture Notes in Computer Science*, volume 263, pages 186–194, 1987.

[83] I. I. for Highway Safety. Iihs. `http://www.iihs.org`, 2016. [Online; accedido 05-Mayo-2016].

[84] I. C. for New Media. World summit awards. `http://www.wsis-award.org/`, 2015. [Online; accedido 05-Mayo-2016].

[85] N. Forest. Senticnel project. `http://ntforest.jimdo.com/productos-servicios/`, 2015. [Online; accedido 05-Mayo-2016].

[86] A. Fougeres, P. Canalda, T. Ecarot, A. Samaali, and L. Guglielmetti. A push service for carpooling. In *IEEE International Conference on Green Computing and Communications*, pages 685–691, 2012.

[87] J. d. Fuentes, A. González-Tablas, and R. A. Overview of security issues in vehicular ad-hoc networks. In *Handbook of Research on Mobility and Computing*, volume IGI Global, 2010.

[88] C. Gañan, J. Muñoz, O. Esparza, J. Mata-Díaz, and J. Alins. Coach: Collaborative certificate status checking mechanism for vanets. In *Journal Network Computer Applications*, volume 36(5), pages 1337–1351, 2013.

[89] M. Garey and D. Johnson. Computers and intractability: A guide the theory of np-completeness. In *Freeman and Co.*, 1979.

[90] Gartner. *Forecast: The Internet of Things.* 2013.

[91] C. GmbH. Carpooling. `http://www.carpooling.es`, 2014. [Online; accedido 05-Mayo-2016].

[92] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. In *Journal of the ACM*, volume 38(3), pages 690–728, 1991.

[93] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[94] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *SIAM Journal on Computing*, volume 18(1), pages 186–208, 1989.

[95] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. In *SIAM Journal on Computing*, volume 17(2), pages 281–308, 1988.

[96] M. Goodrich, M. Shin, R. Tamassia, and W. Winsborough. Authenticated dictionaries for fresh attribute credentials, trust management. In *Lecture Notes in Computer Science*, volume 2692, pages 332–347, 2003.

[97] M. Goodrich, R. Tamassia, N. Triandopoulos, and R. Cohen. Authenticated data structures for graph and geometric searching. In *Lecture Notes in Computer Science*, volume 2612, pages 295–313, 2003.

[98] Google. Brillo: Iot operating system. `https://developers.google.com/brillo/`, 2016. [Online; accedido 05-Mayo-2016].

[99] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode. Adaptive traffic lights using car-to-car communication. In *IEEE Vehicular Technology Conference, VTC2007-Spring*, pages 21–25, 2007.

[100] J. Granjal, E. Monteiro, and J. Sa Silva. Security for the internet of things: A survey of existing protocols and open research issues. In *IEEE Communication Surveys and Tutorials*, volume 17(1), pages 1294–1312, 2015.

[101] S. Grzonkowski, W. Zaremba, M. Zaremba, and B. McDaniel. Extending web applications with a lightweight zero knowledge proof authentication. In *Conference on Soft Computing as Transdisciplinary Science and Technology*, pages 65–70, 2008.

[102] X. Guo, M. Srivastav, S. Huang, D. Ganta, M. Henry, L. Nazhandali, and P. Schaumont. Asic implementations of five sha-3 finalists. In *IEEE Design, Automation and Test in Europe Conference and Exhibition*, pages 1006–1011, 2012.

[103] H. C. Hall. Helb project. `https://customers.microsoft.com/Pages/CustomerStory.aspx?recid=571`, 2015. [Online; accedido 05-Mayo-2016].

[104] A. Hamieh, J. Ben-Othman, A. Gueroui, and F. Nait-Abdesselam. Detecting greedy behaviors by linear regression in wireless ad hoc networks. In *IEEE International Conference on Communications*, pages 1–6, 2009.

[105] A. Hamieh, J. Ben-Othman, and M. L. Detection of radio interference attacks in vanet. In *Global Telecommunications Conference*, pages 1–5, 2009.

