# Design of Secure Mobile Payment Protocols for Restricted Connectivity Scenarios

## Tesis Doctoral

**Autor:** Jesús Téllez Isaac

**Director:** Jose María Sierra Cámara

**Tutor:** Miguel Soriano Ibañez

Mayo 2012

# Índice general

# Capítulo 1

# Introducción

## 1.1. Justificación de la Unidad Temática de la Tesis

Desde la aparición de la primera tarjeta de crédito en los años 50 (seguido por las tarjetas de débito), el llamado *dinero plástico* ha sido aceptado mundialmente como un método de pago en el cual el pagador debe presentar físicamente la tarjeta y firmar un recibo de pago como evidencia de pago. El nacimiento y la expansión de Internet ha ayudado al desarrollo de sistemas de comercio electrónico que permiten pago en-línea con tarjetas de crédito y/o débito. Como consecuencia, el pago electrónico se convirtió en un tema importante del comercio electrónico y su seguridad no sólo hará que las actividades de compra sean más convenientes y flexibles de ejecutar sino que habilitará nuevas oportunidades de negocio [8].

La aparición de las redes móviles e inalámbricas hizo posible la extensión del comercio electrónico a un nuevo uso y área de investigación: el comercio electrónico móvil (que incluye el pago móvil). Esta nueva área de aplicación trajo consigo la necesidad de realizar esfuerzos para asegurar el pago móvil.

La observación en la tendencia de crecimiento del comercio electrónico móvil y la aparición de situaciones en las cuales este tipo de comercio no es posible debido a restricciones de conectividad presente en alguna de las entidades que forman parte del sistema de pago electrónico móvil, representan el punto de partida de esta tesis doctoral. Así, este trabajo se enfoca en el reto que representan las restricciones de comunicación en el diseño de los futuros sistemas de pago electrónico móvil a fin de ampliar el espectro de posibilidades del comercio electrónico móvil en el mundo real.

Un análisis de la evolución del comercio electrónico móvil muestra los esfuerzos realizados en sus inicios para adaptar los sistemas de pago electrónico existentes para redes fijas a las especificaciones de las redes inalámbricas lo que permitió un impulso importante para el m-comercio y aprovechar los esfuerzos realizados para el e-comercio [20, 24, 23]. Sin embargo, dado que la tecnología inalámbrica permitió el surgimiento de nuevos modelos de comercio, se hizo necesario diseñar nuevos sistemas de pagó móvil capaces de explotar estos nuevos

modelos a fin de aumentar las posibilidades de pago a los usuarios desde sus dispositivos móviles.

La mayoría de los sistemas de pago móvil existentes en la literatura están basados en escenarios de donde las entidades se pueden comunicar entre sí de manera directa(llamado Modelo de Conectividad Completa, ver figura 1.1) y no admiten modelos de negocio donde las restricciones de comunicación directa entre entidades del sistema no es un impedimento para realizar las transacciones comerciales. Es por ello que los futuros sistemas de pago móvil deberían considerar aquellas situaciones en las cuales la comunicación directa entre dos entidades del sistema no es posible (de manera temporal o permanente) básicamente por la imposibilidad de una de las entidades de conectarse a Internet. Sin embargo, este tipo de restricción no debe impedir que la transacción comercial se lleve a cabo con niveles de seguridad similares a aquellas situaciones en las cuales todas las entidades del sistema de pueden comunicar entre sí de manera directa.

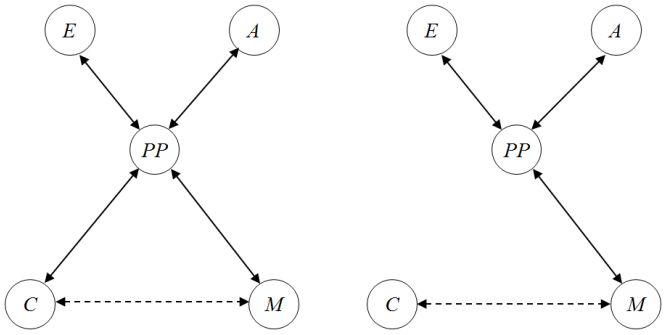


Figura 1.1: Modelo de Conectividad Completa (izquierda) y Modelo Centrado en el Comercio (derecha).

Diversos escenarios de conectividad restringida podrían ser definidos (basado en la interacción de las 3 entidades que conforman un sistema de pago: cliente, comercio y pasarela de pago) pero los de mayor interés, desde el punto de vista práctico, son los que se describen a continuación [4]:

1. *Modelo Centrado en el Comercio:* En este modelo (ver Figura 1.1), el *cliente* **C** sólo se puede conectar con el *comercio* **M** a través de un enlace de corto alcance o mediano alcance (como Infrarrojo, Wi-Fi, Bluetooth o NFC[1]). En otras palabras, el cliente no podrá conectarse de manera directa con su banco (llamado *Emisor* **E**) y tendrá que hacerlo a través del vendedor y la *pasarela de pago* **PP**.

   Este escenario permite desarrollar *sistemas de pago en línea* donde el usuario puede realizar compras seguras desde un dispositivo móvil sin necesidad de estar conectado a Internet, lo que se traduce en un ahorro para el usuario ya que no tiene que pagar ningún coste de conexión a Internet.

[1]Near Field Communication (NFC) es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia, que permite el intercambio de datos entre dispositivos a menos de 10cm

2. *Modelo Centrado en el Cliente:* A diferencia del modelo anterior, en este escenario (ver figura 1.2) el comercio **M** no puede conectarse con su banco (llamado *Adquiriente* **A**) mientras que el cliente **C**, utiliza su dispositivo móvil para comunicarse con su banco a través de Internet y un enlace de corto alcance para comunicarse con el comercio **M** (de manera similar que en el Modelo Centrado en el Comercio).

   Los sistemas de pago basados en este modelo permiten que el comercio pueda vender sus productos y/o servicios de manera segura aún cuando no cuente con la infraestructura necesaria para comunicarse de manera directa con su banco. La comunicación entre el comercio **M** y el adquiriente **A** se hará a través del cliente **C**, quien requerirá mayor poder computacional para ejecutar todas las operaciones necesarias para la autorización del pago.

3. *Modelo Centrado en la Pasarela de Pago:* En este modelo, mostrado en la Figura 1.2, no existe comunicación directa entre el cliente **C** y el comercio **M** por lo que se debe hacer a través de la pasarela de pago **PP** quien actúa como intermediario entre ambas entidades. Esta situación introduce algunas desventajas que se mencionan a continuación:

   - La restricción de comunicación que introduce este modelo, aumenta el número de mensajes que se intercambian entre **C** y **M** a fin de lograr la autorización del pago.
   - Tanto el cliente como el comercio deben tener la capacidad de conectarse a Internet para poder comunicarse con **PP**.
   - Pese a que **C** y **M** pueden estar físicamente cerca, no podrían utilizar un enlace de corto alcance para comunicarse, dada la restricción de comunicación establecida por el modelo.

   Estas desventajas deben ser consideradas a la hora de diseñar sistemas de pago electrónico basados en este modelo.

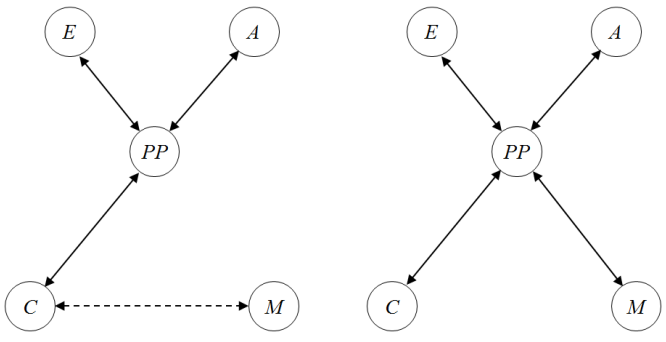


Figura 1.2: Modelo Centrado en el Cliente (izquierda) y Modelo Centrado en la Pasarela de Pago (derecha).

Por lo expuesto antes, esta tesis pretende solventar la carencia actual en el mundo científico de investigaciones previas que contemplen el problema de pago en línea desde dispositivos móviles y en escenarios de conectividad restringida, a través del diseño y desarrollo de protocolos de pago que preserven el bajo poder computacional requerido por los dispositivos móviles y que se ajusten a escenarios con restricciones de comunicación (donde dos de las entidades que los conforman no pueden intercambiar información de manera directa y deben hacerlo a través de otra entidad). Asimismo, los protocolos que resulten del desarrollo de esta tesis doctoral deben ser aplicables a otros tipos de redes como es el caso de las redes de automóviles ad-hoc (con las siglas en inglés VANETs, de Vehicular Ad-Hoc Networks) en donde existen servicios que demandan el pago en línea [19].

En este punto conviene señalar que que los nodos (vehículos) que forman las redes VANETs tienen instalado un dispositivo llamado OBU (Onboard Unit) que permite la comunicación entre el vehículo y la infraestructura de carretera. Por otra parte, las capacidades de comunicación de la OBU pueden ser utilizadas por la AU (Application Unit) que podría estar integrada a la OBU de manera permanente o podría ser una dispositivo portable (como un PDA, teléfono móvil o dispositivo de juego) que puede ser dinámicamente acoplado o desacoplado de la OBU [5]. De esta forma, los protocolos diseñados en el contexto de esta tesis doctoral pueden ser usados en las redes VANETs a través del uso de un dispositivo portable como AU.

Los protocolos que surjan del desarrollo de esta tesis doctoral, deben ofrecer los mismos niveles de seguridad que aquellos diseñados para escenarios sin restricciones de conectividad. Es por ello que los protocolos que se proponen en esta tesis deben satisfacer las siguientes propiedades de seguridad de la transacción [22, 1]:

- **Autenticación de las Entidades:** cada entidad del sistema debe poder autenticar a la entidad que intenta comunicarse con ella.
- **Privacidad de la Transacción:** cada entidad involucrada debe ser capaz de asegurar que los mensajes son revelados al destinatario previsto y no a una entidad no autorizada.
- **Integridad de la Transacción:** cada entidad involucrada puede asegurar que los mensajes recibidos no son alterados durante la transmisión.
- **No-repudio de Transacciones:** cada entidad involucrada no puede negar las transacciones que ella ha ejecutado.

## 1.2. Resumen de la tesis

La aparición de las redes móviles e inalámbricas hizo posible la extensión del comercio electrónico a un nuevo uso y área de investigación: el comercio electrónico móvil (llamado m-comercio y que incluye el pago móvil) que se refiere a cualquier transacción de comercio electrónico realizada desde un dispositivo móvil y usando redes inalámbricas. Los sistemas de pago móviles existentes en la literatura, en su mayoría, están basados en escenarios de conectividad completa donde las entidades se pueden comunicar entre sí de manera directa pero

no admiten modelos de negocio donde las restricciones de comunicación directa entre entidades del sistema no es un impedimento para realizar las transacciones comerciales. Es por ello que se requieren de sistemas de pago móvil que consideren aquellas situaciones en las cuales la comunicación directa entre dos entidades del sistema no es posible (de manera temporal o permanente) básicamente por la imposibilidad de una de las entidades de conectarse a Internet. Con el objetivo de solventar la carencia actual en el mundo científico de investigaciones previas que contemplen el problema de pago en línea desde dispositivos móviles y en escenarios de conectividad restringida, en esta tesis se desarrollan un conjunto de protocolos de pago seguro (que utilicen tanto criptografía simétrica como criptografía asimétrica no-tradicional) que preservan el bajo poder computacional requerido por los dispositivos móviles, ajustarse a escenarios con restricciones de comunicación (donde dos de las entidades que los conforman no pueden intercambiar información de manera directa y deben hacerlo a través de otra entidad) y ofrecer los mismos niveles de seguridad que aquellos diseñados para escenarios de conectividad completa. Los protocolos propuestos son aplicables a otros tipos de redes, como es el caso de las redes de automóviles ad-hoc (con las siglas en inglés VANETs, de Vehicular Ad-Hoc Networks) en donde existan servicios que demandan el pago en línea y escenarios con restricciones de comunicación. Por otra parte, la implementación, en un lenguaje de programación multiplataforma, realizada de los protocolos diseñados demuestra que el rendimiento de los mismos es adecuado para dispositivos con capacidad computacional reducida.

## 1.3. Objetivos de la Tesis

A fin de responder a las necesidades presentadas anteriormente, se hace necesario disponer de protocolos de pago para dispositivos móviles que contemplen aquellos escenarios en los cuales dos de las entidades del modelo operacional aunque no se puedan comunicar de manera directa pero que puedan mantenerse los mismos niveles de seguridad de aquellos protocolos diseñados para escenarios donde las entidades se pueden comunicar con otras sin restricciones.

La presente tesis tiene por objetivo principal, ***desarrollar protocolos de pago seguros para escenarios de conectividad restringida que preserven el bajo poder computacional requerido por los dispositivos móviles y que estimulen el desarrollo de aplicaciones que respondan a las nuevas realidades y exigencias del comercio móvil (m-comercio)***.

A fin de alcanzar el objetivo principal expuesto anteriormente, se proponen un conjunto de objetivos parciales que se analizan en profundidad a continuación:

1. **Estudio de la evolución del Pago Electrónico desde dispositivos móviles**

   La revisión del estado del arte de los sistemas de pago electrónico desde dispositivos móviles, constituye un aspecto importante en el desarrollo de esta tesis ya que permite estudiar la evolución de los sistemas de pago móviles existentes en la literatura y su adecuación a escenarios con restricciones de conectividad.

Adicionalmente, el estudio anterior permitirá determinar las bondades y deficiencias de cada una de las propuestas de los sistemas de pago electrónico desde dispositivos móviles más relevantes y que deberán ser tomadas en consideración para el diseño de los protocolos que se proponen en esta tesis doctoral.

2. **Diseño de Protocolos de Pago Seguros para escenarios de conectividad restringida**

   Partiendo del análisis de los nuevos modelos operacionales que detallan: a) las entidades que los conforman y sus relaciones, b) las técnicas criptográficas recomendadas para asegurar la comunicación entre las entidades, c) los métodos de autenticación a utilizar en el presente y en el futuro, y d) la infraestructura de hardware de cada entidad y las restricciones de comunicación; se establecen las especificaciones de los protocolos de pago a diseñar para cada uno de los escenarios de conectividad restringida presentados en la parte introductoria de este documento.

   Los protocolos diseñados deben garantizar que requieren de bajo poder computacional para su ejecución en las entidades que integran el modelo operacional (especialmente en aquellas que utilicen dispositivos móviles con limitaciones) y ofrecer equivalentes niveles de seguridad que los protocolos de pago diseñados para escenarios de comunicación completa. El último aspecto será clave para generar la confianza suficiente que estimule la creación de aplicaciones (en el mundo real) que utilicen los protocolos diseñados en esta tesis doctoral y ayudar a disminuir el escepticismo que pueda existir en la actualidad respecto a la aplicabilidad de los mismos.

3. **Implementación y Evaluación de los Protocolos de Pago diseñados**

   A fin de validar la factibilidad de los protocolos de pago diseñados en esta tesis doctoral, se ha desarrollado un sistema de pago móvil basado en los protocolos propuestos. Este sistema de pago debe ser codificado en un lenguaje de programación multiplataforma que permita, sin realizar cambios, ejecutarlo en diversos sistemas operativos.

   La implementación del sistema de pago permitirá evaluar el rendimiento de los protocolos diseñados (más allá del análisis teórico realizado durante la fase de diseño) a fin de demostrar que los mismos son adecuados para redes móviles inalámbricas. Por otra parte, la implementación convierte la propuesta teórica de esta tesis en una propuesta aplicable al mundo real.

4. **Aplicación de los Protocolos a otros entornos con restricciones de comunicación**

   Finalmente, reviste de importancia demostrar las posibilidades de aplicabilidad de los protocolos diseñados en el marco de esta tesis a otros

ambientes distintos para los cuales fueron creados pero donde se presenten situaciones similares de conectividad restringida.

Las redes VANETs representan un área emergente de investigación en la que existen escenarios de conectividad restringida donde se requieren aplicaciones seguras, especialmente en el campo de pago electrónico [19]. Esto representa una posibilidad cierta de aplicar los protocolos desarrollados a este tipo de redes, demostrando que el diseño de los mismos es flexible y con posibilidades de adaptabilidad a diversos entornos con características similares.

# Capítulo 2

# Resultados Obtenidos

## 2.1. Discusión de los Resultados Obtenidos

A lo largo del proceso de investigación que envuelve el desarrollo de la Tesis, se han logrado aportes que han sido validados por la comunidad científica internacional a través de la evaluación de artículos enviados a congresos y/o revistas internacionales de reconocido prestigio. Cada publicación y sus aportes se describen brevemente en los siguientes párrafos y se añaden de manera completa en los anexos de este documento.

El artículo titulado ***"Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices"***, publicado en el libro del memorias de la conferencia internacional **Computational Science and Its Applications, ICCSA 2006 (Lecture Notes in Computer Science, Factor de Impacto (ISI): 0.402)** [15], tiene por objetivo diseñar un protocolo seguro de pago (de ahora en adelante referido como *KCMS*, de Kiosk Centric Model con criptografía simétrica) para un escenario donde el cliente no puede comunicarse de manera directa con el emisor (modelo centrado en el comercio) debido a la ausencia de acceso a Internet en su dispositivo móvil. El protocolo utiliza operaciones de clave simétrica, que requieren bajo poder computacional, y una conexión de corto o mediano alcance (como Bluetooth, Infrarojo o Wi-Fi) entre el cliente y el comercio.

Los resultados obtenidos demuestran que se pueden realizar pagos de manera segura aún cuando el cliente no pueda comunicarse en forma directa con su banco, dada la restricción del cliente de conectarse a Internet desde su dispositivo móvil y la imposibilidad de implementar otros tipos de mecanismos de comunicación (como SMS, teléfono, etc.). Estos supone que el cliente debe utilizar al comercio como intermediario para lograr comunicarse con el emisor.

Por otra parte, el análisis cuantitativo del rendimiento del protocolo KCMS muestra que pese a la restricción de comunicación mencionada anteriormente, las transacciones de pago se pueden realizar con un rendimiento aceptable.

A pesar de las ventajas que ofrecen las operaciones de clave simétrica para las redes inalámbricas (comprobado en la publicación anterior), el manejo de las claves es complejo debido a que la clave secreta compartida debe ser acordada por las entidades y cualquier participante debe mantener $n$ número de claves

secretas (una por cada entidad con quien se establece una comunicación).

Precisamente, la criptografía asimétrica surgió con el objetivo de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos. Sin embargo, este tipo de criptografía requiere alto poder computacional y el uso de certificados para evitar el conocido problema de autenticación donde un impostor podría suplantar a un usuario con una clave pública válida pero incorrecta (porque la clave no pertenece al usuario [3]).

El certificado de clave pública debe ser verificado frente una Autoridad Certificadora (CA), lo que causa intercambios adicionales de información durante una transacción. Lo anterior constituye un problema para esquemas con restricciones de conectividad donde una entidad no puede comunicarse con la CA para verificar el certificado. En consecuencia, se requiere el uso de esquemas de firma digital no-tradicionales aplicables a escenarios con este tipo de restricciones de comunicación.

Los esquemas de firma digital con recuperación de mensaje usando claves públicas auto-certificadas [7, 3] proporcionan esquemas de cifrado y autenticación que integran los mecanismos de cifrado y firma, lo que permite que sólo al receptor especificado, verificar y recuperar el mensaje original. La autenticación de la clave pública puede ser lograda implícitamente con la verificación de la firma, sin el uso de un certificado emitido por una CA. Adicionalmente, dado que las claves públicas son calculadas tanto por el usuario como por la autoridad del sistema (llamada SA y la encargada de generar los parámetros del sistema en la fase de inicialización del sistema), se reduce los requisitos computacionales y en consecuencia, se pueden utilizar en dispositivos móviles.

En el artículo titulado ***"Anonymous payment in a Kiosk centric model using digital signature scheme with message recovery and low computational power devices"***, publicado en la revista **Journal of Theoretical and Applied Electronic Commerce Research (CORE Quality Rating: C)** [21], se propone un protocolo de pago anónimo (referido de ahora en adelante como *KCMA*, de Kiosk Centric Model con criptografía asimétrica no tradicional) para el mismo escenario de conectividad restringida utilizado en el artículo anterior, con la diferencia (y que constituye el aporte del artículo) que por primera vez se utiliza un esquema de firma digital con recuperación de mensaje usando claves públicas autocertificadas en un protocolo de pago. Esto permitió demostrar que se pueden usar operaciones de clave asimétricas en escenarios con restricciones d e comunicación sin necesidad de comunicación con una autoridad certificadora (CA) para verificar la validez de un certificado.

Una vez que se diseñaron los dos protocolos (usando dos técnicas criptográficas diferentes, de acuerdo a los objetivos de esta tesis doctoral) para el escenario de conectividad restringida centrado en el comercio, correspondió hacer los mismo pero el escenario centrado en el cliente. En este sentido, se publicó en el libro de memorias del evento **18th International Workshop on Database and Expert Systems Application, DEXA 2007 (Factor de Impacto (Citeseer): 0.27)** el artículo titulado ***"Anonymous Account-Based Mobile Payment Protocol for a Restricted Connectivity Scenario*** [9]. En este artículo se propone un protocolo de pago anónimo (referido de ahora en adelante como *CCMS*, de Client Centric Model con criptografía simétrica) para un sistema de pago basado en el escenario centrado en el cliente, utilizando operaciones

de clave simétrica que requieren bajo poder computacional. Por otra parte, el cliente se convierte en un intermediario para permitir la comunicación entre el comercio y su banco (adquiriente, en inglés **Acquirer**) de manera segura.

Los resultados demuestran que el comercio puede vender bienes y/o servicios de manera segura incluso sin poderse comunicar de manera directa con su banco. Por otra parte, el funcionamiento de este protocolo supone un aumento entre el intercambio de mensajes entre las entidades que lo conforman y mayor poder de cómputo con respecto al protocolo KCMS.

El artículo titulado ***"Anonymous Payment in a Client Centric Model for Digital Ecosystems"***, publicado en el libro de memorias de la conferencia **IEEE International Conference on Digital Ecosystems and Technologies, IEEE-DEST 2007 (CORE Quality Rating: C)** [12], propone un protocolo de pago anónimo para sistemas de pago basados en el modelo centrado en el cliente (referido de ahora en adelante como CCMA), utilizando un esquema de firma digital con recuperación de mensaje usando claves públicas autocertificadas a un protocolo de pago. Como resultado, el trabajo demuestra que el vendedor puede vender sus productos de manera segura aún cuando no pueda comunicarse directamente con su banco para validar el pago realizado por el cliente. Por otra parte, el hecho de que debe esperar a que la aprobación del pago sea reenviada por el Cliente no supone un riesgo para el vendedor ya que éste no entregará los bienes comprados hasta que reciba la autorización de pago.

Culminado el diseño de los dos protocolos (usando dos técnicas criptográficas diferentes) para el escenario de conectividad restringida centrado en el cliente, se elaboró el artículo titulado ***"A Secure Payment Protocol for Restricted Connectivity Scenarios in M-Commerce "***, que fue aceptado para ser publicado en el libro de memorias de la conferencia internacional **8th International Conference on Electronic Commerce and Web Technologies, EC-Web'07, (Lecture Notes in Computer Science, CORE Quality Rating: B, Factor de Impacto (ISI): 0.402, Factor de Impacto (Citeseer): 0.29)** [10]. El principal aporte de este artículo es la propuesta de un protocolo de pago seguro que funciona en dos escenarios de conectividad restringida (tanto para el modelo centrado en el cliente como para el modelo centrado en el vendedor), utilizando el esquema de firma digital con recuperación de mensaje usando claves públicas autocertificadas. De esta forma, independientemente del escenario en donde ocurra el pago móvil, el usuario lo podrá realizar desde su dispositivo móvil de manera segura. Dado que el protocolo funciona tanto para un escenario donde el cliente puede hacer uso de Internet como para donde no lo tiene disponible, el dispositivo móvil del cliente debe tener la capacidad de poder conectarse a Internet aún cuando el escenario donde ocurra el pago no permita su uso.

En los artículos publicados previamente, los protocolos propuestos fueron evaluados a fin de mostrar que pese a las limitaciones de comunicación que imponen los modelos operaciones utilizados para sus diseños, son capaces de realizar transacciones con rendimientos similares a la de protocolos de pago existentes ([2, 16, 18]). Sin embargo, tal cual sucede con otros protocolos de pago existentes en la literatura, sin una implementación concreta es imposible asegurar que los protocolos diseñados ofrecen ventajas sobre otros y que pueden ser utilizados en aplicaciones del mundo real.

En virtud de lo expuesto antes, se realizó el artículo titulado ***"Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks"***, publicado en la revista **Electronic Commerce Research (Springer, CORE Quality Rating: A, Factor de Impacto (JCR): 0.552)** [13]. Este artículo propone una versión mejorada (tomando en consideración los comentarios realizados por los revisores de la primera versión que fue presentada en ICSSA 2006) del protocolo KCMS que incluye la protección de la identidad real del cliente (a través de una técnica de anonimato). El objetivo principal de este artículo es demostrar que el protocolo KCMS puede ser ejecutado en dispositivos móviles de diferentes capacidades para realizar transacciones de pago en línea, con rendimientos adecuados para redes inalámbricas y redes VANETs. Para ello, se implementó el protocolo KCMS en JAVA ME para el cliente (tanto para un teléfono móvil -Nokia $^{TM}$ N95- y para un PDA -Palm$^{TM}$ T|X-) y JAVA SE para las otras entidades que forman parte del protocolo de pago.

Los resultados empíricos obtenidos de la evaluación de rendimiento de la implementación del protocolo KCMS demostraron que una transacción de pago puede ser completada con un promedio de **6.84 segundos** utilizando un dispositivo Nokia $^{TM}$ N95 y **5.66 segundos** usando un dispositivo Palm$^{TM}$ T|X. Por otra parte, el tamaño del programa que contiene la implementación del protocolo y que se utiliza en el cliente es pequeño lo que permite que pueda ser usado en dispositivos con capacidades de almacenamiento reducidas. En el caso del dispositivo Nokia, se requieren **68 kilobytes** para almacenar el programa mientras que **123 kilobytes** para ser almacenado en el dispositivo Palm Pilot.

Una vez que se diseñaron todos los protocolos para los escenarios restringidos descritos al principio en párrafos anteriores, se hizo necesario demostrar la posibilidad de aplicabilidad de los protocolos diseñados en esta tesis a otros ambientes distintos para los cuales fueron creados pero donde se presenten situaciones de conectividad restringida. En tal sentido, las redes VANETs representan un área emergente de investigación en la que existen escenarios de conectividad restringida y donde se requieren aplicaciones seguras, especialmente en el campo de pago electrónico [19]. Lo anterior evidencia la posibilidad cierta de aplicar los protocolos desarrollados a las redes VANETs, demostrando así que el diseño de los mismos es flexible y tiene posibilidades de crecer en el tiempo.

El artículo titulado ***"A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks"***, publicado en la revista **Computer Communications (Elsevier, Factor de Impacto (JCR): 0.933)** [11], tiene por objetivo demostrar que el protocolo KCMA puede ser utilizado para pagos seguros en línea en escenarios restringidos vehículo-a-carretera en VANETs, utilizando un enlace de corto alcance y sin revelar información privada.

Los resultados permiten demostrar la viabilidad de aplicación del protocolo KCMA a aquellos escenarios en VANETs en los cuales existen restricciones de conectividad. Además, se demostró que el protocolo es resistente a los ataques de repetición y suplantación. De esta forma, se contribuye a expandir las posibilidades de pago en-línea en VANETs.

En el contexto de la investigación de VANETs, se hacía necesario investigar los posibles ataques que podrían comprometer la seguridad de este tipo de redes

y su incidencia en los protocolos de pago diseñados en el contexto de esta tesis. Así, surge el artículo titulado ***"Security attacks and solutions for vehicular ad hoc networks"***, publicado en la revista **IET Communications** (IET Research Journals, Factor de Impacto (JCR): 0.751) [14]. El objetivo de este artículo discutir las principales amenazas y ataques que pueden ser explotados en VANETs y presentar las soluciones de seguridad correspondientes que pueden ser implementadas para frustrar esos ataques.

A través de los resultados de la investigación se identificaron problemas de seguridad en los siguiente aspectos: anonimato, manejo de claves, privacidad, localización y reputación. Por otra parte, se identificaron las correspondientes soluciones de seguridad reportados recientemente en la literatura.

Este artículo obtuvo el premio **IET Premium Award for Communications 2011** que se otorga a los mejores artículos publicados durante el período 2009-2010. Dicho premio se entregará el próximo 9 de noviembre del presente año en el evento: *2011 IET Ambition and Achievement Awards Ceremony.*

# Capítulo 3

# Indicios de Calidad

## 3.1. Indicios de Calidad de las Publicaciones

El trabajo de investigación presentado en esta tesis doctoral ha sido validado internacionalmente en diferentes revistas y conferencias de seguridad y comercio electrónico donde los expertos han proporcionado sus valiosos comentarios y reflexiones que han permitido mejorar nuestras investigaciones. Las publicaciones realizadas durante el desarrollo de la tesis en revistas arbitradas y conferencias internacionales en las Tablas 3.1 y 3.2. En ambas tablas, se muestra información de los indicios de calidad de cada una de ellas.

| **Año** | **Título del Artículo** | **Revista** | **Indicios de Calidad** |
|---|---|---|---|
| 2006 | Anonymous payment in a Kiosk centric model using digital signature scheme with message recovery and low computational power devices [21] | Journal of Theoretical and Applied Electronic Commerce Research | CORE Quality Rating: C |
| 2010 | Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks [13] | Electronic Commerce Research | CORE Quality Rating: A<br>SCImago Ranking Q1 (SJR): 0.04<br>Factor de Impacto (JCR): 0.552 |
| 2008 | A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks [11] | Computer Communications | Factor de Impacto (JCR): 0.933 |
| 2010 | Security attacks and solutions for vehicular ad hoc networks [14] | IET Communications | Factor de Impacto (JCR): 0.751<br>Premio: **IET Premium Award for Communications 2011** |

Tabla 3.1: Indicios de Calidad de las Publicaciones realizadas en Revistas Arbitradas durante el desarrollo de la Tesis Doctoral.

| Año | Título del Artículo | Revista | Indicios de Calidad |
|---|---|---|---|
| 2006 | Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices [15] | Computational Science and Its Applications | Factor de Impacto (ISI): 0.402 |
| 2007 | Anonymous Account-Based Mobile Payment Protocol for a Restricted Connectivity Scenario [9] | 18th International Workshop on Database and Expert Systems Application | Factor de Impacto (Citeseer): 0.27 |
| 2007 | Anonymous Payment in a Client Centric Model for Digital Ecosystems [12] | IEEE International Conference on Digital Ecosystems and Technologies | CORE Quality Rating: C |
| 2007 | A Secure Payment Protocol for Restricted Connectivity Scenarios in M-Commerce [10] | 8th International Conference on Electronic Commerce and Web Technologies | CORE Quality Rating: B<br>Factor de Impacto (ISI): 0.402<br>Factor de Impacto (Citeseer): 0.29 |

Tabla 3.2: Indicios de Calidad de las Publicaciones realizadas en Conferencias Internacionales durante el desarrollo de la Tesis Doctoral.

# Capítulo 4

# Conclusiones

## 4.1. Conclusiones Finales

En el marco de esta tesis doctoral, fueron propuestos un conjunto de protocolos de pago, basados en cuenta (Account-based, en inglés), adecuados para los escenarios de conectividad restringida presentados en la parte introductoria de este documento y que funcionan eficientemente en ambientes inalámbricos. Para cada escenario, se diseñaron dos protocolos: uno que utiliza operaciones de criptografía simétrica en todas las entidades y otro que emplea operaciones de criptografía a simétrica no tradicional.

Los protocolos diseñados demostraron la posibilidad cierta de realizar transacciones de pago seguras en ambientes con restricciones de comunicación (donde una entidad no se puede comunicar de manera directa con otra), ofreciendo los mismos niveles de seguridad que aquellos basados en los escenarios de conectividad completa . Para superar dicha restricción sin el uso de intermediarios externos (que podrían incorporar nuevos elementos de seguridad), se utiliza cualquier entidad del protocolo como intermediario para permitir el envío de mensajes entre las dos entidades que no puedan comunicarse entre sí de manera directa debido a restricciones de comunicación.

El análisis comparativo de rendimiento realizado a los protocolos propuestos en esta tesis, centrado particularmente en el número de operaciones criptográficas realizadas por cada entidad involucrada en cada protocolo, permitió conocer las ventajas y desventajas de los protocolos propuestos con respecto a otros protocolos de pago existentes en la literatura (SET [18], 3KP[1] [2], KSL [17] y LMPP [6]).

El análisis del número de operaciones criptográficas realizadas por cada entidad involucrada (ver tabla 4.1), muestra que los protocolos KCMS (Kiosk Centric Model, con criptografía simétrica) y PCMS (Payment Gateway Centric Model, con criptografía simétrica), a pesar de haber sido diseñado para escenarios con diferentes restricciones de conectividad, en ambos casos el cliente ejecuta en mismo número de operaciones criptográficas. Sin embargo, en el caso del protocolo CCMS (Client Centric Model, con criptografía simétrica), el

[1]El protocolo 3KP es usado en nuestra comparación ya que es el único de la familia iKP que proporciona el mismo nivel de seguridad que el protocolo SET.

cliente ejecuta un número mayor de operaciones criptográficas debido a que esta entidad es la encargada de permitir el intercambio de mensajes entre el comercio y la pasarela de pago (debido a las restricciones impuestas por el escenario para el cual fue diseñado este protocolo). En consecuencia, los requisitos computacionales del protocolos CCMS son mayores con respecto a los protocolos KCMS y PCMS.

Por otra parte, los protocolos diseñados KCMS, CCMS y PCMS tienen rendimientos y servicios de seguridad similares a los protocolos SET, 3KP, KSL y LMPP pese a las restricciones de comunicación impuestas por los modelos operacionales para los cuales fueron diseñados.

En cuanto a los protocolos KCMA (Kiosk Centric Model, con criptografía asimétrica no tradicional) y CCMA (Client Centric Model, con criptografía asimétrica no tradicional), el cliente ejecuta en mismo número de operaciones criptográficas. Sin embargo, en el caso del protocolo CCMA, el cliente ejecuta un número mayor de operaciones criptográficas debido a que esta entidad actúa como intermediario entre el comercio y la pasarela de pago para el intercambio de mensajes. En consecuencia, los requisitos computacionales del protocolos CCMA son mayores con respecto a los protocolos KCMA y PCMA (Payment Gateway Centric Model, con criptografía asimétrica no tradicional).

Pese a que en algunos casos los protocolos KCMA, CCMA y PCMA realizan un número mayor de operaciones criptográficas que otros protocolos (que utilizan criptografía asimétrica) existentes en la literatura (como SET y 3KP), los requisitos computacionales son similares o menores debido a que las claves públicas son calculadas entre la autoridad del sistema y el usuario. Esto permite comprender que la criptografía asimétrica no tradicional es una alternativa cierta para ser utilizada en dispositivos móviles ya que permite evitar los problemas de rendimiento.

La implementación realizada al protocolo KCMS, permite asegurar que los protocolos diseñados ofrecen ventajas sobre otros existentes en la literatura y que pueden ser utilizados en aplicaciones comerciales. Además, el uso de un teléfono móvil y una PDA como dispositivos del lado del cliente en las implementaciones, demostró que los protocolos propuestos pueden ser ejecutados en dispositivos móviles de diferentes capacidades y con rendimientos adecuados para redes inalámbricas. Adicionalmente, el tamaño relativamente pequeño de la aplicación de pago permite que sea instalada en dispositivos móviles con capacidades limitadas de almacenamiento como los teléfonos móviles.

Finalmente, la aplicación de los protocolos KCMS y KCMA a las redes VANETs (cuyos nodos poseen suficiente suficiente poder computacional pero que pueden utilizar dispositivos móviles de bajo poder computacional para comunicarse), donde existen escenarios de conectividad restringida que requieren aplicaciones de pago seguras, demostró que el diseño de los protocolos es flexible y puede ser adaptado a diversos entornos con características similares.

Esta tesis doctoral deja abierta la posibilidad del desarrollar en el futuro, nuevas investigaciones que permitan avanzar en el campo de los protocolos para escenarios de conectividad restringida. Entre las investigaciones futuras se pueden mencionar las siguientes:

- Estudio de las curvas elípticas como técnica criptográfica para los proto-

colos propuestos en esta tesis doctoral y su impacto en el rendimiento y requisitos computacionales de los mismos.

- Desarrollo de una lógica formal para analizar la propiedad de responsabilidad (en inglés, Accountability) de protocolos para escenarios de conectividad restringida. Esta lógica formal debe ser capaz de analizar los mensajes cifrados con criptografía simétrica como con criptografía asimétrica (incluyendo la no tradicional).

- Rediseño de los protocolos propuestos para permitir que los mensajes puedan ir desde un vehículo (cliente) hasta el equipo al borde de la carretera (el comercio o la pasarela de pago), pasando por varios vehículo pero en un escenario de conectividad restringida.

| **Operaciones Criptográficas** | | **Número de Operaciones Criptográficas** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **KCMS** | **CCMS** | **PCMS** | **KCMA** | **CCMA** | **PCMA** | **SET** | **iKP** | **KSL** | **LMPP** |
| Cifrado con Clave Pública | **C** | - | - | - | - | - | - | 1 | 1 | - | - |
| | **M** | - | - | - | - | - | - | 1 | - | - | - |
| | **PP** | - | - | - | - | - | - | 1 | - | - | - |
| Descifrado con Clave Pública | **C** | - | - | - | - | - | - | - | - | - | - |
| | **M** | - | - | - | - | - | - | 1 | 4 | - | - |
| | **PP** | - | - | - | - | - | - | 2 | 1 | - | - |
| Firma Digital | **C** | - | - | - | 2 | 4 | 2 | 1 | 1 | - | - |
| | **M** | - | - | - | 3 | 2 | 2 | 3 | 1 | - | - |
| | **PP** | - | - | - | 1 | 1 | 3 | 1 | 1 | - | - |
| Verificación de la Firma Digital | **C** | - | - | - | - | - | - | - | 2 | 3 | - |
| | **M** | - | - | - | - | - | - | - | 2 | 2 | - |
| | **PP** | - | - | - | - | - | - | - | 1 | 2 | - |
| Cifrado/Descifrado con clave Simétrica | **C** | 3 | 5 | 3 | - | - | - | 1 | - | 4 | 5 |
| | **M** | 5 | 4 | 4 | - | - | - | - | - | 5 | 6 |
| | **PP** | 2 | 3 | 3 | - | - | - | 1 | - | 2 | - |
| Función Resumen ($H$) | **C** | 2 | 2 | 2 | 1 | 1 | 1 | 3 | 2 | 2 | 2 |
| | **M** | 1 | 1 | 1 | 1 | 1 | - | 2 | 4 | - | 1 |
| | **PP** | - | 3 | - | - | - | - | - | 1 | - | - |
| Función Resumen con clave ($KH$) | **C** | 2 | 3 | 2 | - | - | - | - | - | 2 | - |
| | **M** | 1 | 2 | 1 | - | - | - | - | 1 | 2 | - |
| | **PP** | - | - | - | - | - | - | - | - | 1 | - |
| Generación de Clave | **C** | 3 | 4 | 2 | - | - | - | - | - | 2 | - |
| | **M** | 4 | 4 | 4 | - | - | - | - | - | 1 | - |
| | **PP** | 2 | 2 | 3 | - | - | - | - | - | 1 | - |
| Autenticación y recuperación del mensaje recibido | **C** | - | - | - | 2 | 2 | 2 | - | - | - | - |
| | **M** | - | - | - | 2 | 1 | 2 | - | - | - | - |
| | **PP** | - | - | - | 1 | 1 | 1 | - | - | - | - |

Tabla 4.1: Número de operaciones criptográficas utilizadas por los protocolos KCMA, CCMA, PCMA, SET [18], iKP [2], KSL [17] y LMPP [6] en el cliente, el comercio y la pasarela de pago.

# Bibliografía


[1] N. Asokan, Philippe Janson, Michael Steiner, and Michael Waidner. State of the art in electronic payment systems. *Advances in Computers*, 53:426–451, 2000.

[2] Mihir Bellare, Juan A. Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Els Van Herreweghen, and Michael Waidner. Design, implementation, and deployment of the ikp secure electronic payment system. *IEEE Journal of Selected Areas in Communications*, 18(4):611–627, 2000.

[3] Ya-Fen Chang, Chin-Chen Chang, and Hui-Feng Huang. Digital signature with message recovery using self-certified public keys without trustworthy system authority. *Applied Mathematics and Computation*, 161(1):211–227, 2005.

[4] Suresh Chari, Parviz Kermani, Sean Smith, and Leandros Tassiulas. Security issues in m-commerce: A usage-based taxonomy. In *E-Commerce Agents, Marketplace Solutions, Security Issues, and Supply and Demand*, pages 264–282, 2001.

[5] Car2Car Communication Consortium. Overview of the c2c-cc system, technical report version 1.0, car2car communication consortium. Technical report, 2007.

[6] Tan S. Fun, Leau Y. Beng, J. Likoh, and R. Roslan. A lightweight and private mobile payment protocol by using mobile network operator. In *International Conference on Computer and Communication Engineering*, pages 162–166, 2008.

[7] Marc Girault. Self-certified public keys. In *EUROCRYPT*, pages 490–497, 1991.

[8] Ren-Junn Hwang, Sheng-Hua Shiau, and Ding-Far Jan. A new mobile payment scheme for roaming services. *Electronic Commerce Research and Applications*, 6(2):184191, 2007.

[9] Jesús Téllez Isaac and José Sierra Camara. An anonymous account-based mobile payment protocol for a restricted connectivity scenario. In *18th International Workshop on Database and Expert Systems Applications(DEXA 2007)*, pages 688–692, 2007.

[10] Jesús Téllez Isaac and José Sierra Camara. A secure payment protocol for restricted connectivity scenarios in m-commerce. In *E-Commerce and Web Technologies, 8th International Conference (EC-Web 2007)*, pages 1–10, 2007.

[11] Jesús Téllez Isaac, José Sierra Camara, Sherali Zeadally, and Joaquín Torres Márquez. A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. *Computer Communications*, 31(10), 2008.

[12] Jesús Téllez Isaac and José Sierra Cámara. Anonymous payment in a client centric model for digital ecosystems. In *Inaugural IEEE International Conference on Digital Ecosystems and Technologies (IEEE-DEST 2007)*, pages 422–427, 2007.

[13] Jesús Téllez Isaac, Sherali Zeadally, and José Sierra Camara. Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks. *Electronic Commerce Research*, 10(2):209–233, 2010.

[14] Jesús Téllez Isaac, Sherali Zeadally, and José Sierra Cámara. Security attacks and solutions for vehicular ad hoc networks. *IET Communications*, 4(7):894–903, 2010.

[15] Jesús Téllez Isaac, José Sierra Camara, Antonio Izquierdo Manzanares, and Mildrey Carbonell Castro. Payment in a kiosk centric model with mobile and low computational power devices. In *International Conference of Computational Science and Its Applications (ICCSA 2006)*, pages 798–807, 2006.

[16] Supakorn Kungpisdan. *Design and Analysis of Secure Mobile Payment Systems*. PhD thesis, Monash University, 2005.

[17] Supakorn Kungpisdan, Bala Srinivasan, and Phu Dung Le. A secure account-based mobile payment protocol. In *International Conference on Information Technology: Coding and Computing (ITCC'04)*, pages 35–39, 2004.

[18] Mastercard and Visa. Set protocol specifications book 1-3, 1997.

[19] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *3rd ACM workshop on Security of ad hoc and sensor networks (SASN 2005)*, pages 11–21, 2005.

[20] Artur Romão and Miguel Mira da Silva. An agent-based secure internet payment system for mobile computing. In *International IFIP/GI Working Conference on Trends in Distributed Systems for Electronic Commerce (TREC'98)*, pages 80–93, 1998.

[21] Jesús Téllez, José Sierra, Antonio Izquierdo, and Mildrey Carbonell. Anonymous payment in a kiosk centric model using digital signature scheme with message recovery and low computational power devices. *Journal of Theoretical and Applied Electronic Commerce Research*, 1(2):1–11, 2006.

[22] Upkar Varshney. Mobile payments. *IEEE Computer*, 35(12):120–121, 2002.

[23] Xiao Feng Wang, Kwok-Yan Lam, and Xun Yi. Secure agent-mediated mobile payment. In *First Pacific Rim International Workshop on Multi-Agents (PRIMA 1998)*, pages 162–173, 1999.

[24] Xun Yi, Chee Kheong Siew, Xiao Feng Wang, and Eiji Okamoto. A secure agent-based framework for internet trading in mobile computing environments. *Distributed and Parallel Databases*, 8(1):85–117, 2000.

# Anexos

# Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices*

Jesús Téllez Isaac, José Sierra Camara,
Antonio Izquierdo Manzanares, and Mildrey Carbonell Castro

Universidad Carlos III de Madrid, Computer Science Deparment,
Avda. de la Universidad, 30, 28911, Leganés (Madrid), Spain
jtellez@gmail, {sierra, aizquier}@inf.uc3m.es,
mildreycc@yahoo.es

**Abstract.** In this paper we present a protocol for a mobile payment system based on a Kiosk Centric Model (proposed by [2]) that employs symmetric-key operations which require low computational power. Our protocol is suitable for mobile payment systems where the customer cannot communicate with the issuer due to the absence of Internet access with her mobile device and the costs of implementing other mechanisms of communication between both of them are high. However, our proposal illustrates how a portable device equipped with a short range link (such Bluetooth, Infrared or Wi-Fi) and low computational power should be enough to interact with a vendor machine in order to buy goods in a secure way.

## 1 Introduction

The popularity of m-commerce has increased in the last years thanks to advances in the portable devices and the rapid development of the mobile communication technologies that have allowed people to use mobile telephones or Personal Digital Assistant (PDA) to access the Internet (to read email, browse web pages or purchase information or goods) anywhere and anytime.

Different mobile payment systems have been proposed in the last years, but the one developed by [8] (called 3-D Secure) has become a standard due to its benefits regarding security and flexibility in the authentication methods. This schema allows the authentication of the payer (customer) when she makes an online payment using a debit or credit card. The transaction flow for this scheme is shown in figure 1 where all the main communications links are protected using SSL/TLS and the communication between the issuer/consumer is mandatory.

Despite of the flexibility that 3-D Secure gives to the issuer to choose the authentication method, relationship between payer and issuer is quite strict (although required for Visa's 3D-Secure scheme) and does not allow the use of schemes in which the communication among these parties is not possible due to: 1) the impossibility of the client to connect to Internet from the mobile device

* This work was partially supported by ASPECTS-M Project (Reference Models for Secure Architectures in Mobile Electronic Payments), CICYT-2004.

M. Gavrilova et al. (Eds.): ICCSA 2006, LNCS 3984, pp. 798–807, 2006.

and 2) the high costs of the infrastructure necessary to implement other mechanisms of communication between the client and the issuer. Most of the mobile payment systems proposed up until now assume the consumer has Internet connectivity through her mobile device, so the restrictions mentioned previously do not represent an important issue. However, it is quite common that the client meets situations in which it is not possible to connect to Internet so it becomes necessary to develop mobile payment systems where the user could use her mobile device as a shopping means, even thought she may not have Internet access.

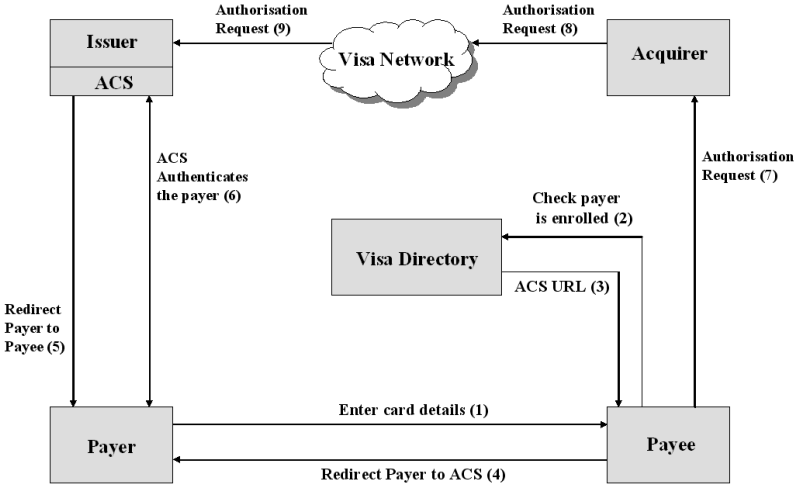


**Fig. 1.** 3-D Secure transaction [1]

On the other hand, in spite of the wide range of mobile devices available, they all have common limitations [6]: 1) poor computational capabilities, 2) limited storage space and 3) short battery life. These limitations prevent that these devices execute, in an efficient way, computations that require a lot of resources, like those of asymmetric cryptography.

Symmetric cryptography (which employs a shared key between two parties) provides, like asymmetric cryptography, message confidentiality, message integrity and party authentication, and represents an alternative in the construction of secure protocols for mobile payment systems, because symmetric-key operations do not require of a high computational power nor additional communications steps (as happens in protocols based on public-key infraestructure where the public-key certificates have to be verified by a Certificate Authority).

In this paper, we present a protocol (that supports both credit-card and debit-card transactions) for a mobile payment system based on a Kiosk Centric Model (proposed by [2]) which overcomes the limitations mentioned before. Our proposal represents an alternative to the restrictions of mobile payment systems (including Visas 3-D Secure) as for the connection between the client and issuer.

Moreover, it uses symmetric-key operations in all engaging parties to reduce both, the setup cost for payment infrastructure and the transaction cost. Another benefit derived of the using of our proposal is a reduction of all parties computation and communications steps (in comparison with protocols based on public-key infraestructure) that make it suitable for mobiles devices with low computational power.

The rest of this paper is organized as follows: In next section, we survey related work. Section 3 presents the proposed system. In section 4, we analyze the scheme proposed. We end with our conclusions in Section 5.

## 2 Related Work

In recent years, several studies have been conducted to improve the security of mobile payment systems. Meanwhile, efforts have also been dedicated to unify concepts and scenarios into frameworks that will be useful to develop new electronic payment systems. Research conducted by [2] is an example of a study that unifies many proposed m–commerce usages into a single framework. This research intended to revise the possible range of mobility scenarios, identifying the security issues for each connectivity scenario. As a result, five scenarios were identified and analyzed: Disconnected Interaction, Server Centric Case, Client Centric Case, Full Connectivity and Kiosk Centric Case. The latest has been considered as the starting point in the design of our proposal.

In [9], payment methods are classified according to several standards and analyzed to point out their advantages and drawbacks. Besides, the research also provides a payment process for mobile devices based on pre–payment and accounts. This proposed solutions requirements are low (both on cost and technical capabilities) and it also has high scalability and security properties. However, their methods and processes are not suitable for our proposal, as our goal is to suggest an scheme based on post-payment [1] and symmetric cryptography.

A secure and efficient one-way mobile payment system is proposed in [4]. In their solution the security of the system is based on the intractability of the discrete logarithm problem and the one-wayness of keyed hash function. As opposed to their goal (designing a mobile payment system with minimal complexity using two public key pairs), our solution aims for devising a scheme that relies on symmetric-key operations instead.

The closest work to ours is [5]. Their work proposed a secure account-based payment protocol suitable for wireless networks that employs symmetric-key operations which require lower computation at all engaging parties than existing payment protocols. While this proposal satisfies the majority of our requirements, we have to reformulate their protocol (from now on, SAMPP) to satisfy the requirements of the scheme that we suggest in this work, where the customer never establishes any connection with the bank during the payment transaction.

[1] Mobile payment where the consumer receives the content and consumes it before paying. Credit cards are an example of credit-based payment methods.

As the payment software (also called wallet software) must be sent to the customer by the issuer through the vendor, it becomes necessary the use of techniques to assure that the program received by the client was created and sent by the issuer, and has not been tampered. In order to obtain the protection of the payment software in the aspects mentioned before, two different proposals related to the aforementioned techniques will be detailed in te following paragraphs.

The first work (proposed by [3]) introduced a new approach to watermarking, called path based watermarking, that embeds the watermark, with relatively low cost, in the dynamic branch structure of the program , and shows how error-correcting and tamper proofing techniques can be used to make path based watermarks resilient against a wide variety of attacks. The other work, proposed by [7], describes three techniques for obfuscation of program design: 1) The class coalescing obfuscation, 2) Class splitting obfuscation, and 3) Type hiding obfuscation. The experimental results (applying theses obfuscations to a medium-size java program) shows that the run-time overhead, in the worst of the case (class splitting obfuscation), is less than 10% of the total running time of the program.