[106] K. Hartke. Practical issues with datagram transport layer security in constrained environments. In *draft-hartke-dice-practical-issues-01*, 2014.

[107] J.-H. Hoepman. *Constructing Ambient Intelligence: AmI 2011 Workshops, Amsterdam, The Netherlands, November 16-18, 2011. Revised Selected Papers*, chapter In Things We Trust? Towards Trustability in the Internet of Things, pages 287–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[108] D. Howard and K. Prince. *Security 2020: Reduce Security Risks This Decade*. 2011.

[109] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. In *IEEE Security and Privacy*, volume 2(3), pages 49–55, 2004.

[110] D. Huffman. A method for the construction of minimum-redundancy codes. In *Proceedings of IRE*, volume 40(9), pages 1098–1101, 1952.

[111] R. Hummen. 6lowpan fragmentation attacks and mitigation mechanisms. In *6th ACM Conf. WiSec*, pages 55–66, 2013.

[112] IEEE. Ieee standard for local and metropolitan area networks, part 15.4, low-rate wireless personal area networks. In *Revision of IEEE Std. 802.15.4-2006*, pages 1–314, 2011.

[113] IEEE-1609. Family of standards for wireless access in vehicular environments (wave). In *US Department of Transportation*, 2006.

[114] IPT-2012-0585-370000. Dephisit: Desarrollo experimental de una plataforma híbrida inalámbrica para sistemas inteligentes de transporte. `http://cryptull.webs.ull.es/DEPHISIT/`, 2012-2016. [Online; accedido 05-Mayo-2016].

[115] iSoftStone. Smart agriculture cloud platform. `http://www.isoftstone.com.cn/en/conleft/ind.aspx?nodeid=655`, 2015. [Online; accedido 05-Mayo-2016].

[116] ISO/IEC-9796-2:2010. Information technology, security techniques, digital signature schemes giving message recovery. In *Part 2: Integer factorization based mechanisms*, 2010.

[117] M. Jakobsson, T. Leighton, S. Micali, and M. Szydlo. Fractal merkle tree representation and traversal. In *Lecture Notes in Computer Science*, volume 2612, pages 314–326, 2003.

[118] D. Jiang and L. Delgrossi. Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In *IEEE Vehicular Technology Conference VTC Spring*, pages 2036–2040, 2008.

[119] D. Johnson, A. Menezes, and A. Vanstone. The elliptic curve digital signature algorithm (ecdsa). In *International Journal of Information Security*, volume 1(1), pages 36–61, 2001.

[120] H. Juma, I. Kamel, and L. Kaya. Protecting the integrity of sensor data. In *Proceedings of 15th IEEE International Conference on Electronics, Circuits and Systems*, pages 902–905, 2008.

[121] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. Spirito. Demo: An ids framework for internet of things empowered by 6lowpan. pages 1337–1339, 2013.

[122] S. Kaushik. Review of different approaches for privacy scheme in vanets. *International Journal*, 5(2), 2012.

[123] S. Keoh, S. Kumar, O. Garcia-Morchon, and E. Dijk. Dtls-based multicast security for low-power and lossy networks (llns). In *draft-keohdice-multicast-security-08*, 2014.

[124] N. Khan and C. Nwe. Implementation of modern traffic light control system. In *International Journal of Scientific and Research Publications*, volume 4(6), 2014.

[125] H. Kim. Protection against packet fragmentation attacks at 6lowpan adaptation layers. In *ICHIT*, volume 17(1), pages 796–801.

[126] A. Kircanski. Cryptanalysis of symmetric cryptographic primitives. In *A Thesis in The Department of Computer Science and Software Engineering at Concordia University*, 2013.

[127] P. Kocher. On certificate revocation and validation. In *Financial Cryptography, Lecture Notes in Computer Science*, volume 1465, pages 172–177, 1998.

[128] K. Laberteaux, J. Haas, and Y. Hu. Security certificate revocation list distribution for vanet. In *Fifth ACM international workshop on VehiculAr InterNETworking*, 2008.