## 3 Scheme Proposed

### 3.1 Notations

- {C, V, P, I, A}: the set of customer, vendor, payment gateway, issuer and acquirer, respectively.
- $\mathrm{ID}_P$ : the identity of party $P$ that contains the contact information of $P$.
- TID: Identity of transaction that includes time and date of the transaction.
- OI: Order information (OI = {TID, $h$(OI, Price)}) where OI and Price are order descriptions and its amount.
- TC: The type of card used in the purchase process (TC=Credit, Debit).
- Stt: The status of the transaction (Stt = {Accepted, Rejected}).
- TIDReq : The request fot TID.
- VIDReq : The resquest for $\mathrm{ID}_V$.
- $\{\mathrm{M}\}_X$ : the message $M$ symmetrically encrypted with the shared key $X$.
- MAC(X,K): Message Authentication Code of the message $X$ with the key $K$.
- $h$(X): the one-way has function of the message $X$.

### 3.2 Operational Model

Generally, operational models for m-commerce found in literature involve transacion between two or more entities. Our operational model is composed of four entities: 1) *Customer*: a user who wants to buy information or goods from the vendor and has a mobile device with low computational power and equipped with a built-in display, keyboard (not necessarily with a QWERTY layout), short range link (such Infrared, Wi-Fi or Bluetooth) and capability to execute a java program, 2) *Vendor*: a computational entity (a normal web or an intelligent vending machine) that wants to sell information or goods and with wich the user

participates in a transaction, 3) *Acquirer*: the vendor's financial institution, 4) *Issuer*: the customer's financial institution, and 5) *Payment Gateway*: additional entity that acts as a medium between acquirer/issuer at banking private network side and customer/vendor at the Internet side for clearing purpose.

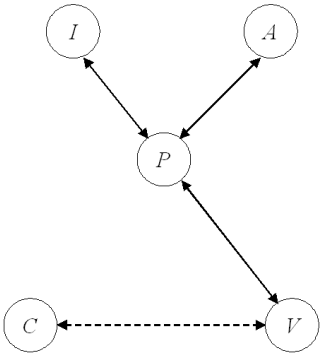


**Fig. 2.** Operational Model

In figure 2, we specify the links among the five entities of our scheme. Note that there is no direct connection involving the customer and the issuer. Moreover, the connection between the customer/vendor (denoted as the dotted arrow) is set up through a short range link (like bluetooth, infrared or Wi-Fi). On the other hand, interaction among the vendor and the payment gateway (depicted as solid arrow in the scheme) should be reliable and secure against passive and active attacks. Therefore, the connection is supposed to be established through a secure wired channel by using a security protocol like SSL/TLS [4]. Note that the issuer, acquirer and payment gateway operates under the banking private network so we do not concern about connections security among these entities.

The protocol based in symmetric cryptography proposed by [5] is a starting point of our protocol. We reformulated this protocol to satisfy the requirements of that, as stated before, pretends to allow the client to make purchases from its mobile device without connecting itself to Internet.

### 3.3 Key Generation Technique

Our scheme handles three different sets of shared keys used for encrypt a message symmetrically. Each one is generated off-line in the entity that will store them.

The first set $\text{VPSec}_j$, $j = 1, \ldots, n$, is generated from the secret VPSec and stored in the vendor and Payment gateway terminals respectively. The other set $\text{CISec}_i$ (stored in the customer's device and issuer's terminal, respectively), $i = 1, \ldots, n$, is generated from the secret CISec. The last set $\text{CVSec}_k$ (where $k = 1, \ldots, n$) is generated from the secret CVSec and are stored in the customers device and the vendors terminal respectively.

In order to generate the sets of shared keys, we apply a Hash algorithm with one-bit cyclic chain function of a master secret each time a session key is generated [5]. The details are shown as follows:

**Generating VPSec$_j$** and **CVSec$_k$**
VPSec$_1$ = $h$(1-bit-shift-of-VPSec), VPSec$_2$ = $h$(2-bit-shift-of-VPSec),...,
VPSec$_n$ = $h$(n-bit-shift-of-VPSec)
CVSec$_1$ = $h$(1-bit-shift-of-CVSec), CVSec$_2$ = $h$(2-bit-shift-of-CVSec),...,
CVSec$_n$ = $h$(n-bit-shift-of-CVSec))

**Generating CISec$_i$**
CISec$_1$ = $h$(1-bit-shift-of-(CDCI,CISec)),
CISec$_2$ = $h$(2-bit-shift-of-(CDCI,CISec)),...,
CISec$_n$ = $h$(n-bit-shift-of-(CDCI,CISec))

### 3.4 Detailed Protocols

Our protocol consists of four sub–protocols: Registration, Purchase, Withdrawal and Deposit. Each sub–protocol has the following main functions:

**Registration ($C \leftrightarrow V, C \leftrightarrow I$):** This sub–protocol involves the customer, the vendor and the issuer. The process starts when the customer shares her credit- and/or debit-card information (CDCI) with her issuer. CDCI contains the long-term secret CISec known only by the customer and her issuer and will be used as an authentication method by the customer in future withdrawals.

In addition, the secret SSWSec is shared between the customer/issuer and will be used as watermark value for the watermarking process at the issuer's side and as software input at customer's side to detect its authenticity.

When the first purchase takes place, $V$ will detect if the wallet software is available in the mobile device. If not, $V$ sends a software request to $P$, which will forward the request to $I$. The issuer intends to protect the software against various types of attacks carried away at any moment, following these steps: 1) First, choose one of the obfuscation methods proposed by [7] and apply it to the java code, and 2) Then, apply a watermarking process (proposed by [3]) to the software (using SSWSec as a watermark value and embedded into the software).

Once the software has been prepared, $I$ will forward it to the $P$, which will send it to $V$, who will finally send it to $C$. After $C$ receives the software, she will install it and check its authenticity using the secret SSWSec. If a problem occurs, $C$ could abort the registration sub–protocol or start the process again.

When the software is successfully installed and working, $C$ generates CVSec and send it to $V$ with ID$_C$ and a nonce $n$ encrypted with the session key K, generated by running AKE protocol with $V$ . Then $V$ sends $h(n,$ CVSec) to $C$ as a confirmation of customer's registration. After the sub–protocol has been completed, $C$ and $V$ can generate a new set of CVSec$_i$ by using the same key generation technique. On the other hand, the vendor registers herself to the Payment Gateway and share the secret VPSec.

**1) C → V:** $\{\text{ID}_c, \text{CVSec}, n\}_K$
**2) V → C:** $h(n, \text{CVSec})$

**Purchase ($C \leftrightarrow V$):** This sub–protocol is carried out between C and V over the wireless channel. The process starts when $C$ sends to $V$ the information necessary to set up the sub–protocol (step 3). After this information exchange ends, $C$ builds up the Payment-script Request with $OI$ and $TC$. Then, $C$ encrypts it and sends to $V$ where the message is decrypted to retrieve $OI$.

**3) C → V:** $\text{ID}_C$, i, TIDReq, VIDReq
**4) V → C:** $\{\text{TID}, \text{ID}_V\}_{CVSeci}$
**5) C → V:** $\{\text{OI}, \text{Price}, \text{MAC}[(\text{Price}, \text{TC}, h(\text{OI}), \text{ID}_V), \text{CISec}_i]\}_{CVSeci}$, $MAC[(OI, Price, ID_C, ID_I), CVSec_{i+1}]$

Note that, although V can decrypt the message using $CVSec_i$, she cannot generate this message since she does not have the necessary $\text{CISec}_i$ to construct $\text{MAC}[(\text{Price}, \text{TC}, h(\text{OI}), \text{ID}_V), \text{CISec}_i]$. Thus, any entity of the mobile payment system can ensure that the message is truly sent from $C$.

**Withdrawal (V ↔ P):** Withdrawal sub–protocol occurs between $V$ and $P$ through a secure wired channel. $V$ decrypts the message received from $C$ (to retrieve $OI$), prepares the Withdrawal-script Request (including $\text{ID}_C$, $\text{ID}_I$, and the index $i$ used to identify the current session key in the set of $\text{CISec}_i$) encrypted with $\text{VPSec}_j$ and then sends it to $P$.

After the script was received by $P$, she forwards it to $I$, adding some information such her identity ($\text{ID}_P$). Here, this script is called Withdrawal–script Request and will be processed by $I$ to approve or reject the transaction.

Once the issuer has processed the request and prepared the Withdrawal–script Response (including *Stt*), she must send it to $P$ who in turn proceeds to forward to $V$. The Deposit sub–protocol is activated by $P$ only when the Withdrawal is approved. Otherwise, $P$ assigns the value Discarded to *Std*.

After the Withdrawal and Deposit sub–protocols are completed, $P$ sends the Withdrawal-script Response to $V$(including the Deposit-script Response). Then $V$ prepares the Payment-script Response and sends it to $C$.

**6) V → P:** $\{\text{MAC}[(\text{Price}, \text{TC}, h(\text{OI}), \text{ID}_V), \text{CISec}_i], \text{j}, \text{ID}_V,$ $h(\text{OI}), \text{i}, \text{TID}, \text{Price}, \text{ID}_C, \text{ID}_I\}_{VPSecj}$, $\text{MAC}[(h(\text{OI}), \text{i}, \text{TID}, \text{ID}_C, \text{ID}_I), \text{VPSec}_{j+1}]$
**7) P → I:** $\text{MAC}[(\text{Price}, \text{TC}, h(\text{OI}), \text{ID}_V), CISec_i], \text{i},$ $h(\text{OI}),\text{TID}, \text{Price}, \text{ID}_C, \text{ID}_V, h(\text{VPSec}_{j+1})$
**8) I → P:** $\text{Stt}, h(\text{Stt}, h(\text{OI}), h(\text{CISec}_i)), \{h(\text{OI}), \text{Stt}, h(\text{VPSec}_{j+1})\}_{CISeci}$
**11) P → V:** $\{\text{Stt}, \{h(\text{OI}), h(\text{VPSec}_{j+1})\}_{CISeci},$ $h(\text{Stt}, h(\text{OI}), h(\text{CISec}_i)), \text{Std}, h(\text{Std}, h(\text{OI}))\}_{VPSecj+1}$
**12) V → C:** $\{\{h(\text{OI}), \text{Stt}, h(\text{VPSec}_{j+1})\}_{CISeci}\}_{CVSeci+1}$

**Deposit ($P \leftrightarrow A$):** This sub–protocol occurs between the $P$ and $A$ trough a secure wired channel when no problems have found at the Withdrawal sub–protocol. Here, the Deposit-script Request is prepared by $P$ who sends it to $A$

who checks the Price received with the negotiated during the purchase process. If they are matched, the value *Accepted* is assigned to *Std* and the total amount of the *OI* is transferred to the vendor's account. Otherwise, the deposit is refused (the value *Discarded* is assigned to *Std*) and it not represents an excuse for $V$ to not deliver the good to $C$ because the Withdrawal sub–protocol has been complete successfully. Then, a dispute occurs between $V$, $P$ and $A$.

The Deposit-script Response is prepared by $A$ and then sent to $P$ in order to complete the deposit sub–protocol.

**9) P → A:** $\mathrm{ID}_p$, Price, TID, Stt, $h$(OI), $\mathrm{ID}_V$, $h(\mathrm{VPSec}_{j+1})$
**10) A → P:** $\mathrm{ID}_A$, Std, $h$(Std, $h$(OI))

After a transaction is completed, each entity of the payment system put in her revocations list, $\mathrm{CVSec}_i$ and $\mathrm{CISec}_i$ to prevent their replay from customer and vendor. In the following purchases, the registration sub–protocol will not occur until the customer is notified to update the secret CVSec. Thus, when become necessary to renew the secret, the customer runs the Registration sub–protocol to get a new CVSec. While the secret is not updated, the customer can use other values in the set of CVSeci to perform transactions. To update the VPSEC, the Payment Gateway sends the new secret to the vendor by using an AKE protocol. Finally, to update the CISec, the issuer has to add a message with the new secret to the Withdrawal-script Response which will be modified as following:

$$\{h(\mathrm{OI}),\ \mathrm{Stt},\ h(\mathrm{VPSec}_{j+1}),\ \mathrm{NewSecret},\ h(\mathrm{NewSecret})\}_{CISeci}.$$

## 4 Analysis

### 4.1 Comparison with SAMPP

In this section, we present a comparison between SAMPP and ours in order to establish the differences between both protocols.

The major difference between both protocols relies on the operational environment in which they are used. In SAMPP, the mobile device has access to the Internet which allows the client to communicate with the issuer when needed whereas our protocol is based on the idea of the consumer not being able to connect directly to the issuer, in consequence, any information or program that the issuer wants to send to the client, will have to do it through the vendor.

Another difference is the distribution method used with the payment software. While in SAMPP the customer must either download the software from the issuer or receive it by e-mail, in our proposal the wallet software must be sent from the issuer to the consumer through the vendor. This has lead us to the inclusion of security mechanisms (such as code obfuscation and watermarking) that assure the software against several types of attacks.

The third difference worth mentioning can be found in the number of sub–protocols that compose the protocol. SAMPP is composed of two sub–protocols whereas ours it is made up of four sub–protocols . In our protocol, each sub–protocol of the payment process is activated when it is needed (like the deposit

sub–protocol that is activated when the issuer approves the withdrawal) and unnecessary steps are avoided (as happens in SAMPP where the Payment Gateway must send the information to the issuer and the acquirer at the same time even though the withdrawal has not been approved).

The fourth difference can be found in the payment modes allowed by both protocols. In SAMPP, at the moment of the purchase, the client can use only his credit card whereas in ours, credit- or debit-card transactions are supported.

The last difference is the exchange of the secret shared between the client and the issuer (CISec). In the case of SAMPP, at the time of updating the CISec secret, a protocol AKE is used (among client/issuer) whereas in ours, the new secret must be sent inserted in the Withdrawal-script Response.

### 4.2 Performance

As SAMPP was reformulated to fit our needs, in this section we perform a comparison of both protocols in terms of performance, focusing on the number of cryptographic operations performed by each one (results of this comparison are shown in table 1). We can see that although operational models are different and our proposal is an evolution of SAMPP, the performance of our protocol is the same that of SAMPP.

**Table 1.** The number of cryptographic operations of SAMPP, and our protocol, respectively

| **Cryptographic Operations** | | **SAMPP** | **Ours** |
|---|---|---|---|
| 1. Symmetric-key encryptions/decryptions | C | 4 | 4 |
| | V | 5 | 5 |
| | P | 2 | 2 |
| 2. Hash functions | C | 2 | 2 |
| | V | - | - |
| | P | - | - |
| 3. Keyed-hash functions | C | 2 | 2 |
| | V | 2 | 2 |
| | P | 1 | 1 |
| 4. Key generations | C | 2 | 2 |
| | V | 1 | 1 |
| | P | 1 | 1 |

## 5 Conclusions

We have proposed a secure protocol which uses symmetric cryptographic techniques. It is applicable to mobile payment systems where direct communication between the client and the issuer does not exist. Thus, the client takes advantage of the infrastructure of the vendor and payment gateway to communicate with the issuer and purchase securely from her mobile device. Our proposal represents

an alternative to all mobile payment systems where the connection between the client and issuer is mandatory, including Visa's 3-D Secure scheme. Moreover, our scheme illustrates how a portable device equipped with a short range link (such Bluetooth, Infrared or Wi-Fi) and low computational power is enough to interact with a vendor machine in order to buy goods in a secure way

The symmetric cryptographic technique used in our protocol has lower computation requirements at both parties (since no public-key operation is required) and offers the capability of dealing with protocol failures and disputes among parties. Moreover, we have shown that our protocol's performance is about the same than that of SAMPP, although this protocol is used in different operational models. As a result, we state that our proposed protocol allows mobile users to have efficient and secure payment systems even if the communication with the issuer is not possible.

## References

1. Al-Meaither, M.: Secure electronic payments for Islamic finance. PhD thesis, University of London, (2004).
2. Chari, S., Kermani, P., Smith, S., and Tassiulas, L.: Security issues in m-commerce: A usagebased taxonomy. In E-Commerce Agents, volume 2033 of Lecture Notes in Computer Science, pages 264-282, Springer-Verlag, (2001).
3. Collberg, C., Carter, E., Debray, S., Huntwork, A., Kececioglu, J., Linn, C., and Stepp, M.: Dynamic path-based software watermarking. In ACMSIGPLAN 2004 Conference on Programming Language Design and Implementation 2004, pages 107-118, ACM, (2004).
4. Ham, W., Choi, H., Xie, Y., Lee, M., and Kim, K.: A secure one-way mobile payment system keeping low computation in mobile devices. In WISA2002, Lecture Notes in Computer Science, pages 287-301. Springer- Verlag, (2002).
5. Kungpisdan, S.: A secure account-based mobile payment system protocol. In ITCC04, International Conference on Information Technology: Coding and Computing, pages 35-39, IEEE Computer Society, (2004).
6. Lei, Y., Chen, D., and Jiang, Z.: Generating digital signatures on mobile devices. In 18th International Conference on Advanced Information Networking and Applications (AINA 2004), pp. 532-535, IEEE Computer Society, (2004).
7. Sosonkin, M., Naumovich, G., and Memon, N.: Obfuscation of design intent in object-oriented applications. In 2003 ACM workshop on Digital rights management (DRM03), pp. 142-153, ACM Press, (2003).
8. Visa International: 3-d secure mobile authentication scenarios version 1.0, (2002). [Online], Available: http : //partnernetwork.visa.com/pf/3dsec/ specifications.jsp.
9. Zheng, X. and Chen, D.: Study of mobile payments system. In IEEE International Conference on Electronic Commerce, pp. 24, IEEE Computer Society, (2003).

**Journal of Theoretical and Applied Electronic Commerce Research**
ISSN 0718–1876 Electronic Version
VOL 1 / ISSUE 2 / AUGUST 2006 / 1 – 11


This paper is Available online at
www.jtaer.com

# Anonymous Payment in a Kiosk Centric Model using Digital signature scheme with message recovery and Low Computational Power Devices

**Jesús Téllez Isaac**[1] **and José Sierra Camara**[2]**,**
**Antonio Izquierdo Manzanares**[2] **and Joaquín Torres Márquez**[2]

[1] Universidad de Carabobo, Computer Science Department (Facyt)
Av. Universidad, Sector Bárbula, Valencia, Venezuela.
jtellez@uc.edu.ve
[2] Universidad Carlos III de Madrid, Computer Science Department,
Avda. de la Universidad, 30, 28911, Leganés (Madrid), Spain.
{sierra,aizquier,jtmarque}@inf.uc3m.es

Received 26 May 2006; accepted 28 July 2006

**Abstract**

In this paper we present an anonymous protocol for a mobile payment system based on a Kiosk Centric Case Mobile Scenario where the customer cannot communicate with the issuer due to absence of Internet access with her mobile device and the costs of implementing other mechanism of communication between both of them are high. Our protocol protects the real identity of the clients during the purchase and employs a digital signature scheme with message recovery using self-certified public keys that reduces the public space and the communication cost in comparison with the certificate-based signature schemes. Moreover, our proposed protocol requires low computational power that makes it suitable for mobile devices. As a result, our proposal illustrates how a portable device equipped with a short range link (such Bluetooth, Infrared or Wi-Fi) and low computational power should be enough to interact with a vendor machine in order to buy goods or services in a secure way.

**Key words:** Anonymous Protocol, Mobile Payment System, Kiosk Centric Model, Digital Signature with message recovery, Self-certifi ed public keys

# 1 Introduction

M-commerce refers to any electronic transaction or information conducted using a mobile device and mobile networks. Its popularity has increased in the last years thanks to advances in the portable devices and the rapid development of the mobile communication technologies that have allowed people to use mobile telephones or Personal Digital Assistant (PDA) to access Internet (to read email, browse web pages or purchase goods) anywhere and anytime.

Advances in the portable devices make m-commerce more profi table and promising, nevertheless there is still a widespread skepticism about buying and paying for them on-line, due to the vulnerability of sensitive information when transmitted through communication channels. Therefore, it is necessary to develop mobile payment systems capable of providing safe and trustworthy communications between the customer and on-line mobile services providers. Moreover, these payment systems should overcome the common limitations existing in mobile devices currently available, which prevent that these devices execute, in an effi cient way, operations that require a lot of computing resources. The common limitations are: 1) poor computational capabilities, 2) limited storage and 3) short battery life.

Different mobile payment systems have emerged in the last years which allow the payment of services/goods from mobile devices, but the one developed by [16] (called 3-D Secure) has become a standard due to its benefi ts regarding security and flexibility in the authentication methods. This scheme allows the authentication of the payer (customer) when she makes an on-line payment using a debit or credit card.

Despite of the flexibility that 3-D Secure provides to the issuer to choose the authentication method (password, symmetric and asymmetric signature, and biometric techniques), the relationship between payer and issuer is quite strict (although required for Visa's 3D-Secure scheme) and does not allow the use of schemes in which the communication among these parties is not possible due to: 1) the impossibility of the client to connect to Internet from the mobile device and 2) the high costs and / or inconveniences of using the infrastructure necessary to implement other mechanisms of communication between the client and the issuer (such SMS, phone call, etc).

Most of the mobile payment systems proposed up until now assume the consumer has Internet connectivity through her mobile device, so the restrictions mentioned previously do not represent an important issue. However, it is quite common that the client meets situations in which it is not possible to connect to Internet so it becomes necessary to develop mobile payment systems where the user could use her mobile device as a shopping means, even thought she may not have Internet access.

Digital Signature can be represented as a secure base in electronic payment system because it provides authentication, data integrity and non-reputation cryptography services [10]. However, the traditional digital signature schemes are based on asymmetric techniques which make the signature computation very expensive and not suitable for mobile devices. Moreover, these schemes suffer from the well-know authentication problem[1] which requires the usage of certifi cate to avoid it. The public-key certifi cate must be verifi ed by a Certifi cate Authority (CA), and that verifi cation causes an additional information exchange during a transaction.

According to our operational model, the above schemes are not suitable because clients interact only with a vendor during a purchase and communication with other party (like CA to verify a certifi cate) is not possible. Therefore, usage of a non-traditional digital signature scheme is required in order to satisfy our requirements. Digital signature with message recovery using self-certifi ed public keys [3],[15] provides an authenticated encryption scheme that integrates the mechanisms of signature and encryption, which enable only the specifi ed receiver to verify and recover the original message. The authentication of the public key can implicitly be accomplished with the signature verifi cation.

*Contributions:* In this paper, we present an anonymous protocol (that supports both credit-card and debit-card transactions) for a mobile payment system based on a Kiosk Centric Model (proposed by [4]) which overcomes the limitations mentioned before and is suitable for mobiles devices with low computational power. Our protocol protects the real identity of the clients during the purchase and employs a digital signature scheme with message recovery using self-certifi ed public keys that reduces the public space and the communication cost in comparison with the certifi cate-based signature schemes [17]. As a result, our proposal represents an alternative to other mobile payment systems with restrictions regarding a mandatory connection between the client and the issuer, like Visa's 3D Secure.

[1] An imposter may impersonate any innocent user with a valid cryptographic but incorrect public key (because it does not belong to the innocent user).

*Outline of this paper:* We begin by presenting the motivation for this work (section 2), followed by the related work that include a description of some known results associated to our research. We then present a brief review of some preliminaries (section 4). More precisely, we give a comparison between symmetric and asymmetric cryptography for mobile payment transactions and an overview of digital signature and self-certifi ed public key. Following this, we present our approach which includes a complete list of notations used in our scheme, the operational model, the initial assumptions and the proposed protocol. In section 6, a security analysis of the proposed protocol is presented. We end this paper with the conclusions in section 7.

## 2 Motivation

In this research, we can distinguish the following objectives: a) eliminate the restriction of those mobile payment systems (including Visa's 3-D Secure) about the direct communication between client and issuer for authentication purposes and b) provide anonymity of data origin to prevent the merchant from associating the client with the messages sent from her. This anonymity implies the protection of relevant information by third parties but not unrestrained anonymity [1].

Our fi rst objective is greatly inspired by the Kiosk Centric Case mobile Scenario proposed by [4]. This scenario is very representative of mobile application frameworks where the client device interacts directly with the kiosk, which in turn connects to the infrastructure. Note that the client's device never communicates with the infrastructure in a direct way but has a feasible connection with the vendor (using a short range link such bluetooth, infrared or wi-fi ).

In order to solve the problem of buy and payment of goods/services in the aforementioned scenario, in section 5, we construct a protocol that allows clients to send from their mobile device a message to the issuer through the vendor (who will not be able to decrypt this message). The proposed protocol (divided in 2 sub–protocols) employs the authentication encryption scheme proposed by [17] that allows only specifi ed receivers to verify and recover the message, so any other receiver will not be able to access the information.

According to our second objective, we make sure not to reveal the real identity of the client to a merchant during the purchase process. In order to achieve this goal, a nickname instead of client's real identity is used when she communicates with the merchant. While a client registers to the issuer, several nicknames are assigned to the client and those nicknames are known only to the client and the issuer. Since the merchant does not know the mapping the nickname and the true identity of a client, the client's privacy is protected [8].

## 3 Related Work

The widespread of m-commerce in recent years has created new security and privacy challenges because of new technology, novel applications, and increased pervasiveness [4]. Several studies have been conducted to improve the security of mobile payment systems. Meanwhile, efforts have also been dedicated to unify concepts and scenarios into frameworks that will be useful to develop new electronic payment systems and to analyze security issues.

Research conducted by [4] is one of those studies that unifi es many proposed m–commerce usages into a single framework. This research intended to revise the possible range of mobility scenarios, identifying the security issues for each connectivity scenario. As a result, fi ve scenarios were identifi ed and analyzed: Disconnected Interaction, Server Centric Case, Client Centric Case, Full Connectivity and Kiosk Centric Case. The latest has been considered as the starting point in the design of our proposal.

The Full Connectivity scenario (where all the entities are directly connected to one another) has been widely used in most of the mobile payment systems proposed up until now [11],[7],[16] because it allows protocol's designers to simplify the protocols and obtain stronger security guarantees than similar applications in the others models.

Most of the protocols proposed in recent years for the Full Connectivity scenario are based on public-key infrastructure (PKI) [2],[7],[10] whereas the remaining employ symmetric-key operations which is more suitable for wireless networks [9]. Unfortunately, usage of those protocols within the Kiosk Centric Case mobile scenario is not possible, as it restricts the communication which allows only interaction between the client and the merchant. However, some protocols could be reformulated to overcome this restriction, achieving the same security and performance but in a different scenario.

For example, Téllez *et al*. [14] reformulate the mobile payment protocol proposed by [9] to satisfy the requirements of their proposal (based on Kiosk Centric Model).

Many signature schemes with message recovery have been proposed in recent years [3],[15],[17]. These schemes allow a signer's public key to be simultaneously authenticated in verifying the signature. As the public keys does not need to be included in a certifi cate to be authenticated by verifi ers (as happens in protocols based on public-key infrastructure), communication with a Certifi cate Authority during a transaction to verify the validity of a certifi cate is not necessary. Therefore, digital signature schemes with message recovery are suitable for mobile payment systems based on a kiosk centric model like the one being suggested in this work.