[129] R. Lakhani-Lakhani. Trabajo fin de grado en ingeniería en informática: Localización de personas en entornos rurales mediante la combinación de sensores y teléfonos móviles usando tecnología ble. In *Dirigido por: Caballero-Gil, P. y Martín-Fernández F. Universidad de La Laguna. ETSI de Informática. 23 julio de 2015. Sobresaliente (9)*, 2015.

[130] R. Lakhani-Lakhani. Trabajo fin de grado en ingeniería en informática: Sistema inteligente de detección y aviso de infracciones en semáforos mediante smartphones. In *Dirigido por: Hernández-Goya C. y Martín-Fernández F. Universidad de La Laguna. ETSI de Informática. 22 julio de 2015. Sobresaliente (10) (por unanimidad)*, 2015.

[131] D. Lapidot and A. Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In *Advances in Cryptology-CRYPT0-90, Lecture Notes in Computer Science*, volume 537, pages 353–365, 1990.

[132] U. Lee and M. Gerla. A survey of urban vehicular sensing platforms. In *Computer Networks*, volume 54(4), pages 527–544, 2010.

[133] M. Lehman. Programs, life cycles, and laws of software evolution. In *Proceedings of the IEEE*, volume 68(9), pages 1060–1076, 1908.

[134] Libelium, CARTIF, E. B. Network, S. C. Hall, and R. A. G. PG. Rescatame project. `http://www.libelium.com/smart_city_air_quality_urban_traffic_waspmote/`, 2011. [Online; accedido 05-Mayo-2016].

[135] S. I. Limited. Mumbai vts. `http://www.soprasteria.in/`, 2014. [Online; accedido 05-Mayo-2016].

[136] X. Lin. Security in vehicular ad hoc networks. In *IEEE Communications Magazine*, volume 46(4), pages 88–95, 2008.

[137] X. Lin, R. Lu, C. Zhang, H. Zhu, P. Ho, and X. Shen. Security in vehicular ad hoc networks. In *IEEE Communications Magazine*, pages 88–95, 2008.

[138] S.-C. Lo, J.-S. Gao, and C.-C. Tseng. A water-wave broadcast scheme for emergency messages in vanet. In *Wireless Personal Communications*, volume 71(1), pages 217–244, 2013.

[139] M. R. Lodeiro-Santiago. Trabajo fin de grado en ingeniería en informática: Sistema de decisión inteligente para la toma de pulsaciones cardíacas usando android wear. In *Dirigido por: Roda-García, José Luis y Martín-Fernández F. Universidad de La Laguna. ETSI de Informática. 22 junio de 2015. Sobresaliente (10) (por unanimidad)*, 2015.

[140] B. M. and M. Yung. Certifying cryptographic tools: the case of trapdoor permutations. In *Advances in Cryptology-CRYPTO-92, Lecture Notes in Computer Science*, pages 442–460, 1992.

[141] J. P. Macker and M. S. Corson. Mobile ad hoc networking and the ietf. In *ACM SIGMOBILE Mobile Computing and Communications Review*, volume 2(1), pages 9–14, 1998.

[142] E. Magistretti, U. Lee, M. Gerla, P. Bellavista, and A. Corradi. Dissemination and harvesting of urban data using vehicular sensing platforms. In *IEEE Transactions on Vehicular Technology*, volume 58(2), pages 882–901, 2009.

[143] A. Martin. On some symmetric lightweight cryptographic designs. In *Doctoral Dissertion, PhD*, 2012.

[144] F. Martín-Fernández. Proyecto fin de carrera en ingeniería en informática: Implementación de comunicaciones seguras en la plataforma symbian para asistencia a la conducción. In *Dirigido por: Caballero-Gil, P. y Caballero-Gil, C. Universidad de La Laguna. ETSI de Informática. 1 junio de 2011. Sobresaliente (10) (por unanimidad)*, 2011.