In order to provide limited but practical anonymity by using limited disclosure of information, some proposals have been suggested in the past [1],[8]. While the cryptography techniques and operational models used in those works are different from ours, we follow the approach of using nicknames usage instead of the real identity, implemented in [8] to prevent a merchant from knowing the customer's identity.

As the payment software (also called wallet software, and that from now on we will assume that is programmed using the Java language due to the multiplatform capabilities of this language) must be sent to the customer by the issuer through the vendor, it becomes necessary the usage of techniques to protect the software against reverse engineering and/or software tampering. To achieve this, we employ the three techniques for obfuscation of program design proposed by [13]: 1) The *class coalescing obfuscation* replaces several classes with a single class, 2) In the *class splitting obfuscation*, a single class is replaced with multiple classes, each responsible for a part of the functionality of the original class, and 3) The *type hiding obfuscation* uses the mechanism of interfaces in java to obscure the type of objects manipulated by the program.

The experimental results (applying the techniques mentioned before to a medium-size java program) show that the run-time overhead, in the worst of possible scenario (class splitting obfuscation), is less than 10% of the total running time of the program.

# 4 Background

In this section, preliminaries necessary for the remainder of this paper will be introduced.

## 4.1 Comparison of Symmetric and Asymmetric Cryptography for Mobile Payment Transactions

Symmetric and asymmetric cryptography have been widely used for secure communications among engaging parties. In Symmetric cryptography, a secret is shared between two parties (called sender and receiver) that want to communicate safely without revealing details of the message. This technique provides message confi dentiality, message integrity and party authentication.

On the other hand, asymmetric cryptography employs a pair of cryptographic keys (public/private key) to allow users to communicate securely without having previous access to a shared secret key. This technique provides all the security properties that the symmetric cryptography does (confi dentiality, message integrity and party authentication),and also provides non-repudiation, which symmetric cryptographic could not provide and is very important for fi nancial transactions that are relevant to fund transfer and good ordering. Normally, it can be achieved by using digital signatures but in symmetric-key based protocols, there is no possibility to prove the originator of an encrypted message because the secret key is shared between two parties [9].

Symmetric-key operations are more suitable for wireless networks than asymmetric ones due to the time required to be processed and their lower computational requirements. However, key management is complex since shared secret key must be agreed upon by both parties and any participant has to maintain $n$ number of secret keys, one for each party she is communicating with. Moreover, authenticity of origin or receipt cannot be proved because the secret key is shared.

### 4.2 Digital Signature and Self-certified public key

Public-key cryptography, introduced in 1976 by Diffi e-Hellman to solve the problem of key management, is a class of cryptography which allows users to communicate securely without having prior access to a shared secret key. In asymmetric cryptosystems, each user has two keys: the private key that is kept secret by the user and could be used to produce a signature for a message and the public key which is made public to all users in a public directory maintained by a system authority (SA) [17]. These systems suffer from well-known authentication problems, being the most important one that an imposter may impersonate any innocent user with a valid cryptographic but incorrect public key (because it does not belong to the innocent user). To deal with the problem, the usage of certifi ed for every public key is applied but this approach puts a high burden on users since they should connect themselves to the SA in order to verify the certifi cate before using the corresponding public key [3].

In 1984, an ID-based public key cryptosystem was proposed by [12], where the public key of a user is computed from her identity(e.g. an email address, and IP address or a complete name). Unfortunately, this approach relies too much on system authority because the user's private key is not chosen by the user, but the SA. To deal with both the problem of verifi cation of public keys without the usage of certifi cates and the drawbacks on the identity-based cryptography, the notion of self-certifi ed public keys cryptosystem was fi rst introduced by [5] where each user chooses her private key, and the user's public key is derived from the signature of the user's private key (signed by the system authority using the system's secret key) and the user's identity. The authentication of the public key can implicitly be accomplished with the signature verifi cation.

## 5 Our Approach

### 5.1 Parties and Notations

All the entities involved in our protocol are called parties and communicate through wireless and wired network.

The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer respectively. The following symbols are used to represent other messages and protocols:

- $ID_P$ : the identity of party $P$ that contains the contact information of $P$.
- $NID_C$ : Client's nickname, temporary identity.
- $K_P$ : party's $K$ public key.
- $K_S$ : party's $K$ private key.
- $E_{P-P'}(X)$ : message X signed and encrypted by the user $ID_P$ to a specifi ed receiver $ID_{P'}$.
- TID: Identity of transaction that includes time and date of the transaction.
- OI: Order information (OI = $\{$TID, OD, $h$(OD, Price)$\}$) where OD and Price are order descriptions and its amount.
- TC: The type of card used in the purchase process (TC=Credit, Debit).
- Stt: The status of transaction (Stt = $\{$Accepted, Rejected$\}$).
- TIDReq : The request fot TID.
- MIDReq : The resquest for $ID_M$.
- XMReq : The request for $x_M$.
- h(M) : the one-way hash function of the message M.

### 5.2 Operational Model

Generally, operational models for m-commerce found in literature involve transaction between two or more entities. Our operational model is composed of fi ve entities:

1. *Client*: a user who wants to buy goods or services from the merchant. Particulary, in our proposal, the user has a mobile device with the following features: a) low computational power (e.g mobile phone, PDA, etc.), b) equipped with a built-in display, an input method and short range link (such Infrared, Wi-Fi or Bluetooth), c) capability to execute a java program, and d) not able to access Internet.

2. *Merchant*: a computational entity that has products or services to offer/sell to the client and with which the user participates in a transaction. This entity could be a normal web server or an intelligent vending machine which the user can connect to using a short range link. Moreover, this entity connects with the Payment Gateway through a secure wired channel allowing the client to communicate with the issuer using this connection. A formal defi nition of the merchant may be found in [4].

3. *Acquirer*: is the merchant's fi nancial institution. It verifi es the validity of the deposited payment instrument and manages the merchant's account including fund transfer.

4. *Issuer*: is the client's fi nancial institution. It provides electronic payment instruments to the client to use in a payment and manage the client's account including fund transfer.

5. *Payment Gateway*: an additional entity that acts as a medium between acquirer/issuer at banking private network side and client/merchant at the Internet side for clearing purpose [9].

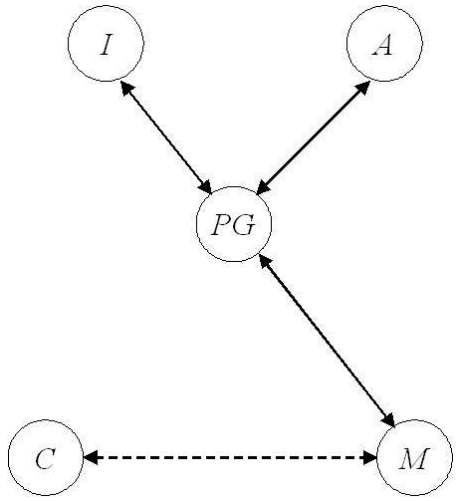


Figure 1: Operational Model.

In fi gure 1, we specify the links among the fi ve entities of our scheme. Note that there is no direct connection involving the client and the issuer. Moreover, the connection between the customer and the vendor (denoted as the dotted arrow) is set up through a wireless channel.

On the other hand, interaction among the vendor and the payment gateway (depicted as solid arrow in the scheme) should be reliable and secure against passive and active attacks. Therefore, the connection is supposed to be established through a secure wired channel by using the well-know security protocol like SSL/TLS [6]. Note that the issuer, acquirer and payment gateway operates under the banking private network, so the security of the messages exchanged among them is out of the scope of this paper.

### 5.3 Initial Assumptions

The initial assumptions for our proposed protocol can be stated as follows:

1. Client registers herself to an issuer before making payments. The registration can be done either personally at the issuer's premises or via the issuer's website.
2. The Client shares her credit- and/or debit-card information (CDCI) with her issuer (who will not reveal it to any merchant). $q$, $p'$ and $q'$ secret and publishes $N$, g and a collision-resistant hash function $h(\cdot)$ to all users
3. The trusted system authority (SA) is responsible for generation system parameters in the system initialization phase (by the procedure described in [17][15]).
4. Every party of the system $P_i$ (whose identity is $ID_{P_i}$) choose a number $K_{S_i}$ as her secret key and computes $x_i$ = $g^{K_{S_i}}$ mod $N$. Then, $P_i$ sends ($x_i$, $ID_{P_i}$) to SA. After receiving ($x_i$, $ID_{P_i}$), the trusted system authority computes and publishes the public key of $P_i$ as $K_{P_i}$=$(x_i - ID_{P_i})^{h^{-1}(ID_{P_i})}$ mod $N$ [17]. As the client uses a nickname instead the real identity to protect her privacy, one $K_{P_i}$ must be generated and published for every nickname assigned to the client.
5. When a client registers to a issuer, several nicknames are assigned to the client and those nicknames are known only by the client and the issuer [8]. Furthermore, with the assistance of issuer, the client sends her nicknames and $x_C$ to SA and receives all system parameters from the system authority.
6. The client holds $C_S$, $ID_I$, $x_I$ and system parameters in her mobile device.
7. The Price and description of the goods and services have been decided by client and merchant.

### 5.4 Signature with Message Recovery Techniques

In order to a sender $P_i$ (with identity $ID_{P_i}$) sign and encrypt a message W to a specifi ed receiver $P_j$ (with identity $ID_{P_j}$), we follow the generation procedure of signature proposed by [17]. First, $P_i$ chooses a random number $y$ and computes $r_1$, $r_2$ and $s$. Afterwards, $P_i$ sends the triple ($r_1$,$r_2$,$s$) as the signature of message W (from now on, $E_{P_i-P_j}$) to the verifi er $P_j$. After receiving ($r_1$,$r_2$,$s$), the verifi er $P_j$ recovers message W and verifi es that the signature is valid using the same procedure described in [17].

### 5.5 Detailed Protocols

Our protocol consists of two sub–protocols. In the *Registration Protocol*, a client requests the values $ID_M$ and $x_M$ from the merchant. Then, if the client does not have the wallet software, $M$ sends a request for the software to the issuer, and it will be delivered to the client through the merchant. After the client receives $x_M$ and the wallet software is available in client's mobile device, the client can start the *Payment Protocol*. The main functions of both protocols are shown as follow:

**Merchant Registration Protocol**

**C** $\rightarrow$ **M:** $\{NID_C, n, MIDReq, XMReq\}_w$
**M** $\rightarrow$ **C:** $\{ID_M, x_M, h(n, x_M)\}_w$
**C** $\rightarrow$ **M:** $h(ID_M, n, x_M)$

Due the absence of connection between the client and the trusted system authority during a payment, the client can not access the public value $x_M$ used in the signature process. As a consequence, the merchant needs to send this value to client. First, $C$ sends to $M$ her nickname $NID_C$, a nonce $n$ for challenge-response, MIDReq and XMReq (the request for $x_M$), encrypted with a session key $w$ generated by running AKE protocol [2] with $C$. Then, $M$ confi rms $C$'s

registration by sending the value $X_M$, merchant's identity ($ID_M$) and $h(n, x_M)$, encrypted with the session key $w$. Finally, $C$ sends $h(ID_M, n, x_M)$ to $M$ as a confi rmation to have received $x_M$.

Afterwards, the merchant will detect if the wallet software is available or not in the mobile device. If not, $M$ sends a software request to $PG$, which will forward the request to $I$. The issuer intends to protect the software against various types of attacks carried away at any moment, preparing the software following these steps:

- Choose one of the obfuscation methods proposed by [13] and apply it to the java code.
- Then, the software is signed using the Authenticated encryption scheme with message linkages proposed by [3],[17].

Once the software has been prepared, the issuer will forward it to the $PG$, which will send it to $C$ (through the merchant) who will install the software after receiving it.

**Payment Protocol**

**1) C → M:** $NID_C, TIDReq$
**M → C:** $E_{M-C}(TID, ID_M)$

**2) C → M:** $E_{C-M}(OI, Price, NID_C, ID_I, VSRequest, h(OI, NID_C, ID_I))$
$VSRequest = E_{C-I}(Price, h(OI), TC, ID_M)$

**3) M → PG:** $E_{M-PG}(VCRequest, ID_M)$
$VCRequest = (VSRequest, h(OI), TID, Price, NID_C, ID_I)$

**4)** Under banking private network,
**4.1) PG → I:** $VSRequest, h(OI), TID, Price, NID_C, ID_M$
**4.2) PG → A:** $Price, ID_M$
**4.3) I,A → PG:** $VSResponse, Stt, h(Stt, h(OI))$
$VSResponse = E_{I-C}(Stt, h(OI))$

**5) PG → M:** $VCResponse$
$VCResponse = E_{PG-M}(Stt, VSResponse, h(Stt, h(OI)))$

**6) M → C:** $PResponse$
$PResponse = E_{M-C}(VSResponse)$

**Step 1:** The Client **C** and merchant **M** exchange the information necessary to start the protocol.

**Step 2:** **C** creates a *Payment Request* (referred to the General Payment Model described in [11],[9]) including C's nickname, M's and I's identity, *Price*, *OI* and *Value-Subtraction Request* (called *VSRequest*, which is encrypted to be recovered only by an issuer **I**). *OI* is used to inform **M** about the goods and prices requested and *Payment Request* is encrypted by **C** to the specifi c receiver **M**. Once the *Payment Request* has been prepared, **C** sends it to **M**. Note that some important fi elds, such as *OI*, *Price*, $NID_C$, $ID_I$, are hashed in order to check if they are modifi ed or replaced with others while in transit.

**Step 3:** **M** decrypts the message received from **C** to retrieve *OI*. The Merchant **M** prepares the *Value-Claim Request* (called *VCRequest*) and then sends it with the merchant's identity to *PG*, encrypted to be recovered only by her in order to ensure that only the payment gateway is the intended recipient of the message. The *Value-Claim Request* contains the forwarded *Value-Substraction Request*, C's nickname, I's identity, order's amount, identity of transaction and the hash of the order information.

**Step 4:** **PG** decrypts the message received from **M** to retrieve *VSRequest* and the others fi elds included in *VCRequest*. Then, **PG** forwards *VSRequest* and other important information, namely: *h(OI)*, *TID*, *Price*, $NID_C$, $ID_I$ to **I** who will

process it to approve or reject the transaction. Also, **PG** sends $ID_M$ and the requested price (*Price*) to claim to acquirer **A** that she is the party whom the requested amount *Price* will be transferred to. After checking the validity of the client's account, the total amount of *OI* is transferred to the merchant's account, the issuer **I** prepares *Value-Substraction Response* (called *VSResponse*) and sends it to **PG** with the approval result (*Stt*). Note that *VSResponse* is encrypted to be recovered only by an issuer **C**.

**Step 5: PG** sends *Value-Claim Response* (called *VCResponse*) encrypted to be recovered only by **M**. *VCResponse* includes *VSResponse* which will be forwarded to **C**. As **M** has her own *OI*, she can compare this fi eld with the received *h(OI)* to check whether or not the message is the response of her request. If they are not matched, **M** sends a message to the PG pointing the problem. The PG may now start a recovery procedure or resend the message.

**Step 6: M** encrypts *Value-Subtraction Response* to be recovered only by **C**. Then, **M** sends it to **C** as *Payment Response* (called *PResponse*). Once **C** receives the message, decrypts it to retrieve the result of her request.

Once the purchase has been completed, the client does not have to run *Merchant Registration Protocol* again unless she wants to perform transaction with a new merchant. Note that, after clients fi nish all purchases with a merchant, she will remove the values $ID_M$ and $x_M$ from her mobile device due to its limited amount of storage.

# 6 Analysis and Discussions

## 6.1 Security issues

After each run of the proposed protocol, the achievement of the following goals will ensure the security of the payment in our mobile payment system.

- **Goal 1**: Authentication between the client and the issuer

  The operational model used in this proposal has a communication restriction: client can not communicate directly with the issuer. Therefore, in order to allow that issuer authenticates a client, *C* has to send a message to *I* (through the merchant) with the following features: 1) resistant to attacks while in transit, 2) recoverable only by the issuer, and 3) able to assure that it has been created and sent by *C*.

  Since the authenticated encryption scheme used in our protocol integrates the mechanisms of signature and encryption, the message *VSRequest* sent by *C* to *I*, satisfi es all the requirements mentioned above and can be used by the issuer to authenticate the client.

- **Goal 2**: Anonymity

  In order to prevent a merchant from knowing the identity of her clients, usage of client's nickname ($NID_C$) instead of her real identity is required during a communication from *C* to *M*. Since the C's nickname is known only by the client and the issuer, merchant cannot map the nickname and C's true identity. Thus, client's privacy is protected and untraceable.

- **Goal 3**: Confi dentiality

  The authentication encryption scheme used in our protocol ensures the encryption of important data of per transaction while in transit. Moreover, since this scheme allows that only the specifi ed receiver can verify and recover the message, any other receiver is unable to do it. For example, the *VSRequest* is created by *C* and encrypted to be recovered only by *I*. Any other party couldn't decrypt the message because requires the issuer's private key which is known only by *I*.

- **Goal 4**: Integrity

  It is important to protect data from being modifi ed or/and replaced while in transit. To achieve that, usage of message digest algorithms and/or digital signature are required. In our protocol, the integrity is mainly ensured by the digital signature with message recover technique. Also, we use a hash function of some information (e.g, *Order Information*, *Client's nickname*, etc), padded into some message in order to ensure the integrity where the digital signature is not used (e.g under banking private network).

using hash value of some important information (e.g, *Order Information*, *Client's nickname*, etc), padded into some messages.

- **Goal 5**: Non-repudiation of Origin (NRO)

  The Non-repudiation of Origin is ensured since the signature of a message is generated by the signer U using her private key $U_S$ (known only by U). Therefore, the signer should not be able to repudiate his signature creation later.

- **Goal 6**: Trust Relationships

  Generally, in any transaction, a party should not trust others unless they can provide a proof of trustworthiness [9]. However, as in our protocol the issuer issues a credit- and/or debit-card to the client and she will not reveal it to any part, we state the trust relationship between the client and the issuer.

## 7 Conclusions and Further work

This paper proposes a protocol for secure payments in a mobile payment system where direct communication between client and issuer does not exist. Our protocol uses a digital signature scheme with message recovery using self-certifi ed public keys. Furthermore, it allows clients to make purchases without disclosing private information and takes advantage of the infrastructure of the *merchant* and *payment gateway* to communicate with the issuer. With this protocol the client is able to purchase securely from her mobile device in a similar way to that of traditional mobile payment systems.

Our proposal represents an alternative to all mobile payment systems based on the Full Connectivity scenario (including Visa's 3-D Secure scheme) where communication between the client and issuer is mandatory. Moreover, we state that a portable device with a short range link (such Bluetooth, Infrared or Wi-Fi) and low computational capabilities is enough for interacting with a merchant in order to buy goods or services in a secure way.

As a result, we assert that our proposed protocol allows mobile users to have effi cient and secure payment systems even if the communication with the issuer is not possible.

In the future, as the proposed protocol includes only non-repudiation of origin, it will be valuable to incorporate more non-repudiation services (such as non-repudiation of receipt, non-repudiation of submission and non-repudiation of delivery) in order to prevent entities from denying that they have sent or received certain messages.

## Acknowledgements

This work was supported in part by ASPECTS-M Project (Reference Models for Secure Architectures in Mobile Electronic Payments), CICYT-2004, however it represents the view of the authors.

## References

[1] N. Asokan, Anonymity in a Mobile Computing Environment, in Proceedings of IEEE Workshop on Mobile Computing Systems and Applications. IEEE, 1994, pp. 200–204.

[2] M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. V. Herreweghen, and M. Waidner, Design, implementation, and deployment of the iKP secure electronic payment system, IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 611–627, 2000.

[3] Y. Chang, C. Chang, and H. Huang, Digital signature with message recovery using self-certifi ed public keys without trustworthy system authority, Journal of Applied Mathematics and Computation, vol. 161, no. 1, pp. 211–227, 2005.

[4] S. Chari, P. Kermani, S. Smith, and L. Tassiulas, Security Issues in M-Commerce: A Usage-Based Taxonomy, in Proceedings of E-Commerce Agents, 2001, pp. 264–282.

[5] M. Girault,Self-Certifi ed Public Keys, in Proceedings of EUROCRYPT, 1991, pp. 490–497.

[6] W. Ham, H. Choi, Y. Xie, M. Lee, and K. Kim, Secure One-way Mobile Payment System Keeping Low Computation in Mobile Devices, in Proceedings of WISA'02, 2002, pp. 287–301.

[7] J. Hall, S. Kilbank, M. Barbeau, and E. Kranakis, WPP: A Secure Payment Protocol for Supporting Credit- and Debit-Card Transactions over Wireless Networks, in Proceedings of International Conference on Telecommunications (ICT 2001). IEEE, 2001.

[8] Z. Hu, Y. Liu, X. Hu, and J. Li, Anonymous Micropayments Authentication(AMA) in Mobile Data Network, in Proceedings of INFOCOM, 2004.

[9] S. Kungpisdan, B. Srinivasan, and P. D. Le, A Secure Account-Based Mobile Payment Protocol, in Proceedings of ITCC (1), 2004, pp. 35–39.

[10] Y. Lei, D. Chen, and Z. Jiang, Generating Digital Signatures on Mobile Devices, in Proceedings of AINA (2), 2004, pp. 532–535.

[11] J. L. A. Peiro, N. Asokan, M. Steiner, and M. Waidner, Designing a generic payment service, IBM Syst. J., vol. 34, no. 1, pp.72–80, 1997.

[12] A. Shamir, Identity-Based Cryptosystems and Signature Schemes, in Proceedings of CRYPTO, 1984, pp. 47–53.

[13] M. Sosonkin, G. Naumovich, and N. D. Memon, Obfuscation of design intent in object-oriented applications, in Proceedings of Digital Rights Management Workshop, 2003, pp. 142–153.

[14] J. Téllez, J. Sierra, A. Izquierdo, and M. Carbonell, Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices, in Proceedings of ICCSA (5), 2006, pp. 798–807.

[15] Y. Tseng, J. Jan, and H. Chien, Digital signature with message recovery using self-certifi ed public keys and its variants, Journal of Applied Mathematics and Computation, vol. 136, no. 2–3, 2003.

[16] Visa International, (2002). 3-d secure mobile authentication scenarios version 1.0. [Online], Available: http://partnernetwork.visa.com/pf/3dsec/specifi cations.jsp

[17] J. Zhang, W. Zou, D. Chen, and Y. Wang, On the Security of a Digital Signature with Message Recovery using Self-certifi ed Public Key, Soft Computing in Multimedia Processing Special Issue of the Informatica Journal, vol. 29, no. 3, pp. 343–346, 2005.

# An Anonymous Account-Based Mobile Payment Protocol for a Restricted Connectivity Scenario

Jesús Téllez Isaac
Universidad de Carabobo,
Computer Science Department (Facyt)
Av. Universidad, Sector Bárbula,
Valencia, Venezuela.
jtellez@uc.edu.ve

José Sierra Camara
Universidad Carlos III de Madrid,
Computer Science Department
Avda. de la Universidad, 30, 28911,
Leganés (Madrid), Spain.
sierra@inf.uc3m.es

## Abstract

*Recent advances in m-commerce have raised the usage of scenarios with communication restrictions. These scenarios create new security challenges which must be considered by protocol's designers in order to achieve the same security capabilities as in those protocols designed for mobile payment systems based on a 'Full connectivity' scenario (where all the entities can exchange messages with each other without intermediaries). In this paper, we propose an anonymous payment protocol for a Client Centric Mobile Scenario where the merchant has no direct communication with the acquirer due to absence of Internet access in her infrastructure, and the unaffordability of other communication technologies due to the inconveniences and costs associated. The proposed protocol uses symmetric-key operations which require low computational power and can be processed much faster than asymetric ones. As a result, our proposal illustrates how a merchant can sell goods in a secure way even if she can not directly communicate with the acquirer.*

## 1. Introduction

During the last years, the importance of mobile commerce has increased while becoming an important part of our daily lifes. Its popularity has increased thanks to advances in portable devices and the rapid development of mobile communication technologies, which have allowed people to use mobile phones and Personal Digital Assistants (PDAs) to access Internet anywhere and anytime.

In spite of advances in the portable devices which make m-commerce more profitable and promising, there is still a widespread skepticism about buying and paying for them on-line, due to the vulnerability of sensitive information when transmitted through communication channels. Therefore, it is necessary to develop mobile payment systems capable of providing safe and trustworthy communications between the customer and on-line mobile services providers. Moreover, these payment systems should overcome the common limitations existing in mobile devices currently available, which prevent that these devices execute, in an efficient way, operations that require a lot of computing resources.

Several mobile payment systems have emerged in the last years which allow payments for services and goods from mobile devices using different kinds of payments: credit- and/or debit-card payments, micropayments and digital coins. The relationship between payee and acquirer is quite strict in most of these mobile payment systems and does not allow the use of schemes in which the communication among these parties is not possible due to: 1) the impossibility of the merchant to connect to Internet and 2) the high costs and / or inconveniences of using the infrastructure necessary to implement other mechanisms of communication between the merchant and the acquirer (such SMS, phone call, etc).

The above restrictions do not represent an important issue for the majority of mobile payment systems proposed up until now because they assume that the merchant has Internet connectivity through his/her infrastructure. Nevertheless, in the real world there are some situations that the merchant meets in which it is not possible to connect to the Internet, so it becomes necessary to develop mobile payment systems where the payee could sell goods/services even thought he/she may not have Internet access.

In order to provide authentication in electronic payment systems (including mobile commerce), the following methods are considered: username/password, symmetric and asymmetric cryptography, and biometry. As username/password does not offer enough security for

1529-4188/07 $25.00 © 2007 IEEE
DOI 10.1109/DEXA.2007.132

m-commerce and biometry is not feasible at present, symmetric and asymmetric signature are chosen for authentication[4]. However, traditional asymmetric signature schemes make the signature computation very expensive and not suitable for mobile devices.

According to our operational model (where merchant cannot communicates with the acquire in a direct way, formally called Client Centric Model [2]), the above schemes are not suitable because merchant interacts only with a client during a purchase and communication with other party (like CA to verify a certificate) is not possible. Therefore, usage of symmetric signature scheme is required in order to satisfies our requirements. Symmetric cryptography (which employs a shared key between two parties) provides, like asymmetric cryptography, message confidentiality, message integrity and party authentication, and represents an alternative in the construction of secure protocols for mobile payment systems, because symmetric-key operations do not require of a high computational power nor additional communications steps (as happens in protocols based on public-key infrastructure where the public-key certificates have to be verified by a Certificate Authority).

In order to solve the problem of buy and payment of goods/services in our operational model, in section 3, we construct a protocol that allows to a merchant to send a message to acquirer through the client (who will not be able to decrypt this message). The proposed protocol (divided in 2 sub–protocols) employs symmetric-key operations in all engaging parties to reduce both, the setup cost for payment infrastructure and the transaction cost. Moreover, it supports credit-card and debit-card transactions and protects the real identity of the clients during the purchase. As a result, our proposal represents an alternative to other mobile payment systems with restrictions regarding a mandatory connection between the merchant and the acquirer.

The outline of the paper is as follows. In section 2, we present the related work that include a description of some known results associated to our research. Following this, we present our approach which includes a complete list of notations used in our scheme, the operational model, the initial assumptions and the proposed protocol. In section 4, a security analysis of the proposed protocol is presented. We concludes the paper in section 5.

## 2 Related Work

Several studies have been conducted to unify concepts and scenarios into frameworks that will be useful to develop new electronic payment systems and to analyze security issues. Recently, [2] conducted a research that unifies many proposed m–commerce usages into a single framework. This research intended to revise the possible range of mobility scenarios, identifying the security issues for each connectivity scenario. As a result, five scenarios were identified and analyzed: Disconnected Interaction, Server Centric Case, Client Centric Case, Full Connectivity and Kiosk Centric Case. The latest has been considered as the starting point in the design of our proposal.

The Full connectivity scenario (where all the entities are directly connected to one another) has been widely used in most of the mobile payment systems proposed up until now [9, 3, 11] because it allows protocol's designers to simplify the protocols and obtain stronger security guarantees than similar applications in the others models.

Most of the protocols proposed in recent years for the Full Connectivity scenario are based on public-key infrastructure (PKI) [1, 3, 7, 10] whereas the remaining employ symmetric-key operations which is more suitable for wireless networks [6]. Unfortunately, usage of those protocols within the Client Centric Case mobile scenario is not possible, as it restricts the communication and not allows direct interaction between the merchant and the acquirer. However, some protocols could be reformulated to overcome this restriction (achieving the same security and performance levels, but in a different scenario), while being suitable for mobile payment systems with Restricted Connectivity (like the one being suggested in this work). For example, Téllez *et al.* [5] reformulate the mobile payment protocol proposed by [6] to satisfy the requirements of their proposal (based on Kiosk Centric Model).

## 3 Our Approach

### 3.1 Operational Model

The m-commerce model employed in our approach is composed of 4 involved entities: $client$, $merchant$, $issuer$ (client's financial institution), and $acquirer$ (merchant's financial institution). An additional entity called $payment\ gateway$ acts acts as a medium between acquirer/issuer at banking private network side and client/vendor at the Internet side for clearing purpose [6].

The five entities in our scheme and their links are shown in Fig. 1. The connection between the client and the merchant (denoted as the dotted arrow) is set up through a wireless channel. On the other hand, there is no direct connection involving the merchant and the acquirer and the connection between the client and the merchant (depicted with a solid arrow) is supposed to be established through a secure wireless channel by using the well-know security protocol likes, as such SSL/TLS. Note that the $issuer$, $acquirer$ and $payment\ gateway$ operates under the banking private network, so the security of the messages exchanged among them is out of the scope of this paper.

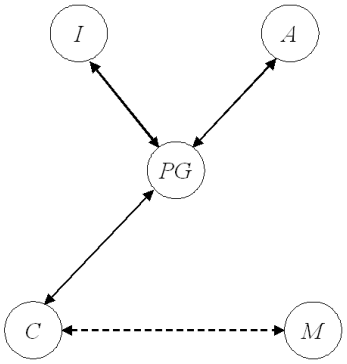


**Figure 1. Operational Model.**

### 3.2 Initial Assumptions

The initial assumptions for our proposed protocol can be stated as follows:

1. Client registers herself/himself to an issuer before making payments. The registration can be done either personally at the issuer's premises or via the issuer's website.
2. The Client shares her credit- and/or debit-card information (CDCI) with her issuer (who will not reveal it to any merchant). CDCI contains the long-term secret CISec known only by the client and her issuer.
3. The client registers herself/himself to the Payment Gateway. Then, Payment Gateway shares the secret $\text{Sec}_{C-PG}$ with the client. Besides, the issuer shares the secret $\text{Sec}_{C-I}$ with the client. All secrets can be distributed by using an authenticated-key exchange (AKE) for wireless networks [8, 1].
4. When a client registers to a issuer, several nicknames are assigned to the client and those nicknames are known only by the client and the issuer [4].

### 3.3 Parties and Notations

All the entities involved in our protocol are called parties and communicate through wireless and wired network.

The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer respectively. The following symbols are used to represent other messages and protocols:

### 3.4 Detailed Protocols

Our protocol consists of two sub–protocols. In $MerchantRegistration\ Protocol$, client shares the values $NID_C$ and the secret $\text{Sec}_{C-M}$ with the merchant.

- $\text{ID}_P$: The identity of party $P$ that contains the contact information of $P$.
- $\text{NID}_C$: Client's nickname, temporary identity.
- TID: Identity of transaction that includes time and date of the transaction.
- OI: Order information (OI = {TID, OD, $h$(OD, Price)}) where OD and Price are order descriptions and its amount.
- TC: The type of card used in the purchase process (TC=Credit, Debit).
- Stt: The status of transaction (Stt = {Accepted, Rejected}).
- TIDReq: The request for TID.
- MIDReq: The request for $\text{ID}_M$.
- $\text{Sec}_{A-B}$: The master secret shared between parties A and B.
- $\text{KS}_{A-B_j}$: The session key shared between parties A and B, generated applying a hash function with *j*-bit cyclic shifting (either left shift or right shift) of $\text{Sec}_{A-B}$. More details about this technique can be found in [6]
- $\{\text{M}\}_x$: The message M symmetrically encrypted with the shared key X.
- $h$(M): The one-way hash function of the message M.
- MAC(X,K): Message Authentication Code (MAC) of the message X with the key K.

Then, if the client does not have the wallet software (which contains both key generation and payment software, and that from now on we will assume that is programmed using the Java language due to the multiplatform capabilities of this language), she connects to issuer's web site to download it or sends a request to the issuer in order to receive it by mail. After the merchant receives $NID_C$, $\text{Sec}_{C-M}$ and the wallet software is available in client's mobile device, the client can start the $Payment\ Protocol$. The main functions of both protocols are shown as follow:

**Merchant Registration Protocol**

**C** $\rightarrow$ **M:** $\{NID_C, Sec_{C-M}, n, MIDReq\}_w$
**M** $\rightarrow$ **C:** $\{ID_M, h(n, NID_C, Sec_{C-M}, ID_M)\}_w$
**C** $\rightarrow$ **M:** $h(ID_M, n)$

Prior to the information exchange, the client **C** generates $Sec_{C-M}$ which is to be shared with the merchant. Then, $C$ sends to $M$ her nickname $NID_C$, $\text{Sec}_{C-M}$, a nonce $n$ for challenge-response and MIDReq, encrypted with a session key $w$ generated by running AKE protocol with $C$. Then, $M$ confirms $C$'s registration by sending the merchant's identity ($ID_M$) and $h(n, NID_C, Sec_{C-M}, ID_M)$,

encrypted with the session key $w$. Finally, $C$ sends $h(ID_M, n)$ to $M$ as a confirmation to have received $ID_M$.

Afterwards, if the wallet software is not available in the mobile device, the client connects to issuer's web site to download it or sends a request to the issuer to receive it by mail. Once the client has received the software, she will install it in the mobile device.

**Payment Protocol**

**1) C → M:** $NID_C, i, TIDReq$
**M → C:** $\{TID, ID_M\}_{KS_{C-M_i}}$

**2) C → M:**
$\{OI, Price, NID_C, ID_I, z, VSRequest\}_{KS_{C-M_i}}$,
$MAC[(Price, h(OI), NID_C, ID_C), KS_{C-M_{i+1}}]$
$VSRequest = MAC[(Price, h(OI), TC, ID_M), KS_{C-I_z}]$

**3) M → C:** $\{VCRequest, h(KS_{M-PG_{k+1}})\}_{KS_{C-M_i}}$,
$MAC[(VCRequest), KS_{M-PG_{k+1}}]$
$VCRequest = \{VSRequest, h(OI), k, z, TID, ID_M, Price, NID_C, ID_I\}_{KS_{M-P_k}}$

**4) C → PG:** $\{VCRequest, h(KS_{M-PG_{k+1}})$
$MAC[(VCRequest), KS_{M-PG_{k+1}}]\}_{KS_{C-PG_j}}$, $j$

**5)** Under banking private network,
**5.1) PG → I:** $NID_C, ID_M, VSRequest, TID$
$h(OI), z, Price, h(KS_{M-PG_{k+1}})$
**5.2) PG → A:** $Price, ID_M$
**5.3) I,A → PG:** $VSResponse, Stt, h(Stt, h(OI))$
$VSResponse = \{Stt, h(OI)\}_{KS_{C-I_z}}$

**6) PG → C:** $PResponse$
$VCResponse = \{Stt, h(Stt, h(OI))\}_{KS_{M-PG_{k+1}}}$
$PResponse = \{VSResponse, VCResponse\}_{KS_{C-PG_{j+1}}}$

**7) C → M:** $\{VCResponse\}_{KS_{C-M_{i+1}}}$

**Step 1:** The client **C** sends her nickname ($NID_C$), the request for the transaction identity (*TIDReq*) and the index *i* (that will be used to generate the session key between the client and the merchant) to **M**. Once the request is received by **M**, he/she sends his/her identity ($ID_M$) and *TID* to **C**, encrypted only to be recovered by the client.

**Step 2:** **C** creates a *Payment Request* (referred to the General Payment Model described in [6]) including I's identity, *Price*, the index *z* and *OI* (used to inform **M** about the goods and prices requested). It also contains the *Value-Substraction Request* (called *VSRequest*, which is to be forwarded to the issuer **I** and includes *Price*, *TC*, $ID_M$ and $h(OI)$. The client **C** encrypts the *Payment Request* to be recovered only by **M** (using the session key $KS_{C-M_i}$) and sends it to the merchant. Note that a MAC function is used in order to check if the message is modified or replaced with other while in transit.

**Step 3:** **M** decrypts the message received from **C** to retrieve *OI*. **M** prepares the *Value-Claim Request* (called *VCRequest*), encrypted with $KS_{M-PG_k}$ in order to ensure that only the Payment Gateway **PG** is the intended recipient of the message. Once *VCRequest* has been prepared, **M** prepares a new message (which includes $h(KS_{M-PG_{k+1}})$, *VCRequest* and its MAC value) and sends it to **C**, encrypted with $KS_{C-M_i}$. The *Value-Claim Request* contains C's nickname, I's and M's identity, order's amount, identity of transaction, the indexes *k* and *z* (used to identify the current session keys $KS_{M-P_k}$ and $KS_{C-I_z}$, respectively), the hash of the order information and the forwarded *VSRequest*.

**Step 4:** Once **C** receives the message sent by **M** in **step 3** (that includes $h(KS_{M-PG_{k+1}})$, *VCRequest* and its MAC value) , encrypts it with $KS_{C-P_j}$ and then sends it to **PG** with the index *j*.

**Step 5:** **PG** decrypts the message received from **C** to retrieve *VSRequest* and the others fields included in *VCRequest*. Then, **PG** forwards *VSRequest*, the indez *z* and other important information, namely: *h(OI)*, *TID*, *Price*, $NID_C$, $ID_M$ and $h(KS_{M-PG_{k+1}})$ to **I** who will process it to approve or reject the transaction. Also, **PG** sends $ID_M$ and the requested price (*Price*) to claim to acquirer **A** that she is the party whom the requested amount *Price* will be transferred to. After checking the validity of the client's account, the total amount of *OI* is transferred to the merchant's account, the issuer **I** prepares *Value-Substraction Response* (called *VSResponse* and encrypted with $KS_{C-I_z}$ ) and sends it to **PG** with the approval result (*Stt*).

**Step 6:** **PG** sends *Payment Response* (called *PResponse*) encrypted with $KS_{C-PG_{j+1}}$ to **C**. *PResponse* include *VSResponse* and *VCResponse* (which will be forwarded to **M**).

**Step 7:** After receiving *PResponse*, **C** decrypts it to retrieve *VSResponse* and *VCReponse*. As **C** has her own *OI*, she can compare this field with the received *h(OI)* to check whether or not the message is the response of her request. If they are not matched, **C** sends a message to the **PG** pointing the problem (then, the payment gateway can start a recovery procedure or resend the message). Otherwise, **C** encrypts *Value-Claim Response*, with $KS_{C-M_{i+1}}$ and sends it to the merchant who in turn proceeds to deliver the goods to the client.

After a transaction is completed, each entity of the payment system put in her revocations list, $\text{Sec}_{C-M}$ and $\text{Sec}_{C-I}$ to prevent their replay from the client and the merchant. In the following purchases, the *Merchant Registration Protocol* will not occur until the customer

is notified to update the secret $\text{Sec}_{C-M}$. Thus, when become necessary to renew the secret, the customer runs the *Merchant Registration Protocol* to get a new $\text{Sec}_{C-M}$. While the secret is not updated, the client can use other values in the set of $\text{Sec}_{C-M_i}$ to perform transactions. To update the $\text{Sec}_{C-I}$, the Issuer sends the new secret to the client by using an AKE protocol. Finally, to update the $\text{Sec}_{PG-I}$, the Payment Gateway has to add a message with the new secret to the *VCResponse* which will be modified as following:

$$\{Stt, Newsec, h(Stt, Newsec, h(OI))\}_{KS_{M-PG_{k+1}}}$$

## 4 Analysis and Discussions

**Transaction Security:** Our protocol satisfies the following transaction securities: A) *Entity authentication* is ensured by symmetric encryption and the secret $\text{Sec}_{C-I}$ (which guarantees that the message is originated by the client), B) *Transaction Privacy* is ensured by the symmetric encryption, and C) *Transaction Integrity* is ensured by MAC.

**Anonymity:** In order to prevent a merchant from knowing the identity of her clients, usage of client's nickname ($NID_C$) instead of her real identity is required during a communication from $C$ to $M$. Since the C's nickname is known only by the client and the issuer, merchant cannot map the nickname and C's true identity. Thus, client's privacy is protected and untraceable.

**Trust Relationships:** Generally, in any transaction, a party should not trust others unless they can provide a proof of trustworthiness [6]. However, as in our protocol the issuer issues a credit- and/or debit-card to the client and she will not reveal it to any part, we state the trust relationship between the client and the issuer.

## 5 Conclusions and Further work

We have proposed a novel protocol for secure payments in a mobile payment system where the merchant does not have direct communication with the acquirer and the messages among these parties must be done across the client. Our protocol employs a symmetric cryptographic techniques which have lower computation requirements at both parties (since no public-key operation is required) and offers the capability of dealing with protocol failures and disputes among parties.

Although our proposed protocol was designed for a mobile payment system based on a Client Centric scenario (where direct communication between the merchant and acquirer is not possible), the security properties are preserved as if we were working in an scenario with full connectivity among the different entities.

As a result, we state that our scheme can be easily applicable for restricted connectivity scenarios in m-commerce due to the low communication and the light computation. Moreover, our scheme illustrates how a merchant can sell goods in a secure way even if she can not directly communicate with the acquirer.

## References


[1] M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. Van Herreweghen, and M. Waidner. Design, implementation and deployment of the $i$KP secure electronic payment system. {*IEEE*} *Journal on Selected Areas in Communications*, 18(4):611–627, 2000.

[2] S. Chari, P. Kermani, S. Smith, and L. Tassiulas. Security issues in m-commerce: A usage-based taxonomy. In *E-Commerce Agents, Marketplace Solutions, Security Issues, and Supply and Demand*, pages 264–282, 2001.

[3] J. Hall, S. Kilbank, M. Barbeau, and E. Kranakis. Wpp: A secure payment protocol for supporting credit- and debit-card transactions over wireless networks. In *IEEE International Conference on Telecommunications (ICT01)*, 2001.

[4] Z. Hu, Y. Liu, X. Hu, and J. Li. Anonymous micropayments authentication(ama) in mobile data network. In *The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, pages 46–53, 2004.

[5] J. T. Isaac, J. S. Camara, A. I. Manzanares, and M. C. Castro. Payment in a kiosk centric model with mobile and low computational power devices. In *Computational Science and Its Applications (ICCSA 2006), volumen 5*, pages 798–807, 2006.

[6] S. Kungpisdan, B. Srinivasan, and P. D. Le. A secure account-based mobile payment protocol. In *International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 1*, pages 35–39, 2004.

[7] Y. Lei, D. Chen, and Z. Jiang. Generating digital signatures on mobile devices. In *18th International Conference on Advanced Information Networking and Applications (AINA'04), volumen 2*, page 532, 2004.

[8] S. B. Park, M. S. Kang, and S. J. Lee. Authenticated key exchange protocol secure against offline dictionary attack and server compromise. In *Grid and Cooperative Computing, Second International Workshop, GCC'03*, pages 924–931, 2003.

[9] J. L. A. Peiro, N. Asokan, M. Steiner, and M. Waidner. Designing a generic payment service. *IBM Systems Journal*, 37(1):72–88, 1998.

[10] H. Wang and E. Kranakis. Secure wireless payment protocol. In *International Conference on Wireless Networks*, pages 576–578, 2003.

[11] X. Zheng and D. Chen. Study of mobile payments system. In *International Conference on Electronic Commerce*, page 24. IEEE Computer Society, 2003.

# Anonymous Payment in a Client Centric Model for Digital Ecosystems

Jesús Téllez Isaac[1] and José Sierra Cámara[2]

[1] Computer Science Department (Facyt), Universidad de Carabobo, Av. Universidad, Sector Bárbula, Valencia, VENEZUELA, e-mail : jtellez@uc.edu.ve

[2] Computer Science Department, Universidad Carlos III de Madrid, Av. Universidad, 30, Leganés (Madrid), ESPAÑA. e-mail: sierra@ inf.uc3m.es

***Abstract*—Most of the payment protocols designed in recent years for mobile payment systems are based on a scenario where all the entities are directly connected to one another. This scenario (formally called "Full connectivity scenario") offers advantages to protocol's designers because it allows them to simplify the design and development of payment protocols without losing security guarantees. However, the "Full connectivity" scenario does not consider those situations in which the merchant cannot communicate with the acquirer due to absence of Internet access in his/her infrastructure, and the use of other communication technologies is unaffordable due to the inconveniences and costs associated. In order to overcome this restriction and contribute to the progress of m-commerce, in this paper we propose a novel anonymous protocol for a mobile payment system based on a Client Centric Model that employs a digital signature scheme with message recovery using self-certified public keys. As a result, our proposal illustrates how a merchant can sell goods in a secure way even if she can not directly communicate with the acquirer.**

***Index Terms*— Anonymous Protocol, Mobile Payment System, Client Centric Model, Digital Signature with message recovery, Self-certified public keys.**

## I. Introduction

Mobile commerce (m-commerce) concerns to any business transaction conducted electronically between at least two parties (where one of these use a mobile device) over mobile networks. Some typical examples of m-commerce are: a) purchasing airline tickets, b) restaurant booking and reservation, c) hotel booking and reservation and d) paying taxis and public transport.

M-commerce's popularity has increased in the last years thanks to advances in portable devices and the rapid development of the mobile communication technologies which have allowed people to use mobile telephones or Personal Digital Assistant (PDA) to access Internet (to read email, browse web pages or purchase goods) anywhere and anytime.

In spite of advances in the portable devices which make m-commerce more profitable and promising, there is still a widespread skepticism about buying and paying for them on-line, due to the vulnerability of sensitive information when transmitted through communication channels. Therefore, it is necessary to develop mobile payment systems capable of providing safe and trustworthy communications between the customer and on-line mobile services providers. Moreover, these payment systems should overcome the common limitations existing in mobile devices currently available, which prevent that these devices execute, in an efficient way, operations that require a lot of computing resources. The common limitations are: 1) poor computational capabilities, 2) limited storage and 3) short battery life.

Different mobile payment systems have emerged in the last years which allow the payment of services/goods from mobile devices, but the one developed by [17] (called 3-D Secure) has become a standard due to its benefits regarding security and flexibility in the authentication methods. However, the relationship between payee and acquirer is quite strict (although required for Visa's 3D-Secure scheme) and does not allow the use of schemes in which the communication among these parties is not possible due to: 1) the impossibility of the merchant to connect to Internet and 2) the high costs and / or inconveniences of using the infrastructure necessary to implement other mechanisms of communication between the merchant and the acquirer (such SMS, phone call, etc).

Most of the mobile payment systems proposed up until now assume the merchant has Internet connectivity through his/her infrastructure, so the restrictions mentioned previously do not represent an important issue. Nevertheless, there are some situations that the merchant meets in which is not possible to connect to Internet so it becomes necessary to develop mobile payment systems where the payee could sells goods/services even thought she may not have Internet access.

Digital Signature can be represented as a secure base in electronic payment system because it provides authentication, data integrity and non-reputation cryptography services [9]. However, the traditional digital signature schemes are based on asymmetric techniques which make the signature computation very expensive and not suitable for mobile devices. Moreover, these schemes suffer from the well-know authentication problem[1] which requires the usage of certificate to avoid it. The public-key certificate must be verified by a Certificate Authority (CA), and that verification causes an additional information exchange during a transaction.

According to our operational model (where merchant cannot with the acquire in a direct way, formally called Client Centric Model [4]), the above schemes are not suitable because merchant interacts only with a client during a pur-

[1] An imposter may impersonate any innocent user with a valid cryptographic but incorrect public key (because it does not belong to the innocent user).

1-4244-0470-3/07/$20.00 ©2007 IEEE

chase and communication with other party (like CA to verify a certificate) is not possible. Therefore, usage of a non-traditional digital signature scheme is required in order to satisfies our requirements. Digital signature with message recovery using self-certified public keys [6], [15] provides an authenticated encryption scheme that integrates the mechanisms of signature and encryption, which enable only the specified receiver to verify and recover the original message. The authentication of the public key can implicitly be accomplished with the signature verification.

In order to solve the problem of buy and payment of goods/services in our operational model, in section III, we construct a protocol that allows to a merchant to send a message to acquirer through the client (who will not be able to decrypt this message). The proposed protocol (divided in 2 sub--protocols) employs the authentication encryption scheme proposed by [16] that allows only specified receivers to verify and recover the message, so any other receiver will not be able to access the information. Moreover, it supports both credit-card and debit-card transactions and protects the real identity of the clients during the purchase. As a result, our proposal represents an alternative to other mobile payment systems with restrictions regarding a mandatory connection between the merchant and the acquirer, like Visa's 3D Secure.

*Outline of this paper:* We begin by presenting a description of some known results (related work). Following this, we present our approach which includes a complete list of notations used in our scheme, the operational model, the initial assumptions and the proposed protocol. In section IV, a security analysis of the proposed protocol is presented. We end this paper with the conclusions in section V.

## II. Related Work

In recent years, new security and privacy challenges have emerged with the widespread of m-commerce. Several studies have been conducted to unify concepts and scenarios into frameworks that will be useful to develop new electronic payment systems and to analyze security issues.

Recently, [4] conducted a research that unifies many proposed m--commerce usages into a single framework. This research intended to revise the possible range of mobility scenarios, identifying the security issues for each connectivity scenario. As a result, five scenarios were identified and analyzed: Disconnected Interaction, Server Centric Case, Client Centric Case, Full Connectivity and Kiosk Centric Case. The latest has been considered as the starting point in the design of our proposal.

Most of the mobile payments systems proposed until now are based in a scenario where all the entities are directly connected to one another (formally called "Full Connectivity scenario") [6], [10], [17] because it allows protocol's designers to simplify the protocols and obtain stronger security guarantees than similar applications in the others models.

Most of the protocols proposed in recent years for the Full Connectivity scenario are based on public-key infrastructure (PKI) [2], [6], [9] whereas the remaining employ symmetric-key operations which is more suitable for wireless networks [8]. Unfortunately, usage of those protocols within the Kiosk Centric Case mobile scenario is not possible, as it restricts the communication which allows only interaction between the client and the merchant. However, some protocols could be reformulated to overcome this restriction, achieving the same security and performance but in a different scenario. For example, Téllez *et al.* [13] reformulate the mobile payment protocol proposed by [8] to satisfy the requirements of their proposal (based on Kiosk Centric Model).

Many signature schemes with message recovery have been proposed in recent years [3], [15], [16]. These schemes allow a signer's public key to be simultaneously authenticated in verifying the signature. As the public keys does not need to be included in a certificate to be authenticated by verifiers (as happens in protocols based on public-key infrastructure), communication with a Certificate Authority during a transaction to verify the validity of a certificate is not necessary. Therefore, and as shown in [14], digital signature schemes with message recovery are suitable for mobile payment systems based on a kiosk centric model like the one being suggested in this work.

In order to provide limited but practical anonymity by using limited disclosure of information, some proposals have been suggested in the past [1], [7]. While the cryptography techniques and operational models used in those works are different from ours, we follow the approach of using nicknames usage instead of the real identity, implemented in [7] to prevent a merchant from knowing the customer's identity.

## III. OUR APPROACH

### A. Parties and Notations

All the entities involved in our protocol are called parties and communicate through wireless and wired network.

The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer respectively. The following symbols are used to represent other messages and protocols:

- $ID_P$ : the identity of party $P$ that contains the contact information of $P$.
- $NID_C$ : Client's nickname, temporary identity.
- $K_P$ : party's $K$ public key.
- $K_S$ : party's $K$ private key.
- $E_{P\text{-}P'}(X)$ : message X signed and encrypted by the user $ID_P$ to a specified receiver $ID_{P'}$.
- TID : Identity of transaction that includes time and date of the transaction.
- OI: Order information (OI = {TID, OD, $h$(OD, Price)}) where OD and Price are order descriptions and its amount.
- TC : The type of card used in the purchase process (TC={Credit, Debit}).

1-4244-0470-3/07/$20.00 ©2007 IEEE

- Stt : The status of transaction (Stt = {Accepted, Rejected}).
- TIDReq : The request for TID.
- MIDReq : The request for $ID_M$.
- XMReq : The request for $x_M$.
- h(M) : g the one-way hash function of the message M.

### *B. Operational Model*

Generally, operational models for m-commerce found in literature involve transaction between two or more entities. Our operational model is composed of five entities:

1. *Client:* a user who wants to buy goods or services from the merchant. Particulary, in our proposal, the user has a mobile device with the following features: a) low computational power (e.g mobile phone, PDA, etc.), b) equipped with a built-in display, an input method and short range link (such Infrared, Wi-Fi or Bluetooth), c) capability to execute a java program, and d) able to access Internet.
2. *Merchant:* a computational entity that has products or services to offer/sell to the client and with which the user participates in a transaction. This entity could be a normal web server or an intelligent vending machine that interacts with the customer using a short range link. Moreover, this entity connects with the Payment Gateway through a secure wireless channel allowing the merchant to communicate with the acquirer using this connection. A formal definition of the merchant may be found in [4].
3. *Acquirer:* is the merchant's financial institution. It verifies the validity of the deposited payment instrument and manages the merchant's account including fund transfer.
4. *Issuer:* is the customer's financial institution. It provides electronic payment instruments to the client to use in a payment and manage the client's account including fund transfer.
5. *Payment Gateway:* an additional entity that acts as a medium between acquirer/issuer at banking private network side and customer/vendor at the Internet side for clearing purpose [8].