[145] F. Martín-Fernández. Riesgos en smartphones e internet de las cosas. In *Primer Workshop en Seguridad en Internet de las Cosas*, 2012.

[146] F. Martín-Fernández. Trabajo fin de máster en ingeniería en informática: Introducción a la seguridad inalámbrica en internet de las cosas. In *Dirigido por: Caballero-Gil, P. y Caballero-Gil, C. Universidad de La Laguna. ETSI de Informática. 11 junio de 2012. Sobresaliente (10) (por unanimidad)*, 2012.

[147] F. Martín-Fernández. Chascar en tenerife. `https://chascarentenerife.wordpress.com/`, 2013. [Online; accedido 05-Mayo-2016].

[148] F. Martín-Fernández. Some uses of the new standard hash function. In *II Workshop on Security in Internet of Things*, 2013.

[149] F. Martín-Fernández. Qdemos. `https://qdemos.wordpress.com/`, 2014. [Online; accedido 05-Mayo-2016].

[150] F. Martin-Fernandez. Source code of the authenticated scheme proposed. `https://github.com/pacomf/ASD`, 2014. [Online; accedido 05-Mayo-2016].

[151] F. Martín-Fernández. Ad-hoc networks for disasters. In *First Workshop on Mobile Tools for Emergencies and Critical Infrastructures*, 2015.

[152] F. Martín-Fernández, C. Caballero-Gil, J. Molina-Gil, and P. Caballero-Gil. Plataforma móvil segura y confiable para carpooling. In *TRANSNOVA: X Foro de Innovaciones tecnológicas para el transporte*, 2013.

[153] F. Martín-Fernández and P. Caballero-Gil. Use of a duplex construction of sha-3 for certificate revocation in vanets. In *Proceedings of International Workshop on Security In Information Systems*, volume 1, pages 3–11, 2013.

[154] F. Martín-Fernández and P. Caballero-Gil. Version of the new sha standard applied to manage certificate revocation in vanets. In *IWANN: International Work Conference on Artificial Neuronal Networks*, volume LNCS 7902, pages 161–168, 2013.

[155] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Analysis of the new standard hash function. In *EUROCAST: XIV International Conference On Computer Aided Systems Theory*, volume LNCS 8111, pages 142–149, 2013.

[156] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Algoritmos criptográficos y aplicaciones seguras para escenarios de transporte. In *CyberCamp: I PhD WorkShop en Seguridad*, 2014.

[157] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Autenticación no interactiva para internet de las cosas. In *XIII Reunión Española sobre Criptología y Seguridad Informática*, 2014.

[158] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Autenticación no interactiva para internet de las cosas. In *I Conferencia de Jóvenes Investigadores en las Islas Canarias*, 2015.

[159] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Detection and report of traffic lights violation using sensors and smartphones. In *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information*, volume 9454, pages 237–248, 2015.

[160] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Efficient management of revoked pseudonyms in vanets using id-based cryptography. In *Proceedings of the 17th International Conference on Enterprise Information Systems*, volume 2, pages 701–708, 2015.

[161] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Esquemas criptográficos en redes mòviles autogestionadas. In *Congreso de Jóvenes Investigadores de la Real Sociedad Matemática Española*, 2015.

[162] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Â¿estamos preparados para la internet de las cosas? In *Reunión Jóvenes Investigadores del Instituto de Desarrollo Regional de la Universidad de La Laguna*, 2015.

[163] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Managing certificate revocation in vanets using hash trees and query frequencies. In *EUROCAST: XV International Conference On Computer Aided Systems Theory*, volume LNCS 9520, pages 57–63, 2015.

[164] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Revocation in vanets based on k-ary huffman trees. In *1st International Workshop on Experiences with the Design and Implementation of Smart Objects*, pages 25–26, 2015.

[165] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Revocation management in vehicular ad-hoc networks. In *IEEE Trustcom/-BigDataSE/ISPA*, volume 1, pages 1210–1217, 2015.

[166] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. A trustworthy distributed social carpool method. In *Euro-Par 2015: Parallel Processing Workshops*, volume 9523, pages 324–335, 2015.

[167] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. Authentication based on non-interactive zero-knowledge proofs for the internet of things. *Sensors*, 16(1):75, 2016.

[168] F. Martín-Fernández, P. Caballero-Gil, and C. Caballero-Gil. An experimental hybrid wireless platform for vehicular networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2016.

[169] F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Conexión segura entre dispositivos móviles para la asistencia a la conducción. In *RECSI: XII Reunión en Criptografía y Seguridad de la Información*, 2012.

[170] F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Implementación de comunicaciones seguras en plataformas móviles para la asistencia a la conducción. In *CIT: X Congreso de Ingeniería del Transporte*, 2012.

[171] F. Martín-Fernández, P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Increasing privacy and trust in cooperative social platforms for vehicular applications. In *Proceedings of International Workshop on Security In Information Systems*, volume 1, pages 3–13, 2014.

[172] F. Martín-Fernández and J. Carballo-Franquis. App patea la palma. `https://play.google.com/store/apps/details?id=com.jelcaf.pacomf.patealapalma&hl=es`, 2015. [Online; accedido 05-Mayo-2016].

[173] F. Martín-Fernández and J. Carballo-Franquis. Web patea la palma. `https://dl.dropboxusercontent.com/u/23163327/index.html`, 2015. [Online; accedido 05-Mayo-2016].

[174] F. Martín-Fernández, A. Rivero-García, and I. Santos-Gonzalez. Shorcial. `https://shorcial.wordpress.com/`, 2014. [Online; accedido 05-Mayo-2016].

[175] F. Martín-Fernández, D. Yanes-García, P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. Detecta atascos de tráfico y aparcamientos libres en tu smartphone. In *SALT: Salón atlántico de logística y transporte*, 2011.

[176] M. Mathew and A. Raj. Threat analysis and defence mechanisms in vanet. *Journal Advances Research in Computer Science Software Engineering*, pages 47–53, 2013.

[177] U. Maurer. Modelling a public-key infrastructure. In *Computer Security-ESORICS 96*, pages 325–350, 1996.

[178] C. Mayer. Security and privacy challenges inthe internet of things. In *Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2009 (WowKiVS 2009)*, volume 17, pages 1–12, 2009.

[179] K. McCurley. A key distribution system equivalent to factoring. In *Journal of Cryptology*, volume 1(2), pages 95–105, 1988.

[180] A. McMichael. The urban environment and health in a world of increasing globalization: issues for developing countries. In *Bulletin of the World Health Organization*, volume 78(9), pages 1117–1126, 2000.

[181] M. N. Mejria, J. Ben-Othmana, and M. Hamdi. Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, 2014.

[182] R. Merkle. Protocols for public key cryptosystems. In *IEEE Symposium on Security and privacy*, volume 1109, pages 122–134, 1980.

[183] R. Merkle. Method of providing digital signatures. In *U.S. Patent No. 4309569*, 1982.

[184] V. Miller. Short programs for functions on curves. In *Unpublished manuscript*, volume 97, pages 101–102, 1986.

[185] R. Minelli and M. Lanza. Software analytics for mobile applications insights and lessons learned. In *15th European Conference on Software Maintenance and Reengineering*, volume 0, pages 144–153, 2011.

[186] R. Minhas and M. Tilal. Effects of jamming on ieee 802.11 p systems. In *Chalmers University of Technology*, 2010.

[187] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. In *Ad Hoc Networks*, volume 10(7), pages 1497–1516, 2012.

[188] J. Molina-Gil, P. Caballero-Gil, and C. Caballero-Gil. Enhancing cooperation in wireless vehicular networks. In *International Workshop on Security in Information Systems*, pages 91–102, 2011.

[189] J. Molina-Gil, P. Caballero-Gil, and C. Caballero-Gil. Self-organized clustering architecture for vehicular ad hoc networks. In *International Journal of Distributed Sensor Networks*, 2015.