In figure 1, we specify the links among the five entities of our scheme. Note that there is no direct connection involving the merchant and the acquirer. Moreover, the connection between the customer and the merchant (denoted as the dotted arrow) is set up through a wireless channel.

On the other hand, interaction among the client and the payment gateway (depicted as solid arrow in the scheme) should be reliable and secure against passive and active attacks. Therefore, the connection is supposed to be established through a secure wireless channel by using the well-know security protocol like SSL/TLS. Note that the issuer, acquirer and payment gateway operates under the banking private network, so the security of the messages exchanged among them is out of the scope of this paper.

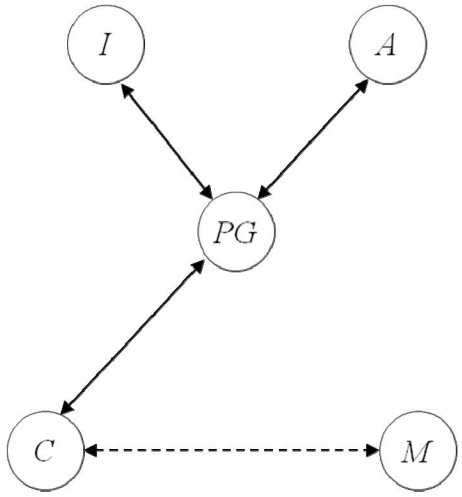


Fig.1 Operational Model

### *C. Initial Assumptions*

The initial assumptions for our proposed protocol can be stated as follows:

1. Client registers herself to an issuer before making payments. The registration can be done either personally at the issuer's premises or via the issuer's website.
2. The Client shares her credit- and/or debit-card information (CDCI) with her issuer (who will not reveal it to any merchant).
3. The trusted system authority (SA) is responsible for generation system parameters in the system initialization phase (by the procedure described in [15], [16]).
4. Every party of the system $P_i$ (whose identity is $ID_{Pi}$) choose a number $K_{Si}$ as her secret key and computes $x_i = g^{K_{Si}} \bmod N$. Then, $P_i$ sends ($x_i$, $ID_{Pi}$) to SA. After receiving ($x_i$, $ID_{Pi}$), the trusted system authority computes and publishes the public key of $P_i$ as $K_{Pi} = ( x_i - ID_{Pi} )^{1/h(ID_{Pi})} \bmod N$ [16]. As the client uses a nickname instead the real identity to protect her privacy, one $K_{Pi}$ must be generated and published for every nickname assigned to the client.
5. When a client registers to a issuer, several nicknames are assigned to the client and those nicknames are known only by the client and the issuer [7].
6. The client holds $C_S$, $ID_I$, and system parameters in her mobile device.
7. The Price and description of the goods and services have been decided by client and merchant.

### *D. Signature with Message Recovery Technique*

In order to a sender $P_i$ (with identity $ID_{Pi}$) sign and encrypt a message W to a specified receiver $P_j$ (with identity $ID_{Pj}$), we follow the generation procedure of signature proposed by [16]. First, $P_i$ chooses a random number $y$ and

1-4244-0470-3/07/$20.00 ©2007 IEEE

computes $r_1$, $r_2$ and $s$. Afterwards, $P_i$ sends the triple ($r_1$, $r_2$, $s$) as the signature of message W (from now on, $E_{Pi\text{-}Pj}$) to the verifier $P_j$. After receiving ($r_1$, $r_2$, $s$), the verifier $P_j$ recovers message W and verifies that the signature is valid using the same procedure described in [16].

### *E. Detailed Protocols*

Our protocol consists of two sub--protocols. In *Merchant Registration Protocol*, client shares the values $NID_C$ and $x_C$ with the merchant. Then, if the client does not have the payment software (also called wallet software, and that from now on we will assume that is programmed using the Java language due to the multiplatform capabilities of this language), she connects to issuer's web site to download it or sends a request to the issuer in order to receive it by mail. After the merchant receives $NID_C$, $x_C$ and the wallet software is available in client's mobile device, the client can start the *Payment Protocol*. The main functions of both protocols are shown as follow:

Merchant Registration Protocol

**C → M:** $\{ NID_C, x_C, n, MIDReq \}_w$
**M → C:** $\{ ID_M, h(n, NID_C, x_C, ID_M) \}_w$
**C → M:** $h(ID_M, n)$

As the merchant can not connect directly with the trusted system authority during a payment, the merchant will not be able to access the public value $x_C$ used in the signature process. In consequence, the client needs to send this value to merchant. First, $C$ sends to $M$ her nickname $NID_C$, a nonce $n$ for challenge-response and MIDReq, encrypted with a session key $w$ generated by running AKE protocol [2] with $C$. Then, $M$ confirms $C$'s registration by sending the merchant's identity ($ID_M$) and $h( n, NID_C, x_C, ID_M)$, encrypted with the session key $w$. Finally, $C$ sends $h(ID_M, n)$ to $M$ as a confirmation to have received $ID_M$.

Afterwards, if the wallet software is not available in the mobile device, the client connects to issuer's web site to download it or sends a request to the issuer to receive it by mail. Once the client has received the software, she will install it in the mobile device.

The issuer intends to protect the software against various types of attacks carried away at any moment, preparing the software following these steps:

- Choose one of the obfuscation methods proposed by [12] and apply it to the java code.
- Then, the software is signed using the Authenticated encryption scheme with message linkages proposed by [3], [16].

Payment Protocol

**1) C → M:** $NID_C$, *TIDReq*
**M → C:** $E_{M\text{-}C}(TID, ID_M)$

**2) C → M:** $E_{C\text{-}M}(OI, Price, ID_I, h(OI, ID_I))$

**3) M → C:** $E_{M\text{-}C}(VCRequest)$
$VCRequest = E_{M\text{-}PG}(h(OI), TID, Price, NID_C, ID_I)$

**4) C → PG:** $E_{C\text{-}PG}(OI, Price, ID_I, ID_M, TID,$
$VSRequest, VCRequest,$
$h(OI, ID_M, ID_I, TID)), NID_C$
$VSRequest = E_{C\text{-}I}(Price, h(OI), TC, ID_M)$

**5)** Under banking private network,
**5.1) PG → I:** $VSRequest, h(OI), TID,$
$Price, NID_C, ID_M$
**5.2) PG → A:** $Price, ID_M$
**5.3) I,A → PG:** $VSResponse, Stt, h(Stt, h(OI))$
$VSResponse = E_{I\text{-}C}(Stt, h(OI))$

**6) PG → C:** $PResponse$
$VCResponse = E_{PG\text{-}M}(Stt, h(Stt,h(OI)))$
$PResponse = E_{PG\text{-}C}(VSResponse, VCResponse)$

**7) C→ M:** $E_{C\text{-}M}$(VCResponse)
$VCesponse = E_{M\text{-}C}(VSResponse)$

**Step 1:** The client **C** sends her nickname ($NID_C$) and the request for the transaction identity (*TIDReq*) to **M**. Once the request is received by **M**, she sends her identity ($ID_M$) and *TID* to **C**, encrypted only to be recovered by the client.

**Step 2: C** creates a *Payment Request* (referred to the General Payment Model described in [10], [8]) including I's identity, *Price* and *OI* (used to inform **M** about the goods and prices requested). The client **C** encrypts the *Payment Request* to be recovered only by **M** and sends it to the merchant. Note that some important fields, such as *OI*, *Price*, $NID_C$, $ID_I$$, are hashed in order to check if they are modified or replaced with others while in transit.

**Step 3: M** decrypts the message received from **C** to retrieve *OI*. **M** prepares the *Value-Claim Request* (called *VCRequest*), encrypted to be recovered only by **PG** in order to ensure that only the payment gateway is the intended recipient of the message. Once the *VCRequest* has been prepared, **M** sends it to **C**, encrypted to be recovered only by the client. The *Value-Claim Request* contains C's nickname, I's identity, order's amount, identity of transaction and the hash of the order information.

**Step 4: C** decrypts the message received from **M** to recover *VCRequest*. Then, **C** prepares the *Value-Substraction Request* (called *VSRequest*, which is encrypted to be recovered only by an issuer **I**) including *Price*, *TC*, $ID_M$ and *h(OI)*. Once the *VSResquest* has been prepared, the client **C** prepares another message (which includes *OI*, *Price*, $ID_I$, *TID*, *VSRequest*, the forwarded *Value-Claim Request* and some important fields hashed to ensure that they will not be modified or replaced with others while in transit) encrypted to the specific receiver **PG**. The later encrypted message is sent by the client **C** to **PG** with her nickname.

**Step 5: PG** decrypts the message received from **C** to re-

1-4244-0470-3/07/$20.00 ©2007 IEEE

trieve *VSRequest* and the others fields included in *VCRequest*. Then, **PG** forwards *VSRequest* and other important information, namely: $h(OI)$, *TID*, *Price*, $NID_C$, $ID_I$ to **I** who will process it to approve or reject the transaction.

Also, **PG** sends $ID_M$ and the requested price (*Price*) to claim to acquirer **A** that she is the party whom the requested amount *Price* will be transferred to. After checking the validity of the client's account, the total amount of *OI* is transferred to the merchant's account, the issuer **I** prepares *Value-Substraction Response* (called *VSResponse*) and sends it to **PG** with the approval result (*Stt*). Note that *VSResponse* is encrypted to be recovered only by an issuer **C**.

**Step 6: PG** sends *Payment Response* (called *PResponse*) encrypted to be recovered only by **C**. *PResponse* include *VSResponse* and *VCResponse* (which will be forwarded to **M**).

**Step 7:** After receiving *PResponse*, **C** decrypts it to retrieve *VSResponse* and *VCReponse*. As **C** has her own *OI*, she can compare this field with the received $h(OI)$ to check whether or not the message is the response of her request. If they are not matched, **C** sends a message to the PG pointing the problem (then, the payment gateway can start a recovery procedure or resend the message).

Otherwise, **C** encrypts *Value-Claim Response* to be recovered only by **M** and sends it to the merchant who in turn proceeds to deliver the goods to the client.

## IV. ANALYSIS AND DISCUSSIONS

*Security issues*

Transaction Security

- Authentication between the client and the issuer: The operational model used in this proposal has a communication restriction: client can not communicates directly with the issuer. Therefore, in order to allow that issuer authenticates a client, *C* has to send a message to *I* (through the merchant) with the following features: 1) resistant to attacks while in transit, 2) recoverable only by the issuer, and 3) able to assure that it has been created and sent by *C*.

  Since the authenticated encryption scheme used in our protocol integrates the mechanisms of signature and encryption, the message *VSRequest* sent by *C* to *I*, satisfies all the requirements mentioned above and can be used by the issuer to authenticate the client.

- Confidentiality: The authentication encryption scheme used in our protocol ensures the encryption of important data of per transaction while in transit. Moreover, since this scheme allows that only the specified receiver can verify and recover the message, any other receiver is unable to do it. For example, the *VSRequest* is created by *C* and encrypted to be recovered only by *I*. Any other party couldn't decrypt the message because requires the issuer's private key which is known only by *I*.

- Integrity: It is important to protect data from being modified or/and replaced while in transit. To achieve that, usage of message digest algorithms is require. In our protocol, integrity is ensured by using hash value of some important information (e.g, *Order Information*, *Client's nickname*, etc), padded into some messages.

Anonymity: In order to prevent a merchant from knowing the identity of her clients, usage of client's nickname ($NID_C$) instead of her real identity is required during a communication from *C* to *M*. Since the C's nickname is known only by the client and the issuer, merchant cannot map the nickname and C's true identity. Thus, client's privacy is protected and untraceable.

Trust Relationships: Generally, in any transaction, a party should not trust others unless they can provide a proof of trustworthiness [8]. However, as in our protocol the issuer issues a credit- and/or debit-card to the client and she will not reveal it to any part, we state the trust relationship between the client and the issuer.

Non-repudiation of Origin (NRO): The Non-repudiation of Origin is ensured since the signature of a message is generated by the signer U using her private key $U_S$ (known only by U). Therefore, the signer should not be able to repudiate his signature creation later.

## V. CONCLUSIONS AND FURTHER WORK

We have proposed a novel protocol for secure payments in a mobile payment system where the merchant does not have direct communication with the acquirer and the messages among these parties must be done across the client. Our protocol employs a digital signature scheme with message recovery using self-certified public keys which allows to a client to make purchases without disclosing private information.

Although our proposed protocol was designed for a mobile payment system based on a Client Centric scenario (where direct communication between the merchant and acquirer is not possible), the security properties are preserved as if we were working in an scenario with full connectivity among the different entities.

Our proposal represents an alternative to all mobile payment systems based on the Full Connectivity scenario (including Visa's 3-D Secure scheme) where communication between the merchant and acquirer is mandatory. As a result, we assert that our proposed protocol illustrates how a merchant can sell goods in a secure way even if she can not directly communicate with the acquirer.

As the digital scheme employed in the proposed protocol includes only non-repudiation of origin, it will be valuable in the future incorporate more non-repudiation services (such as non-repudiation of receipt, non-repudiation of

1-4244-0470-3/07/$20.00 ©2007 IEEE

submission, etc.) in order to prevent entities from denying that they have sent or received certain messages.

## VI. References


[1] N. Asokan," Anonymity in a Mobile Computing Environment", in Proceedings of IEEE Workshop on Mobile Computing Systems and Applications. IEEE, 1994, pp. 200–204.

[2] M. Bellare, J. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. V. Herreweghen, and M. Waidner, "Design, implementation, and deployment of the iKP secure electronic payment system", IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 611–627, 2000.

[3] Y. Chang, C. Chang, and H. Huang, "Digital signature with message recovery using self-certified public keys without trustworthy system authority", Journal of Applied Mathematics and Computation, vol. 161, no. 1, pp. 211–227, 2005.

[4] S. Chari, P. Kermani, S. Smith, and L. Tassiulas, "Security Issues in M-Commerce: A Usage-Based Taxonomy", in Proceedings of E-Commerce Agents, 2001, pp. 264–282.

[5] M. Girault, "Self-Certified Public Keys", in Proceedings of EUROCRYPT, 1991, pp. 490–497.

[6] J. Hall, S. Kilbank, M. Barbeau, and E. Kranakis, "WPP: A Secure Payment Protocol for Supporting Credit- and Debit-Card Transactions over Wireless Networks", in Proceedings of International Conference on Telecommunications (ICT 2001). IEEE, 2001.

[7] Z. Hu, Y. Liu, X. Hu, and J. Li, "Anonymous Micropayments Authentication(AMA) in Mobile Data Network", in Proceedings of INFOCOM, 2004.

[8] S. Kungpisdan, B. Srinivasan, and P. D. Le, "A Secure Account-Based Mobile Payment Protocol", in Proceedings of ITCC (1), 2004, pp. 35–39.

[9] Y. Lei, D. Chen, and Z. Jiang, "Generating Digital Signatures on Mobile Devices", in Proceedings of AINA (2), 2004, pp. 532–535.

[10] J. L. A. Peiro, N. Asokan, M. Steiner, and M. Waidner, "Designing a generic payment service", IBM Syst. J., vol. 34, no. 1, pp.72–80, 1997.

[11] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", in Proceedings of CRYPTO, 1984, pp. 47–53.

[12] M. Sosonkin, G. Naumovich, and N. D. Memon, "Obfuscation of design intent in object-oriented applications", in Proceedings of Digital Rights Management Workshop, 2003, pp. 142–153.

[13] J. Téllez, J. Sierra, A. Izquierdo, and M. Carbonell, "Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices", in Proceedings of ICCSA (5), 2006, pp. 798–807.

[14] J. Téllez, J. Sierra , A. Izquierdo and J. Márquez, "Anonymous Payment in a Kiosk Centric Model with Mobile using Digital signature scheme with message recovery and Low Computational Power Devices", Journal of Theoretical and Applied Electronic Commerce Research, vol. 1, no. 2, pp. 1-11, 2006.

[15] Y. Tseng, J. Jan, and H. Chien, Digital signature with message recovery using self-certified public keys and its variants, Journal of Applied Mathematics and Computation, vol. 136, no. 2–3, 2003.

[16] J. Zhang, W. Zou, D. Chen, and Y. Wang, "On the Security of a Digital Signature with Message Recovery using Self-certified Public Key", Soft Computing in Multimedia Processing Special Issue of the Informatica Journal, vol. 29, no. 3, pp. 343–346, 2005.

[17] V. International, (2002). 3-d secure mobile authentication scenarios version 1.0. [Online], Available: http://partnernetwork.visa.com/pf/3dsec/specifications.jsp.


## VII. Acknowledgements

This work was supported in part by ASPECTS-M Project (Reference Models for Secure Architectures in Mobile Electronic Payments), CICYT-2004, however it represents the view of the authors.

1-4244-0470-3/07/$20.00 ©2007 IEEE

# A Secure Payment Protocol for Restricted Connectivity Scenarios in M-Commerce

Jesús Téllez Isaac[1] and José Sierra Camara[2]

[1] Universidad de Carabobo, Computer Science Department (Facyt)
Av. Universidad, Sector Bárbula, Valencia, Venezuela
jtellez@uc.edu.ve
[2] Universidad Carlos III de Madrid, Computer Science Department,
Avda. de la Universidad, 30, 28911, Leganés (Madrid), Spain
sierra@inf.uc3m.es

**Abstract.** A significant number of mobile payment systems have been proposed in recent years, most of them based on a scenario where all the entities are directly connected one to another (formally called "Full connectivity scenario"). Despite of the advantages that the aforementioned scenario offers to protocol's designers, regarding design simplification and development of payment protocols without losing security capabilities, the full connectivity scenario does not consider those situations in which the client cannot directly communicate with the issuer (Kiosk Centric Model) or the merchant has no direct communication with the acquirer (Client Centric Model). In order to overcome this restriction and contribute to the progress of m-commerce, in this paper we propose an anonymous protocol that uses a digital signature scheme with message recovery using self-certified public keys that is suitable for both the Kiosk Centric Model and Client Centric Model. As a result, our proposal shows that m-commerce is possible in restrictive connectivity scenarios, achieving the same security capabilities than other protocols designed for mobile payment systems based on "Full connectivity scenario".

**Keywords:** Payment Protocol, Self-certified public keys, Digital Signature with message recovery, Mobile Payment System.

## 1 Introduction

Several mobile payment systems have emerged in the last years which allow payments for services and goods from mobile devices using different kinds of payments: credit-card payments, micropayments and digital coins. The relationship between payee and acquirer is quite strict in most of these mobile payment systems and does not allow the use of schemes in which the communication among these parties is not possible due to: 1) the impossibility of the merchant to connect to Internet and 2) the high costs and/or inconveniences of using the infrastructure necessary to implement other mechanisms of communication between the merchant and the acquirer (such SMS, phone call, etc.).

G. Psaila and R. Wagner (Eds.): EC-Web 2007, LNCS 4655, pp. 1–10, 2007.

The above restrictions do not represent an important issue for the majority of mobile payment systems proposed up until now because they assume that engaging parties are able to connect to Internet. Nevertheless, in the real world there are some situations that the merchant meets in which it is not possible to connect to the Internet, so it becomes necessary to develop mobile payment systems where the payee could sell goods/services even thought he/she may not have Internet access.

According to our operational models (where client cannot communicate directly with issuer, or merchant cannot communicate with the acquirer in a direct way, the traditional digital signature schemes based on asymmetric techniques are not suitable because one party (client or merchant, depending on the scenario) has connectivity restrictions and consequently, communication with others parties (as a CA, for verifying a certificate) is not possible during a purchase. Therefore, usage of a non-traditional digital signature scheme is required in order to satisfy our requirements.

In order to eliminate the restriction of those mobile payment systems based on the Full Connectivity Scenario regarding the direct communication between client and issuer, and among merchant and acquirer for authentication purposes, in section 3, we design a protocol that allows to a party (A) to send a message to another peer (B) through a third party (who will not be able to decrypt this message) in the those scenarios. The proposed protocol employs the authentication encryption scheme proposed by [13] that allows only specified receivers to verify and recover the message, so any other receiver will not be able to access the information. Moreover, it supports both credit-card and debit-card transactions and protects the real identity of the clients during the purchase. As a result, our proposal represents an alternative to other mobile payment systems with restrictions regarding a mandatory connection among two of its parties.

*Outline of this paper:* We begin by presenting the related work. Then, we present our approach which includes a complete list of notations, the operational model and the proposed protocol. In section 4, a security analysis of the proposed protocol is presented. We end this paper with the conclusions in section 5.

## 2 Related Work

Recently, [3] conducted a research that unifies many proposed m–commerce usages into a single framework. This research intended to revise the possible range of mobility scenarios, identifying the security issues for each connectivity scenario. As a result, five scenarios were identified and analyzed: Disconnected Interaction, Server Centric Case, Full Connectivity, Kiosk Centric Case and Client Centric Case. The last two have been considered as the starting point in the design of our proposal.

Most of the protocols proposed in recent years for the Full Connectivity scenario are based on public-key infrastructure (PKI) [1,4,8,12] whereas the remaining employ symmetric-key operations which is more suitable for wireless networks [7]. Unfortunately, usage of those protocols is not possible in scenarios

where direct interaction among two of its parties is not allowed due to the communication restriction imposed by the model (as happens in Kiosk Centric Model or Client Centric Model). However, some protocols could be reformulated to overcome this restriction (achieving the same security and performance levels, but in a different scenario), while being suitable for mobile payment systems with Restricted Connectivity. For example, Téllez *et al.* [9] reformulate the mobile payment protocol proposed by [7] to satisfy the requirements of their proposal.

A few number of signatures schemes with message recovery have been proposed in recent years which illustrate how a signer's public key can be simultaneously securely authenticated during the signature verification, avoiding communication with a Certificate Authority during a transaction in order to verify the validity of a certificate since the certificate is embedded in public key itself. Therefore, and as shown in [10], digital scheme signature schemes with message recovery are suitable for mobile payment protocols based on a restrictive connectivity scenarios like the one being suggested in this work.

## 3 Our Approach

### 3.1 Parties and Notations

All the entities involved in our protocol are called parties and communicate through wireless and wired network. The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer respectively. The following symbols are used to represent other messages and protocols:

- $\mathrm{ID}_P$ : the identity of party $P$ that contains the contact information of $P$.
- $\mathrm{NID}_C$ : Client's nickname, temporary identity.
- $\mathrm{K}_P$ : party's $K$ public key.
- $\mathrm{K}_S$ : party's $K$ private key.
- $\mathrm{E}_{P-P'}(\mathrm{X})$ : message X signed and encrypted by $ID_P$ to a specified receiver $ID_{P'}$, following the generation procedure of signature proposed by [13].
- TID: Identity of transaction that includes time and date of the transaction.
- OI: Order information (OI = {TID, OD, $h$(OD, Price)}) where OD and Price are order descriptions and its amount.
- TC: The type of card used in the purchase process (TC={Credit, Debit}).
- TS: The type of scenario used during a payment (TC={Kiosk, Client})
- DCMA : The status of the direct connection between the merchant and the acquirer (DCMA = {Connected, NO-Connected}). The default value is *NO-Connected*.
- DCCI : The status of the direct connection between the client and the issuer (DCCI = {Connected, NO-Connected}). The default value is *NO-Connected*.
- Stt: The status of transaction (Stt = {Accepted, Rejected}).
- TIDReq : The request for TID.
- MIDReq : The request for $\mathrm{ID}_M$.
- MPReq : The request for $\mathrm{M}_P$.
- DCMAReq : The request for DCMA.
- $h(M)$ : the one-way hash function of the message M.

### 3.2 Operational Model

Our operational models, Kiosk Centric Model and Client Centric Model (figure 1 and figure 2, respectively), are composed of five entities:

1. *Client*: a user who wants to buy goods or services from the merchant, equipped with a short range link (such Infrared, Wi-Fi or Bluetooth). Only in the Client Centric Model, the client is able to access Internet.
2. *Merchant*: a computational entity (such an intelligent vending machine) that offers or sells products or services to the client, and with which the user participates in a transaction using a short range link. In Kiosk Centric Model, this entity connects with the Payment Gateway through a secure channel allowing the merchant to communicate with the acquirer using this connection whereas in Client Centric Model, direct communication with the Issuer is not possible so it must take place through the client.
3. *Acquirer*: is the merchant's financial institution.
4. *Issuer*: is the customer's financial institution.
5. *Payment Gateway*: an additional entity that acts as a medium between acquirer/issuer at banking private network side and client/vendor at the Internet side for clearing purpose [7].

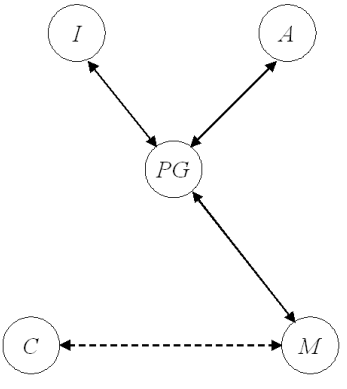


**Fig. 1.** Kiosk Centric Model

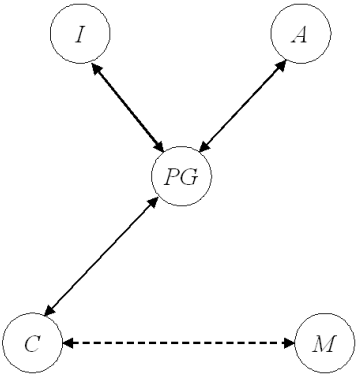


**Fig. 2.** Client Centric Model

The links among the five entities of our operational models are specified in figure 1 and 2. Note that, in both operational models, the connection between the client and the merchant (denoted as the dotted arrow) is setup up through a wireless channel.

On the other hand, interaction among client and payment gateway or between merchant and payment gateway (depicted as the solid arrow in any of the operational models) should be reliable and secure against passive and active attacks. Note that the issuer, acquirer and payment gateway operates under the banking private network, so the security of the messages exchanged among them is out of the scope of this paper.