[190] MongoDB. Mongodb. `http://www.mongodb.org`, 2011. [Online; accedido 05-Mayo-2016].

[191] G. Motors. Onstar. `https://www.onstar.com/us/en/home.html`, 2016. [Online; accedido 05-Mayo-2016].

[192] J. Muñoz, O. Esparza, C. Gañán, J. Mata-Díaz, J. Alins, and I. Ganchev. Mht-based mechanism for certificate revocation in vanets. In *Wireless networking for moving objects: protocols, architectures, tools, services and applications*, pages 282–300, 2014.

[193] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X. 509 internet public key infrastructure online certificate status protocol-ocsp. In *RFC 2560*, 1999.

[194] M. Naor and K. Nissim. Certificate revocation and certificate update. In *IEEE Journal on Selected Areas in Communications*, volume 18(4), pages 561–570, 2000.

[195] T. T. Networks. Sonata project. `http://www.taptapnetworks.com/`, 2010. [Online; accedido 05-Mayo-2016].

[196] M. Nogueira, H. Silva, A. Santos, and P. Guy. A security management architecture for supporting routing services on vanets. In *IEEE Transmission Networks Services Management*, volume 9(2), pages 156–168, 2012.

[197] NS2. ns-2 simulator. `http://www.isi.edu/nsnam/ns/`, 2015. [Online; accedido 05-Mayo-2016].

[198] U. D. of Transportation. Traffic safety facts 2008 report. In *National Statistics*, 2008.

[199] OpenDataCanarias. Ii concurso open data canarias. `http://www.opendatacanarias.es/blog/ii-concurso-open-data-canarias-2014`, 2014. [Online; accedido 05-Mayo-2016].

[200] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure vehicular communications: Design and architecture. In *IEEE Communications Magazine*, volume 46(11), pages 2–8, 2008.

[201] B. Parno and A. Perrig. Challenges in securing vehicular networks. In *Workshop on Hot Topics in Networks*, pages 1–6, 2005.

[202] C. S. Patil, R. Karhe, and M. A. Aher. Development of mobile technology: A survey. In *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, volume 1(5), pages 374–379, 2012.

[203] Phillips and Ericsson. Zero site. `http://www.newscenter.philips.com/es_es/standard/news/press/2014/20140224-philips-ericsson.wpd#.VvZ6aOLJyM9`, 2014. [Online; accedido 05-Mayo-2016].

[204] W. J. Pires, T. d. Paula Figueiredo, H. Wong, and A. Loureiro. Malicious node detection in wireless sensor networks, in: 18th international parallel and distributed processing symposium. In *Proceedings IEEE*, page 24, 2004.

[205] N. Pukhovskiy and R. Lepshokov. Real-time carpooling system. In *International Conference on Collaboration Technologies and Systems*, pages 648–649, 2011.

[206] A. Rawat, S. Sharma, and R. Sushil. Vanet: security attacks and its possible solutions. In *Journal Information Operating Management*, volume 3(1), pages 301–304, 2012.

[207] M. Raya. Eviction of misbehaving and faulty nodes in vehicular networks. In *IEEE Journal on Selected Areas in Communications*, volume 25(8), pages 1557–1568, 2007.

[208] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Third ACM workshop on Security of ad hoc and sensor networks*, 2005.

[209] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. In *Journal Computer Security*, volume 15(1), pages 39–68, 2007.

[210] M. Raya, D. Jungels, and P. Papadimitratos. Certificate revocation in vehicular networks. In *Laboratory for computer Communications and Applications*, 2006.

[211] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing vehicular communications. In *IEEE Wireless Communications*, volume 13(5), pages 8–15, 2006.

[212] A. Research. *ABI Research Internet of Everything Research Service*. 2013.

[213] R. Retting, S. Ferguson, and C. Farmer. Reducing red light running through longer yellow signal timing and red light camera enforcement: results of a field investigation. In *Accident Analysis and Prevention*, volume 40(1), pages 327–333, 2008.

[214] R. Retting, A. Williams, D. Preusser, and H. Weinstein. Classifying urban crashes for countermeasure development. In *Accident Analysis and Prevention*, volume 27(3), pages 283–294, 1995.

[215] RFDuino. Rfduino project. `http://www.rfduino.com/`, 2016. [Online; accedido 05-Mayo-2016].

[216] A. Rivero-Garcia. Asistente de conduccion en intersecciones mediante dispositivos moviles, crossroad. In *Proyecto Fin de Master dirigido por Hernández-Goya, C. y Caballero-Gil, P.*, 2014.

[217] A. Roberts, H. Pimentel, and S. Karayevm. Cabfriendly: A cloud-based mobile web app. In *Amazon's EC2*, 2011.

[218] R. Rodrigo, C. Alcaraz, J. Lopez, and N. Sklavos. Key management systems for sensor networks in the context of the internet of things. In *Computer Electronical Engineering*, volume 37(2), pages 147–159, 2011.

[219] R. Roman, J. Zhou, and J. Lopez. On the features and challenges of security and privacy in distributed internet of things. In *Computer Networks*, pages 2266–2279, 2013.

[220] K. Rose, S. Eldridge, and L. Chapin. The internet of things: An overview. understanding the issues and challenges of a more connected world. In *Internet Society*, pages 1–50, 2015.

[221] S. RoselinMary, M. Maheshwari, and T. M. Early detection of dos attacks in vanet using attacked packet detection algorithm (apda). In *International Conference on Information Communication and Embedded Systems*, pages 237–240, 2013.

[222] RTC-2014-1648-8. Atlas: Aplicaciones de la tecnología lte para aumentar la seguridad. `http://cryptull.webs.ull.es/ATLAS/`, 2015-2018. [Online; accedido 05-Mayo-2016].

[223] S. Safi and M. M. Movaghar, A. A novel approach for avoiding wormhole attacks in vanet. In *First Asian Himalayas International Conference on Internet*, pages 1–6, 2009.

[224] G. Samara and W. Al-Salihy. A new security mechanism for vehicular communication networks. In *IEEE International Conference on Cyber Security*, volume 1, pages 18–22, 2012.

[225] G. Samara, S. Ramadas, and W. AlSalihy. Design of simple and efficient revocation list distribution in urban areas for vanets. In *International Journal of Computer Science*, volume 8, 2010.

[226] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki. Caravan project: providing location privacy for vanet. Technical report: DTIC Document, 2005.

[227] L. Sarakis, T. Orphanoudakis, H. C. Leligou, S. Voliotis, and A. Voulkidis. Providing entertainment applications in vanet environments. In *IEEE Wireless Communications*, volume 23(1), pages 30–37, 2016.

[228] B. Schneier. Applied cryptography. protocols, algorithms, and source code in c. In *John Wiley and Sons*, 1996.

[229] S. E. SEGITTUR. Apptourism awards. `http://www.segittur.es/es/proyectos/proyecto-detalle/CONCURSO-The-AppTourism-Awards-2015/#.VwjP-fnhCM8`, 2015. [Online; accedido 05-Mayo-2016].

[230] S. Sicari, A. Rizzardi, L. Frieco, and A. Coen-Porisini. Security, privacy and trust in internet of things: The road ahead. In *Computer Networks*, volume 76, pages 146–164, 2015.

[231] P. Simmonds. Security. *Computer World*, (1):45, 2007.

[232] D. Singelee and B. Preneel. Location verification using secure distance bounding protocols. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*, page 7, 2005.

[233] A. Startup. Amovens. `https://www.amovens.com`, 2015. [Online; accedido 05-Mayo-2016].

[234] A. Studer, E. Shi, F. Bai, and A. Perrig. Tacking together efficient authentication, revocation, and privacy in vanets. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 1–9, 2009.