### 3.3 Initial Assumptions

The initial assumptions for our proposed protocol can be stated as follows:

1. Client registers herself to an issuer before making payments. The registration can be done either personally at the issuer's premises or via the issuer's website. During the above process, the client shares her credit- and/or debit-card information (CDCI) with her issuer (who will not reveal it to any merchant). On the other hand, the issuer assigns several nicknames to the client and those nicknames are know only by the client and the issuer [6]. In the Kiosk Centric Model, the client sends (with the assistance of the issuer) her nicknames and $x_C$ to SA and receives all system parameters from the SA.
2. The system authority (SA) is responsible for generation of the system parameters in the system initialization phase (as described in [13][11]).
3. Every party of the system $P_i$ (whose identity is $ID_{P_i}$) choose a number $\mathrm{K}_{S_i}$ as her secret key and computes $x_i = \mathrm{g}^{K_{S_i}} \bmod N$. Then, $P_i$ sends ($x_i$, $ID_{P_i}$) to SA. After receiving ($x_i$, $ID_{P_i}$), the SA computes and publishes the public key of $P_i$ as $K_{P_i}=(x_i - ID_{P_i})^{h^{-1}(ID_{P_i})} \bmod N$ [13]. As the client uses a nickname instead the real identity to protect her privacy, one $K_{P_i}$ must be generated and published for every nickname assigned to the client.
4. The client holds $C_S$, $ID_I$, and system parameters in her mobile device. Also, in Kiosk Centric Model, client holds $I_P$.

### 3.4 Detailed Protocols

Our Protocol consists of two sub–protocols: the *Merchant Registration Protocol* and the *Payment Protocol*. The main functions of both protocols are shown as follows:

**Merchant Registration Protocol**

$$
\begin{aligned}
&\mathbf{C} \rightarrow \mathbf{M}: \ \{NID_C, n, MIDReq, DCMAReq\}_w \\
&\mathbf{M} \rightarrow \mathbf{C}: \ \{ID_M, h(n, NID_C, ID_M)\}_w \\
&\mathbf{C} \rightarrow \mathbf{M}: \ \text{IF } (TS = \text{"}Kiosk\text{"}) \text{ THEN} \\
&\qquad\qquad \{n, MPReq\}_w \\
&\qquad\quad \text{ELSE} \\
&\qquad\qquad \{n, C_P\}_w \\
&\mathbf{M} \rightarrow \mathbf{C}: \ \text{IF } (TS = \text{"}Kiosk\text{"}) \text{ THEN} \\
&\qquad\qquad \{n, M_P, h(n, M_P)\}_w \\
&\qquad\quad \text{ELSE} \\
&\qquad\qquad \{n, C_P, h(n, C_P)\}_w
\end{aligned}
$$

As our protocol is designed to work on two different operational models, the first step is to determine in which one them the payment is going to take place. First, **C** assigns the value *Connected* to $DCCI$ if he/she is able to connect to internet from his/her mobile device. Then, **C** sends to **M** her nickname $NID_C$,

a nonce $n$ for challenge-response, $MIDReq$, $DCMAReq$ and $h(n, NID_C, ID_M)$, encrypted with a session key $w$ generated by running AKE protocol [1] with $C$. After **M** receives the message, he/she sends the merchant's identity, $DCMA$, encrypted with the session key $w$. Note that **M** assigns the value *Connected* is she/he has direct communication with the acquirer.

After **C** receives $DCMA$, he/she will determine the operational model to be used by the protocol, comparing $DCCI$ with $DCMA$ and takes the decision following this rule:

IF ($DCCI$="NO-Connected") AND ($DCMA$ = "Connected") THEN
    ASSIGN the value "*Kiosk*" TO $TS$
ELSE
    ASSIGN the value "*Client*" TO $TS$

Once the operational model to be used has been decided, **C** prepares a new message and sends it to **M**. The message includes a new nonce $n$ for challenge-response and MPReq (if the value of $TS$ equals "*Kiosk*") or $C_P$ (when the value of $TS$ is "*Client*"). After **M** receives the message, **M** confirms C's registration by sending another message, encrypted with the session key $w$. This message includes $n$, $M_P$ and $h(n, M_P)$ when $TS$ has the value "Client", or $n$, $C_P$ and $h(n, C_P)$ if the value of $TS$ equals to "Kiosk".

Afterwards, if the wallet software is not available in the mobile device, the client will obtain it through one of the following methods: 1) Sending a software request to $I$, in the Kiosk Centric Model, or 2) Connecting to issuer's web site to download it or sending a request to the issuer to receive it by mail. This method is valid for the Client Centric Model.

Once the client has received the software, she will install it in the mobile device. Note that, in the Kiosk Centric Model, the software is signed using the Authenticated encryption scheme with message linkages proposed by [2,13].

**Payment Protocol**

**1)** **C** $\rightarrow$ **M:** $NID_C, TIDReq$
**M** $\rightarrow$ **C:** $E_{M-C}(TID, ID_M)$

**2)** **C** $\rightarrow$ **M:** $E_{C-M}(OI, Price, NID_C, ID_I, VSRequest)$
$VSRequest = E_{C-I}(Price, OI, TC, ID_M)$

**3)** **Merchant:** IF ($TS$ = "$Kiosk$") THEN
**M** $\rightarrow$ **PG:** $VCRequest, ID_M, TS$
ELSE
**M** $\rightarrow$ **C:** $E_{M-C}(VCRequest)$
$VCRequest = E_{M-PG}(VSRequest, h(OI), TID, Price, NID_C, ID_I)$

**4)** **Client:** IF ($TS$ = "$Client$") THEN
**C** $\rightarrow$ **PG:** $E_{C-PG}(VCRequest, ID_M, TS), NID_C$

**5)** Under banking private network,
**5.1) PG → I:** $VSRequest, h(OI), TID, Price, NID_C, ID_M$
**5.2) PG → A:** $Price, ID_M$
**5.3) I,A → PG:** $VSResponse, Stt, h(Stt, h(OI))$
$VSResponse = E_{I-C}(Stt, h(OI))$

**6) Payment Gateway:** IF ($TS =$ "$Kiosk$") THEN
**PG → M:** $VCResponse$
ELSE
**PG → C:** $PResponse$
$VCResponse = E_{PG-M}(Stt, VSResponse, h(Stt, h(OI))$
$PResponse = E_{PG-C}(VSResponse, VCResponse)$

**7) Merchant:** IF ($TS =$ "$Kiosk$") THEN
**M → PG:** $E_{M-C}(PResponse)$
$PResponse = E_{M-C}(VSResponse)$

**8) Client:** IF ($TS =$ "$Client$") THEN
**C → M:** $E_{C-M}(PResponse)$

**Step 1:** The client **C** and the merchant **M** exchange the information necessary to start the protocol.

**Step 2: C** creates a *Payment Request* (referred to the General Payment Model described in [7]) including C's nickname, I's identity, *Price*, *OI* (used to inform **M** about the goods and prices requested) and *Value-Substraction Request* (called *VSRequest*, which is encrypted to be recovered only by an issuer **I** and includes *Price*, *TC*, $ID_M$ and OI). The client **C** encrypts the *Payment Request* to be recovered only by **M** and sends it to the merchant.

**Step 3: M** decrypts the message received from **C** to retrieve *OI*. **M** prepares the *Value-Claim Request* (called *VCRequest*, which contains C's nickname, $ID_I$, *Price*, identity of transaction and order information), encrypted to be recovered only by **PG** in order to ensure that only the payment gateway is the intended recipient of the message. Once the *VCRequest* has been prepared, if the value of $TS$ equals "*Kiosk*", **M** sends it to **PG** with M's identity and $TS$. Otherwise, **M** sends *VCRequest* to **C**, encrypted to be recovered only by the client.

**Step 4:** This step occurs only when the value of $TS$ is "*Client*". **C** decrypts the message received from **M** to recover *VCRequest*. Then, **C** prepares a new message (which includes $ID_M$, $TS$ and the forwarded $Value - Claim\ Request$) encrypted to the specific receiver **PG**. The later encrypted message is sent by the client **C** to **PG** with her/his nickname ($NID_C$).

**Step 5: PG** decrypts the message received from **C** to retrieve *VSRequest* and the others fields included in *VCRequest*. Then, **PG** forwards *VSRequest* and other important information, namely: *h(OI)*, *TID*, *Price*, $NID_C$, $ID_I$ to **I** who will process it to approve or reject the transaction. Also, **PG** sends $ID_M$ and the requested price (*Price*) to claim to acquirer **A** that she is the party whom

the requested amount *Price* will be transferred to. After checking the validity of the client's account, the total amount of *OI* is transferred to the merchant's account, the issuer **I** prepares *Value-Substraction Response* (called *VSResponse*) and sends it to **PG** with the approval result (*Stt*). Note that *VSResponse* is encrypted to be recovered only by an issuer **C**.

**Step 6:** If the value of $TS$ equals "*Kiosk*", the payment gateway (**PG**) sends $Value - Claim\ Response$ (called $VCResponse$, that includes $VSResponse$, which will be forwarded to **C**) encrypted to be recovered only by **M**. Otherwise, **PG** sends *Payment Response* (called *PResponse*) encrypted to be recovered only by **C**. *PResponse* includes $VSResponse$ and $VCResponse$ (which will be forwarded to **M**).

Note that in the Kiosk Centric Model (when the value of $TS$ is "*Kiosk*"), as **M** has her/his own *OI*, she/he can compare this field with the received *h(OI)* to check whether or not the message is the response of her/his request. If they are not matched, **M** sends a message to the **PG** pointing the problem.

**Step 7:** This step occurs only when the value of $TS$ is "*Kiosk*". **M** encrypts $Value - Subtraction\ Response$ to be recovered only by **C**. Then, **M** sends it to **C** as *Payment Response* (called *PResponse*). Once **C** receives the message, decrypts it to retrieve the result of her/his request.

**Step 8:** This step is performed only if the value of $TS$ equals "*Client*". After receiving *PResponse*, **C** decrypts it to retrieve $VSResponse$ and $VCReponse$. Then, **C** compares her/his own *OI* with the received $h(OI)$ to check whether or not the message is the response of her/his request. If they are not matched, **C** sends a message to the **PG** pointing the problem (then, the payment gateway can start a recovery procedure or resend the message). Otherwise, **C** encrypts $Value-Claim\ Response$ to be recovered only by **M** and sends it to the merchant who in turn proceeds to deliver the goods to the client.

Once the purchase has been completed, the client does not have to run *Merchant Registration Protocol* again unless she wants to perform transaction with a new merchant. Note that, after client finish all purchase with a merchant, she will remove merchant's information from her mobile device.

## 4 Analysis and Discussions

### 4.1 Security Issues

**Transaction Security**

- **Authentication:** Each one of the operational models used in this proposal have a communication restriction. Therefore, in order to allow a party (A) to authenticate another peer (B), a sender has to send a message to a specified receiver through a third party with the following features: 1) resistant to attacks while in transit, 2) recoverable only by the specified receiver, and 3) able to assure that it has been created and sent by the sender.

  As the authenticated encryption scheme used in our protocol integrates the mechanisms for signature and encryption, it satisfies all the requirements

mentioned above and can be used by any party of the system to authenticate another peer in a secure way.
- **Confidentiality:** In our protocol, the important data of each transaction is protected while in transit because the scheme used enables only the specified receiver to verify & recover the message transmitted by the sender, while any other party cannot perform such operations.
- **Integrity:** Any important data should be protected from being modified or/and replaced while in transit. In our protocol, the integrity is ensured by the digital signature with message recovery technique.

**Anonymity:** In order to prevent a merchant from knowing the identity of a client, a nickname $NID_C$ instead of his/her real identity is used during a communication from $C$ to $M$. Therefore, as merchant cannot map the nickname and C'S true identity, client's privacy is protected and untraceable.

**Non-Repudiation of Origin (NRO):** As in our protocol, a signer U generates the signature of a message using his/her private key $U_S$ (known only by U), he/she should not be able to repudiate his signature creation later. Therefore, Non-repudiation of Origin is ensured.

## 5 Conclusions and Further Work

We have proposed a secure protocol for secure payments in a mobile payment system where client cannot directly communicate with the issuer or the merchant has no direct communication with the acquirer. Therefore, the messages among the parties that can not communicate directly must be done across a third party.

The proposed protocol employs a digital signature scheme with message recovery using self-certified public keys which allows to a client to make purchases without disclosing private information and using a feasible connection with the merchant through a short range link.

Our proposal represents an alternative to all mobile payment systems based on the "Full Connectivity scenario" where communication among all the engaging parties is mandatory. As a result, we assert that our proposal shows how m-commerce is possible in restrictive connectivity scenarios, achieving the same security capabilities than other protocols designed for mobile payment systems based on "Full connectivity scenario".

As the digital scheme employed in the proposed protocol includes only non-repudiation of origin, it will be valuable in the future incorporate more non-repudiation services in order to prevent entities from denying that they have sent or received certain messages.

## Acknowledgement

This work was supported in part by ASPECTS-M Project (Reference Models for Secure Architectures in Mobile Electronic Payments), CICYT-2004, however it represents the view of the authors.

## References


1. Bellare, M., Garay, J., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G., Herreweghen, E., Waidner, M.: Design, implementation and deployment of the iKP secure electronic payment system. IEEE Journal on Selected Areas in Communications 18(4), 611–627 (2000)
2. Chang, Y., Chang, C., Huang, H.: Digital signature with message recovery using self-certified public keys without trustworthy system authority. Applied Mathematics and Computation 161(1), 211–227 (2005)
3. Chari, S., Kermani, P., Smith, S., Tassiulas, L.: Security issues in m-commerce: A usage based taxonomy. In: Liu, J., Ye, Y. (eds.) E-Commerce Agents. LNCS (LNAI), vol. 2033, pp. 264–282. Springer, Heidelberg (2001)
4. Hall, J., Kilbank, S., Barbeau, M., Kranakis, E.: WPP: A Secure Payment Protocol for Supporting Credit- and Debit-card Transactions Over Wireless Networks. In: IEEE International Conference on Telecommunications (ICT) (2001)
5. Ham, W., Choi, H., Xie, Y., Lee, M., Kim, K.: A secure one-way mobile payment system keeping low computation in mobile devices. In: WISA2002. LNCS, pp. 287–301. Springer, Heidelberg (2002)
6. Hu, Z., Liu, Y., Hu, X., Li, J.: Anonymous Micropayments Authentication (AMA) in Mobile Data Network. In: The 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM), pp. 7–11 (2004)
7. Kungpisdan, S.: A secure account-based mobile payment system protocol. In: International Conference on Information Technology: Coding and Computing (ITCC), pp. 35–39 (2004)
8. Lei, Y., Chen, D., Jiang, Z.: Generating digital signatures on mobile devices. In: 18th International Conference on Advanced Information Networking and Applications (AINA 2004), pp. 532–535. IEEE Computer Society, Los Alamitos (2004)
9. Téllez, J., Sierra, J., Izquierdo, A., Carbonell, M.: Payment in a Kiosk Centric Model with Mobile and Low Computational Power Devices. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 798–807. Springer, Heidelberg (2006)
10. Téllez, J., Sierra, J., Izquierdo, A., Márquez, J.: Anonymous Payment in a Kiosk Centric Model with Mobile using Digital signature scheme with message recovery and Low Computational Power Devices. Journal of Theoretical and Applied Electronic Commerce Research 1(2), 1–11 (2006)
11. Tseng, Y., Jan, J., Chien, H.: Digital signature with message recovery using self-certified public keys and its variants. Applied Mathematics and Computation 136(2-3), 203–214 (2003)
12. Wang, H., Kranakis, E.: Secure Wireless Payment Protocol. In: International Conference on Wireless Networks, pp. 576–578 (2003)
13. Zhang, J., Zou, W., Chen, D., Wang, Y.: On the Security of a Digital Signature with Message Recovery using Self-certified Public Key. Soft Computing in Multimedia Processing, Special Issue of the Informatica Journal 29(3), 343–346 (2005)

Computer Communications 31 (2008) 2478–2484

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks

Jesus Tellez Isaac [a,*], Jose Sierra Camara [b], Sherali Zeadally [c], Joaquin Torres Marquez [b]

[a] Computer Science Department (FACYT), Universidad de Carabobo, Avda. Universidad, Sector Bárbula, Valencia, Venezuela
[b] Computer Science Department, Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911 Leganés (Madrid), Spain
[c] Department of Computer Science and Information Technology, University of the District of Columbia, Washington DC 2008, USA

ARTICLE INFO

*Article history:*
Received 10 March 2008
Accepted 11 March 2008
Available online 22 March 2008

*Keywords:*
Ad hoc networks
Vehicular networking
Performance
Protocol
Security
Transaction

ABSTRACT

Advances in *vehicular ad hoc networks* (*VANETs*) have triggered the development of many new attractive applications such as payment services which require the design of payment systems that satisfy additional requirements associated with VANETs. The wide range of scenarios (with or without connectivity restriction) arising from vehicle-to-vehicle and vehicle-to-roadside communications have opened up new security challenges which must be considered by payment systems designers to achieve the same security capabilities independently of the scenario where the payment occurs. We propose a payment protocol aimed at those scenarios where the client cannot communicate directly with the issuer (client's financial institution) for authentication purpose. The proposed protocol uses a non-traditional digital signature scheme that reduces the communication cost (compared to certificate-based signature schemes), and increases the efficiency of the payment process (due to the low communications costs involved). Our protocol supports both credit-card and debit-card transactions, and protects the real identity of clients during the payment and can be used by any portable device.



## 1. Introduction

In the last decade, we have seen the emergence of different types of communication technologies that can provide network connectivity to mobile users. Common examples include *infrastructure-based networks* (which includes cellular networks and IEEE 802.11 [15] (WiFi) networks) and *mobile ad hoc networks*. While infrastructure-based networks depend on the existence of an infrastructure (point of access), mobile ad hoc networks do not require such previous deployment, and hence promise to be more flexible and easily deployable [21].

A new type of ad hoc network is emerging, in which vehicles constitute the mobile nodes in the network. This type of network, called *vehicular ad hoc network* (*VANET*), is a form of *mobile ad-hoc network* (*MANETs*) that aims to provide communications among nearby vehicles (also known as *inter-vehicle communication* -(***IVC***)-) and between vehicles and nearby roadside base-station (also referred to as *vehicle-to-roadside communication* -(***VRC***)-). Different communication technologies including various versions of the IEEE 802.11 standards, the 802.20 Mobile Broadband for Wireless Access standard, and the 802.16e Mobile WiMAX standard have been proposed to enable this new generation of sentient vehicles [7].

The application space for *vehicle-to-vehicle* and *vehicle-to-roadside* communications is vast and opens tremendous business opportunities and research challenges with *security* as an important one. Thus, VANETs are envisioned to support the development of a wide of new attractive applications that can be divided into two major categories [19,20,25]:

(1) *Safety-related applications:* Applications in this category share a common characteristic: the relevance to life-critical situations where the existence or lack of a service may affect life-endangering accidents. Hence, the security of this category is mandatory. Example applications in this category include: collision avoidance, cooperative driving, traffic optimization, lane-changing assistance, traffic signs violations warning, and road conditions warnings.
Applications in this category usually require direct vehicle-to-vehicle communication due to stringent delay requirements.
(2) *Comfort applications:* This type of application improves passenger comfort and traffic efficiency and/or optimizes the route to a given destination. Examples of applications in the category include: payment services (e.g., toll collection, parking lot payment), location-based services (e.g., finding the closest fuel station) and infotainment (e.g., Internet Access, music download). Some of these applications will

* Corresponding author.
*E-mail addresses:* jtellez@uc.edu.ve (J.T. Isaac), sierra@inf.uc3m.es (J.S. Camara), szeadally@udc.edu (S. Zeadally), jtmarque@inf.uc3m.es (J.T. Marquez).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.03.012

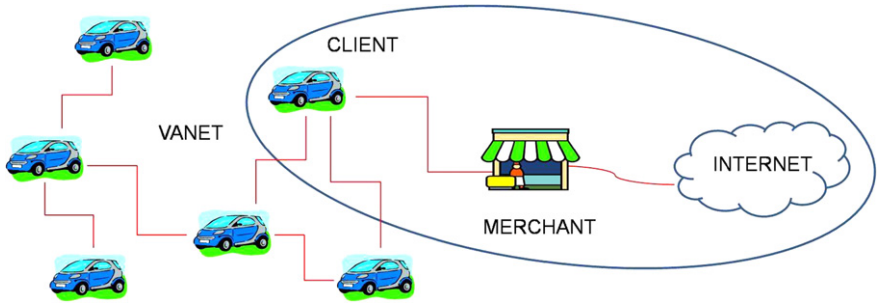


**Fig. 1.** A restricted connectivity scenario in VANETs.

be free, while others would require a service subscription or a one-time payment [8]. Moreover, security is also required in this application category, especially in the case of payment services.

The industry and the academia have concentrated their research efforts in the safety-related applications because this is an important area of the automotive domain. Nevertheless, since the comfort applications offer great business opportunities, it is expected that research in this area will continue to attract attention of researchers and designers to develop a wide range of non-safety applications.

To enable payments in VANETs, it is necessary to build payment systems that satisfy the additional requirements associated with vehicular ad hoc networks. As mentioned previously, both vehicle-to-vehicle and vehicle-to-roadside communications open up new security challenges that must be considered by payment system designers to achieve the same security capabilities independently of the scenario where payments occur.

For example, the area denoted by a circle in Fig. 1 shows a restricted connectivity scenario where a car (henceforth referred to as the client), with an on-board unit (OBU) and an application unit (AU) can only connect to the merchant during a payment transaction due to the lack of Internet access with its AU. This situation creates a security problem because the client cannot send any kind of messages directly to the issuer and has to do it through the merchant (who should not be able to change the content of the messages but must keep evidence of the payment). Note that in our scheme, the merchant takes an active role in the payment process because it acts as a proxy to allow the communication between the client and the issuer.

In the real world, the above situation can be represented by the following example: a client is on the road and stops at a gas station to purchase gas or some others goods at the gas station store. In both cases, if the client is not able to communicate with the issuer (to authorize the payment) from the application unit (due to the absence of the infrastructure necessary such road-side units (**RSUs**), or a Hotspot **HS**-), the client should still be able to perform the payment using the merchant's infrastructure.

In the context of the electronic payment systems, digital signature can be re-presented as a secure base because it provides authentication, data integrity, and non-reputation cryptography services [17,10]. However, the traditional digital signature schemes are based on asymmetric techniques which make signature computations very expensive and make then not suitable for portable devices (such as the ones typically attached to an OBU [5]). Moreover, these schemes suffer from the well-known authentication problem[1] which requires the usage of certificates to avoid it [18]. The public-key certificate must be verified by a Certificate Authority (CA), and that verification causes an additional information exchange during a transaction.

In this work, we propose a protocol for the case where we do not have direct communication between the client and the issuer. As a result, many of the schemes discussed above are not suitable because the client has connectivity restrictions, and consequently, communication with other parties (such a Certification Authority, for verifying a certificate) is not possible during a payment transaction. Therefore, the use of a non-traditional digital signature scheme is required to satisfy the requirements of the protocol proposed in this research. Digital signature with message recovery using self-certified public keys [6,23], provides an authenticated encryption scheme that integrates the mechanisms of signature and encryption, which enable only the specified receiver to verify and recover the original message. The authentication of the public key can implicitly be accomplished with the signature verification.

To address the issue of direct communication between the client and the issuer for authentication purposes, we design a payment protocol that allows the client to send a message to the issuer through a merchant (who will not be able to decrypt this message). The proposed protocol, called Kiosk centric model payment protocol for VANETs (henceforth referred to as the KCM-VAN Protocol), employs the authentication encryption scheme proposed by [26] that only allows specified receivers to verify and recover the message. This means that, no other receiver will be able to access the information. Moreover, it supports both credit-card and debit-card transactions, and protects the real identity of the clients during the payment and can be used with a AU (including portable device). Our proposed secure payment protocol supports the following security requirements [3,24,16]:

(1) User anonymity: During the payment process, neither the merchant nor the payment gateway need to know the client's real identity.
(2) Non-repudiation: Parties should not be able to claim that the transaction on their behalf was made without their knowledge. Thus, the merchant prevents the possibility that a client denies to have made a purchase. At the same time, the client puts in place mechanisms to correct account errors or security breaches.
(3) Integrity: The transaction data has to remain intact during transmission and cannot be altered by outside parties without the modification being noticeable.
(4) Authentication: Any role involved in a transaction (such as merchant, client, etc.) should not be impersonated by malicious attackers to damage fully or partially the payment scheme.

The rest of the paper is organized as follows. Section 2 introduces and proposes a payment protocol for vehicle-to-roadside scenarios in VANETs. We analyze the security and performance

[1] An imposter may impersonate any innocent user with a valid but incorrect public key (because the key does not belong to the innocent user [6]).

of the proposed protocol in Section 3. Finally, conclusions are given in Section 4.

## 2. Proposed secure payment protocol for vehicle-to-roadside scenarios in VANETs

### 2.1. The Kiosk centric model payment protocol for VANETs (KCM-VAN) Model

Our proposed secure payment protocol uses the following entities:

(1) Client: a user who wants to purchase goods or services from the merchant. In our proposed protocol, the client is a user-side entity equipped with an On-Board Unit (called **OBU**) and/or an Application Unit (called **AU** that may use the OBU's communication capabilities). An AU can be an integrated part of a vehicle and be permanently connected to an OBU or could be a portable device such as a Personal Digital Assistant (PDA), a mobile phone or a gaming device that can dynamically attach to and detach from an OBU [5].
(2) Merchant: an entity that has products or services to offer or sell. This entity could be a computational one (such as a normal web server, a roadside computing station or an intelligent vending machine) or a physical one (such as a gas station that makes it possible to pay from within an AU) which the user can connect to using a short range link (using wireless technologies such as Wi-fi or Bluetooth). Moreover, this entity connects with the Payment Gateway (an entity which provides the necessary infrastructure to allow a merchant to accept credit card and other forms of electronic payment) through a secure channel allowing the client to communicate with the issuer using this connection.
(3) Acquirer: is the merchant's financial institution. It verifies the validity of the deposited payment instrument and manages the merchant's account including fund transfer.
(4) Issuer: is the client's financial institution. It provides electronic payment instruments to the client to use in a payment and manages the client's account including fund transfer.
(5) Payment gateway: an additional entity that acts as a medium between acquirer/issuer at banking private network side and client/merchant at the Internet side for clearing purpose [16].

The five entities in KCM-VAN and their interactions are shown in Fig. 2. Note that there is no direct interaction between the client and the issuer. Moreover, the connection between the client and the merchant (denoted as the dotted arrow) is set up through a wireless channel (such as IEEE 802.15.1, Bluetooth, IEEE 802.11a/b/g).