[235] T. Sukuvaara, K. Maenpaa, R. Ylitalo, P. Nurmi, and E. Atlaskin. Interactive local road weather services through vanet-capable road weather station. In *Conference: World Congress on ITS*, volume 20, 2014.

[236] SUMO. Sumo simulator. `http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/`, 2015. [Online; accedido 05-Mayo-2016].

[237] I. Sumra, J.-L. Ab Manan, and H. Hasbullah. Timing attack in vehicular network. In *Proceedings of the 15th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society (WSEAS)*, pages 151–155, 2011.

[238] L. Sweeney. k-anonymity: a model for protecting privacy. In *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, volume 10(5), pages 557–570, 2002.

[239] A. Taivalsaari, T. Mikkonen, M. Anttonen, and A. Salminen. The death of binary software: End user software moves to the web. In *Ninth International Conference on Creating, Connecting and Collaborating through Computing*, pages 17–23, 2011.

[240] R. Teal. Carpooling: Who, how and why. transportation research. In *IEEE International Conference on Energy Conference and Exhibition*, volume 21, pages 203–214, 1987.

[241] E. Team. Estimote. `http://estimote.com/`, 2013. [Online; accedido 05-Mayo-2016].

[242] TEC2014-54110-R. Casus: Cooperación móvil segura aplicada a situaciones de emergencia e infraestructuras críticas de transporte. `http://cryptull.webs.ull.es/CASUS/`, 2015-2018. [Online; accedido 05-Mayo-2016].

[243] P. Thubert. Rpl: Ipv6 routing protocol for low-power and lossy networks. In *RFC 6550*, 2012.

[244] TIN2008-02236/TSI. Muove: Mejora de la seguridad vial mediante la planificacion, diseno e integracion de servicios criptograficos en vanets. `TIN2008-02236/TSI`, 2009-2011. [Online; accedido 05-Mayo-2016].

[245] TIN2011-25452. Tueri: Tecnologías seguras y eficientes para las redes inalámbricas en la internet de las cosas con aplicaciones en transporte y logística. `http://cryptull.webs.ull.es/TUERI/`, 2012-2014. [Online; accedido 05-Mayo-2016].

[246] M. Toorani and A. Beheshti. Lpki-a lightweight public key infrastructure for the mobile environments. In *IEEE Singapore International Conference on Communication Systems*, pages 162–165, 2008.

[247] J. Turian. Using alchemyapi for enterprise-grade text analysis. In *AlchemyAPI*, 2013.

[248] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. In *Cornell University Library*, 2011.

[249] A. Wasef and X. Shen. Maac: message authentication acceleration protocol for vehicular ad hoc networks. In *IEEE conference on Global telecommunications*, pages 4476–4481, 2009.

[250] D. Watts and S. Strogatz. Collective dynamics of small-world networks. In *Nature 393*, pages 440–442, 1998.

[251] R. H. Weber. Internet of things: New security and privacy challenges. In *Computer Law and Security Review*, volume 26, pages 23–39, 2010.

[252] L. with Sea Change Strategies and E. Research. The state of friendship in america 2013. In *A crisis of confidence*, 2013.

[253] B. Wu, J. Chen, J. Wu, and M. Cardei. A survey of attacks and countermeasures in mobile adhoc networks. In *Wireless network Security*, pages 103–135, 2007.

[254] G. Xie, J. Chen, and I. Neamtiu. Towards a better understanding of software evolution: An empirical study on open source software. In *IEEE International Conference on Software Maintenance*, pages 51–60, 2009.

[255] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and H. A. Vehicular ad hoc networks (vanets): status, results, and challenges. In *Telecommunications Systems*, volume 50(4), pages 217–241, 2012.

[256] J. Zhang, S. Sagar, and E. Shihab. The evolution of mobile apps: An exploratory study. In *Proceedings DeMobile '13*, pages 1–8, 2013.

[257] Zimride. Zimride platform. `http://www.zimride.com/`, 2015. [Online; accedido 05-Mayo-2016].

I'll see you in another life, brother.