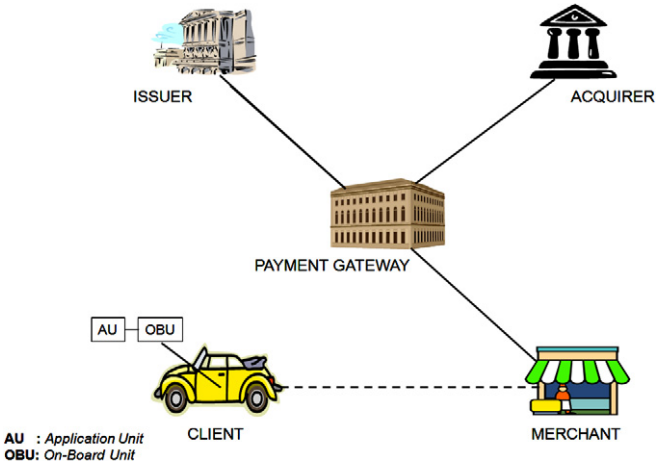


**Fig. 2.** Operational model for KCM-VAN payment protocol.

Interaction between the merchant and the payment gateway (depicted as solid arrow in the scheme) should be reliable and secure against passive and active attacks. Therefore, the connection should be established through a secure wired channel using the well-known security protocol such as secure socket layer/transport layer security (SSL/TLS). Note that the issuer, the acquirer and the payment gateway operate under the banking private network, so the security of the messages exchanged among them is beyond of the scope of this paper.

Before receiving payment services, the client must register with an Issuer. Generally, this registration can be done either personally at the issuer's premises or via the issuer's website. During the client's registration, the following steps are performed:

(1) The Client shares his/her credit- and/or debit-card information (CDCI) with the issuer (who will not reveal it to any merchant).
(2) The Issuer assigns several nicknames to the client. Those nicknames are known only to the client and the issuer [12].
(3) The client (C) sends (with the assistance of the issuer) her nicknames and $x_C$ (a value computed by the client from his/her private key and used by SA to create client's public key) to system authority (SA) who in turn sends all system parameters to **C**. Thus, the client holds $C_S$ (client's private key), $ID_I$ (issuer's id), system parameters and $I_P$ (issuer's public key) in her Application Unit (AU).

### 2.2. Notations

All the entities involved in our protocol are called parties and communicate through wireless and wired networks.

The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer, respectively. The following symbols are used to represent other messages and protocols:

- $\mathbf{ID_P}$: The identity of party $P$ that contains the contact information of $P$
- $\mathbf{NID_C}$: Client's nickname, temporary identity
- $\mathbf{K_P}$: Party's $K$ public key
- $\mathbf{K_S}$: Party's $K$ private key
- **TID:** Identity of transaction that includes time and date of the transaction
- $\mathbf{E_{P_i-P_j}(M)}$: Message $M$ signed and encrypted by the user $ID_{P_i}$ to a specified & receiver $ID_{P_j}$
- $\mathbf{TST_P}$: Timestamp generated by P
- **Stt:** The status of transaction (Stt = {Accepted, Rejected})
- **OI:** Order information (OI = {TID, OD, $h$(OD,Price)}) where OD and Price are order descriptions and its amount
- **TC:** The type of card used in the purchase process (TC = {Credit, Debit})
- **TIDReq:** The request for TID
- **MIDReq:** The request for $ID_M$
- **MPReq:** The request for $M_P$
- h($M$): The one-way hash function of the message $M$
- **A** → **B**: A sending message x to B

The signature and encryption of a message $M$ by a sender $P_i$ (with identity $ID_{P_i}$) to a specified receiver $P_j$ (with identity $ID_{P_j}$),

is generated as follows (using the procedure proposed by [26]): First, $P_i$ chooses a random number $y$ and computes:

$$r_1 = M \cdot \left(x_j^{h(ID_{P_j})} + ID_{P_j}\right)^{-yh(M)} \bmod N \quad (1)$$

$$r_2 = M \cdot \left(x_j^{h(ID_{P_j})} + ID_{P_j}\right)^{-yr_1} \bmod N \quad (2)$$

$$s = r_1 \cdot y - K_{S_i} \cdot h(r_2) \quad (3)$$

Afterwards, $P_i$ sends the triple $(r_1, r_2, s)$ as the signature of message $M$ (henceforth referred to as $E_{P_i-P_j}(M)$) to the verifier $P_j$. After receiving $(r_1, r_2, s)$, the verifier $P_j$ recovers message $M$ and verifies that the signature is valid using the following procedure (as described in [26]):

$$M = r_2 \cdot \left(g^s \cdot \left(x_j^{h(ID_{P_i})} + ID_{P_i}\right)^{-h(r_2)}\right)^{k_{S_j}} \bmod N \quad (4)$$

Then, the verifier $P_j$ further verifies the validity of the signature, computing: $(r_1 \cdot M^{-1})^{r_1} \bmod N = (r_2 \cdot M^{-1})^{h(M)}$.

### 2.3. System initialization and user registration

In our scheme, a system authority (SA) is responsible for generating system parameters [26,23] (such as $p$, $q$, $p'$, $q'$, $h(\cdot)$, $N$ and $g$). To achieve this, the SA chooses:

(1) Same size safe large primes $p$ and $q$ which satisfy $p = 2p' + 1$ and $q = 2q' + 1$.
(2) A RSA modulus $N = p \cdot q$.
(3) A generator g of the order $p' \cdot q'$.
(4) A collision-resistant hash function $h(\cdot) = 2H(\cdot) + 1$ (where $H(\cdot)$ is either SHA1 or MD5 hash functions) [11] which accepts a variant-length input string of bits and produces a fixed-length output string.

The parameters $p$, $q$, $p'$ and $q'$, are preserved privately whilst $g$, $N$, and the hash function $h(\cdot)$ are publicly known. Once the parameters have been generated, every user of the system $P_i$ (whose identity is $ID_{P_i}$) chooses a number $K_{S_i}$ for a secret key and computes $x_i = g^{K_{S_i}} \bmod N$. Then, $P_i$ sends $(x_i, ID_{P_i})$ to SA. After receiving $(x_i, ID_{P_i})$, the system authority computes and publishes the public key of $P_i$ as $K_{P_i} = (x_i - ID_{P_i})^{h(ID_{P_i})^{-1}} \bmod N$ [26].

When the user registers as the client (using a nickname instead the real identity to protect one's privacy), one $K_{P_i}$ must be generated and published for every nickname assigned to the user by the issuer.

The transmitted messages between the user and the SA during the registration process are shown in Fig. 3.

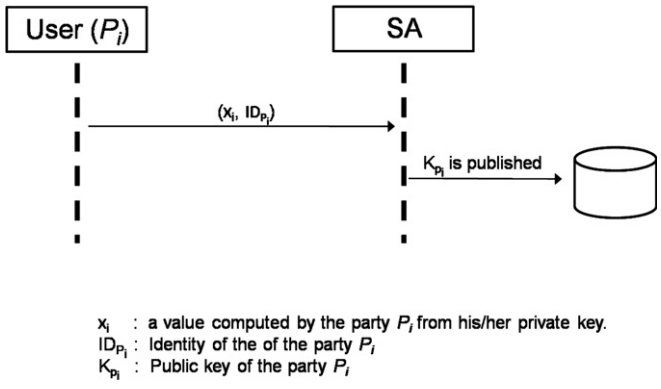


**Fig. 3.** Registration of a user.

### 2.4. The proposed Kiosk centric model payment protocol for VANETs (KCM-VAN)

Before the client can start the *KCM-VAN payment protocol*, a wallet software should be installed before the payment protocol can be executed. The software is prepared by the issuer and could be installed into the AU with the help of the vehicle manufacturer or by sending a request (through the merchant and the payment gateway) to the issuer **I**. It is worth noting that in the latter case, we have to take into account other limitations such as bandwidth, link availability, time required for the installation, etc. which may make the 'on-the-fly' installation of the software unfeasible.

The main functions of our proposed protocol are as follows:

(**1**) **C** → **M:** $NID_C, TIDReq, MIDReq, MPReq$
**M** → **C:** $E_{M-C}(TID, ID_M, M_P)$

(**2**) **C** → **M:** $E_{C-M}(OI, Price, TST_C, NID_C, ID_I, VSRequest)$
$VSRequest = E_{C-I}(Price, TST_C, h(OI), TC, ID_M)$

(**3**) **M** → **PG:** $VCRequest, ID_M$
$VCRequest = E_{M-PG}(VSRequest, TST_M, h(OI), TID, Price, NID_C, ID_I)$

(**4**) Under banking private network,
(**4.1**) **PG** → **I:** $VSRequest, h(OI), TID, Price, NID_C, ID_M$
(**4.2**) **PG** → **A:** $Price, ID_M$
(**4.3**) **I,A** → **PG:** $VSResponse, Stt, h(Stt, h(OI))$
$VSResponse = E_{I-C}(Stt, h(OI))$

(**5**) **PG** → **M:** $VCResponse$
$VCResponse = E_{PG-M}(Stt, VSResponse, h(Stt, h(OI)))$

(**6**)**M** → **C:** $PResponse$
$PResponse = E_{M-C}(VSResponse)$

**Step 1:** The client **C** and merchant **M** exchange the information necessary to start the protocol by performing the following sub-step.
1–1: **C** sends his/her nickname ($NID_C$), the request for the transaction identity ($TIDReq$), M's public key request($M_P$) and the request for the merchant identity request ($MIDReq$) to **M**.
1–2: **M** receives the request and sends back its identity ($ID_M$), $TID$ and $M_P$ to **C**, encrypted only to be recovered by the client.

**Step 2:** Client **C** creates a *Payment Request* (referred to as the General Payment Model described in [1,16]) in the following sub-steps.
2–1: A *Value-Subtraction Request* (called *VSRequest*, encrypted to be recovered only by an issuer **I**) is created and it includes $Price$, $TST_C$, $TC$, $ID_M$ and $h(OI)$.
2–2: A new message is created which includes C's nickname, I's identity, $Price$, $OI$ (used to inform **M** about the goods and prices requested), *VSRequest* and timestamp $TST_C$ read from C's clock.
2–3: The message created is encrypted by the previous sub-step (henceforth referred to as the *Payment Request*) to be recovered only by **M**.
2–4: The *Payment Request* is sent to the merchant.

**Step 3:** The merchant **M** generates the *Value-Claim Request* (called *VCRequest*) by performing the following sub-steps.
3–1: The message received from **C** is decrypted to extract $OI$ and $TST_C$.
3–2: The timeliness of the *Payment Request* is verified. If the check is successful, the following sub-steps will be performed.

3–3: The *VCRequest* is prepared, and contains the *VSRequest*, $TST_M$, $h(OI)$, identity of transaction, order's amount, C's nickname, and *I*'s identity.

3–4: The *VCRequest* (to be recovered by **PG**) is encrypted in order to ensure that only the payment gateway is the intended recipient of the message.

3–5: The encrypted message is sent in sub-steps 3–4 to the **PG** with M's identity ($ID_M$).

**Step 4:** Using private network of the banking institution, the Payment Gateway **PG** performs the following sub-steps to verify the payment.

4-1: The *VCRequest* is decrypted to retrieve *VSRequest* and the others fields.

4–2: The timeliness of *VCRequest* is verified. If the check is successful, the following steps are executed.

4–3: The *VSRequest* and other important, such as: *h(OI)*, *TID*, *Price*, $NID_C$, $ID_I$ are forwarded to **I** where it is decided whether to approve or reject the transaction.

4–4: $ID_M$ and the requested price (*Price*) are sent to confirm to the Acquirer **A** that the client is the party whom the requested amount *Price* will be transferred to.

4–5: The approved result (*Stt*) and *value-subtraction-response* (called *VSResponse*) are received from the issuer **I** and encrypted to be recovered only by **C**. It is worth noting that the *VSResponse* is prepared by the issuer after (a) checking the timeliness of *VSRequest* and the validity of the client's account, and (b) after transferring the total amount of *OI* to the merchant's account.

**Step 5:** The Payment Gateway (**PG**) generates the *Value – Claim Response* (called *VCResponse*) in the following sub-steps.

5–1: A *VCResponse* is created that includes *Stt*, *VSResponse* (which will be forwarded to **C**) and $h(Stt,h(OI))$.

5–2: The *VCResponse* is encrypted only to be recovered by **M**.

5–3: The *VCResponse* is sent to **M**.

**Step 6:** Merchant **M** performs the following sub-steps.

6–1: The *VCResponse* is decryoted to retrieve *VSResponse* and other fields.

6–2: The merchant's own *OI* is compared with the received *h(OI)*. If they not match, then the client performs Sub-step 6-3a, otherwise the client performs Sub-step 6-3b.

6–3a: A message is sent to the Payment Gateway to notify it of the response failure. The Payment Gateway then starts a recovery procedure or resends the message.

6–3b: The *Payment Response* (called *PResponse* and represents the result of the client's request) is created and includes the *VSResponse*.

6–4: The *PResponse* is encrypted to be recovered by the client **C**.

6–5: Send *PResponse* to **C**.

Once the client has finished all purchases with a merchant, the merchant's information will be removed from his/her AU because of its limited amount of storage. Fig. 4 shows the transmitted messages among the parties of the system during the execution of our proposed *KCM-VAN Payment Protocol*.

## 3. Proposed payment protocol analysis and discussions

### 3.1. Security analysis

In this section, we analyze the security of our proposed secure Payment Protocol. In the case of the KCM-VAN protocol, the client uses a nickname $NID_C$ (a temporary identity known only to the client and the issuer) instead of his/her real identity. As a result neither the merchant nor the payment gateway can map the nickname to the client's true identity. This anonymity protects relevant information from third parties but not unrestrained anonymity [2].

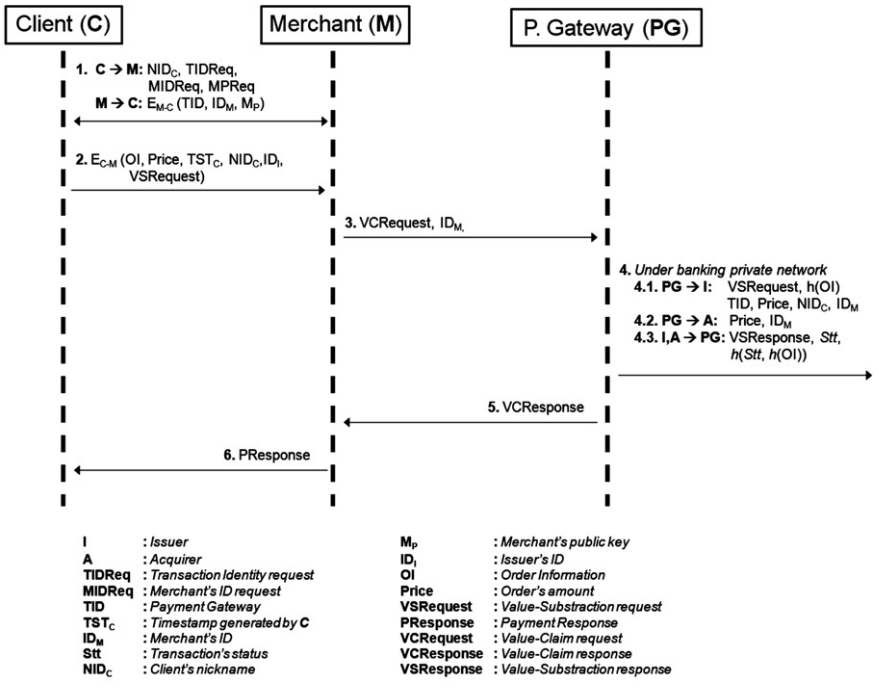


**Fig. 4.** KCM-VAN payment protocol messages flow.

In the case of our proposed protocol, a signer $U$ generates the signature of a message using his/her private key $U_S$ (known only by $U$ and which cannot be derived by an adversary E from U's signature because of the difficulty of the discrete logarithm problem [6,23]). The signer should not be able to repudiate his signature creation later. Therefore, non-repudiation of origin (NRO) is ensured.

The Confidentiality of messages transmitted in each transaction should be protected while in transit, specially in wireless networks where anybody can eavesdrop the transmitted messages easily. In our protocol, we protect those confidential messages by employing an authentication encryption scheme that enables only the specified receiver to verify and to recover the message transmitted by the sender. Any other party cannot perform such operations.

Although generally in any transaction a party should not trust others unless they can provide a proof of trustworthiness [16], in our protocol we state the trust relationship between the client and the issuer because the client has a credit- and/or debit-card issued by the issuer who will not reveal it to any party.

The KCM-VAN Protocol prevents direct communication between the client and issuer. Therefore, to allow a party (A) to authenticate another peer (B), a sender needs to transmit a message to a specific receiver through a third party with the following features:

(1) Resistance to attacks while in transit (integrity),
(2) Recoverable only by the specified receiver,
(3) Authentication assurance (being able to ensure that the message has been created and sent by the sender).

The authenticated encryption scheme used in our protocol integrates the mechanisms for signature and encryption, our proposed protocol satisfies all the requirements mentioned above and can be used by any party of the system to authenticate another peer in a secure way.

In the following paragraphs, we adopt the realistic analysis method proposed by [4] to discuss two possible attacks against our proposed protocol.

- **Replaying attack:** If an intruder $E$ wants to impersonate a legal user by replaying the user's transmitting contents, the timestamp included in the transmitted message ensure the freshness of the message and avoids replaying attacks. Thus, our proposed protocol is secure against replaying attack.
- **Impersonating attack:** An intruder $E$ tried to impersonate a user $U_i$ to SA, which results in SA being cheated. Since $E$ does not have its own valid public key $K_{P_i}$ to pass the authentication as a valid system user, the intruder has to forge a public key but for this to be possible, the intruder must know $p'$ and $q'$ (only known by SA) to compute $h(ID_{P_i})^{-1}$. This is hard to achieve because of the well-known factoring problem. As a result, impersonating attacks fail with our protocol.

### 3.2. Performance analysis

#### 3.2.1. Storage cost

Storage limitations is an important issue in an application unit (AU) which should be considered in any protocol for vehicular ad hoc networks. We focus only in the client's storage requirements for our proposed protocol because the client is the only party in our scheme who uses an AU (which could be a portable device with storage limitations) that interacts with the system.

The total number of bytes required by all the values/parameters that will be stored into the AU are summarized in the Table 1. It should be noted that we need to add 71000 bytes required for the wallet software. Thus the storage's cost will be 71384 bytes (70 kbytes). Given the low storage requirements, it is possible for any AU to store the va-lues/parameters and the wallet software required by our proposed protocol.

The performance of our proposed secure protocol is discussed in terms of *computation cost* and *Communication cost* which are crucial parameters interests in mobile systems.

Computation cost: Generally, the merchant, the *payment gateway*, the *issuer* and the *acquirer* have enough powerful computational resources to execute several modular multiplications; hence, we only take into consideration the client's computational capability.

The client needs to generate the signature of the message being sent and authenticates and recovers a message received during execution of the proposed *payment protocol*. We focus below on the time complexity required by the client to perform the above mentioned procedures.

In Table 2, the notations used for analysing the performance of the proposed protocol are presented. Note that the time for addition or subtraction with or without modulo $N$ is relatively small compared to those of $T_m$, $T_{mm}$ and $T_{exp}$, and have been ignored in the analysis of the time complexities of the our proposed scheme.

To continue with the performance evaluation, we need to discuss the length of system parameters. For practical and security considerations, we adopt the length of the system parameters suggested by Girault [9] as follows: (a) the system authority's secret parameters $p$ and $q$ should be with greater than 45 bytes against the exhaustive search attack, (b) each user $U_i$ chooses a secret key $U_{s_i}$ with 20 bytes and (c) $|h|$ is bounded to 16 bytes [22].

Adopting the time complexity analysis presented in [23], we can conclude that in our proposed protocol a *signer* requires $3T_{exp} + 3T_h + 2T_{mm} + 3T_m$ to generate a digital signature whilst the time needed by a *specified receiver* to authenticate and recover the message received from the signer is $4T_{exp} + 2T_h + 2T_{mm}$. These computation results are reasonable in that they can be computed multiple times on an Application Unit with a moderate consumption of power and memory.

Communication cost: Due to the mobile environment of VANET, messages transmitted across the wireless network require reliable transmission over the network. In our proposed protocol, we focus

**Table 1**
Storage's cost required for the client's device

| Value/parameter to be stored | Number of bytes needed |
|---|---|
| Client's identity | 10 |
| Client's nicknames | 100 (10 bytes per nickname) |
| Client's secret key | 20 |
| Client's public key | 20 |
| Issuer's identity | 10 |
| Issuer's public key | 28 |
| RSA modulus $N$ | 90 |
| Generator g | 90 |
| Hash function $h(\cdot)$ | 16 |
| Total | 384 bytes |

**Table 2**
Notations used for performance analysis of the proposed secure protocol

| | Notation | Description |
|---|---|---|
| Communication cost | $\|N\|$ | Size of modular $N$ |
| | $\|h\|$ | Output size of a one-way hash function $h$ |
| Computation Cost | $T_h$ | Time for calculating the adopted one-way function $h$ |
| | $T_m$ | Time for multiplication without modulo $N$ |
| | $T_{mm}$ | Time for multiplication with modulo $N$ |
| | $T_{exp}$ | Time for exponentiation with modulo $N$ |

on the communication taking place between the client and the merchant.

The size of an individual signature $|r_1| + |r_2| + |s|$ is bounded by $3|N|$ bytes [13] which is reasonable for all the messages transmitted through the wireless network (compared with the computation and communication cost of other protocols like the ones presented in [14]).

## 4. Conclusions and further work

We have proposed a protocol for secure protocol for on-line payments in a vehicle-to-roadside Restricted Scenario in VANETs where the client cannot directly communicate with the issuer. As a result, messages among those parties must be done through the merchant who should not be able to change the content of the messages but keep proof of the payment.

The proposed protocol employs a digital signature scheme with message recovery using self-certified public keys which allow clients to make purchases without disclosing private information and using a short range link (such as that provided by Bluetooth or Wi-fi) to communicate with the merchant. Moreover, the chosen digital signature scheme reduces the communication costs and increases the efficiency of the payment process.

Although our proposed protocol was designed for a payment system based on a restrictive connectivity scenario, the security properties are preserved for a scenario with full connectivity among the different entities. Moreover, our proposed secure protocol can withstand replay and impersonating attacks. We also argue that our proposal is efficient and can be deployed for those practical scenarios of restricted connectivity in VANETs.

In the future, we will explore the possibility to reformulate the proposed protocol to satisfy the requirements of other restrictive connectivity scenarios in VANETs (such as a scenario where the merchant cannot communicate directly with the acquirer).

## Acknowledgements

This work was supported in part by ASPECTS-M Project (reference models for secure architectures in mobile electronic payments-CICYT-2004), but all ideas expressed represent the view of the authors. We also thank the anonymous reviewers for their constructive suggestions and remarks which helped to improve the quality and presentation of this paper.

## References

[1] J. Abad-Peiro, N. Asokan, M. Steiner, M. Waidner, Designing a generic payment service, IBM Systems Journal 37 (1) (1998) 72–88.
[2] N. Asokan, Anonymity in a mobile computing environment, in: Proceedings of the Workshop on Mobile Computing Systems and Applications, 1994, pp. 200–204.
[3] N. Asokan, P.A. Janson, M. Steiner, M. Waidner, The state of the art in electronic payment systems, Computer 30 (1997) 28–35.
[4] G. Bella, S. Bistarelli, Information assurance for security protocols, Computers and Security 24 (4) (2005) 322–333.
[5] Car2Car Communication Consortium. Overview of the C2C–CC System. Technical Report version 1.0, Car2Car Communication Consortium, 2007.
[6] Y. Chang, C. Chang, H. Huang, Digital signature with message recovery using self-certified public keys without trustworthy system authority, Applied Mathematics and Computation 161 (1) (2005) 211–227.
[7] D. Cottingham, I. Wassell, R. Harle, Performance of IEEE 802.11a in vehicular contexts, in: Vehicular Technology Conference, 2007, pp. 854–858.
[8] F. Dötzer, Privacy issues in vehicular ad hoc networks, in: 5th International Workshop on Privacy Enhancing Technologies, 2005, pp. 197–209.
[9] M. Girault, Self-certified public keys, in: Proceedings of EUROCRYPT'91, 1991, pp. 491–497.
[10] L. Gollan, C. Meinel, Digital signatures for automobiles? in: Wireless and Optical Communications (WOC 2002), 2002.
[11] L. Hernández Encinas, A. Martín del Rey, J. Muñoz Masqué, A weakness in authenticated encryption schemes based on Tseng et al.'s schemes, International Journal of Network Security 7 (2) (2008) 185–187.
[12] Z. Hu, Y. Liu, X. Hu, J. Li, Anonymous micropayments authentication (AMA) in mobile data network, in: 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM), 2004, pp. 46–53.
[13] S.-J. Hwang, Y.-H. Lee, Repairing ElGamal-like multi-signature schemes using self-certified public keys, Applied Mathematics and Computation 156 (1) (2004) 73–83.
[14] R.-J. Hwang, F.-F. Su, A new efficient authentication protocol for mobile networks, Computer Standard and Interfaces 28 (2) (2005) 241–252.
[15] IEEE Computer Society LAN MAN Standards Committee, wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std 802.11-1999. The Institute of Electrical and Electronics Engineers, 1999.
[16] S. Kungpisdan, A secure account-based mobile payment system protocol, in: International Conference on Information Technology: Coding and Computing (ITCC), 2004, pp. 35–39.
[17] Y. Lei, D. Chen, Z. Jiang, Generating digital signatures on mobile devices, in: 18th International Conference on Advanced Information Networking and Applications (AINA 2004), 2004, pp. 532–535.
[18] A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.
[19] P. Papadimitratos, A. Kung, J.-P. Hubaux, F. Kargl, Privacy and identity management for vehicular communication systems: a position paper, in: Workshop on Standards for Privacy in User-centric Identity Management, 2006.
[20] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: 3rd ACM Workshop on Security of ad hoc and Sensor Networks (SASN'05), 2005, pp. 11–21.
[21] A. Saha, D. Johnson, Modeling mobility for vehicular ad-hoc networks, in: 1st ACM International Workshop on Vehicular ad hoc Networks, 2004, pp. 91–92.
[22] W. Stallings, Cryptography and Network Security: Principles and Practice, second ed., Prentice Hall, New Jersey, 1999.
[23] Y. Tseng, J. Jan, H. Chien, Digital signature with message recovery using self-certified public keys and its variants, Applied Mathematics and Computation 136 (2–3) (2003) 203–214.
[24] U. Varshney, Mobile payments, Computer 35 (2002) 120–121.
[25] S. Yousefi, M. Mousavi, M. Fathy, Vehicular ad hoc networks (VANETs): challenges and perspectives, in: 6th International Conference on ITS Telecommunications, 2006, pp. 761–766.
[26] J. Zhang, W. Zou, D. Chen, Y. Wang, On the security of a digital signature with message recovery using self-certified public key, Informatica (Slovenia) 29 (3) (2005) 343–346.

Published in IET Communications
Received on 16th March 2009
Revised on 10th June 2009
doi: 10.1049/iet-com.2009.0191

In Special Issue on Vehicular Ad Hoc and Sensor Networks

ISSN 1751-8628

# Security attacks and solutions for vehicular *ad hoc* networks

*J.T. Isaac*[1] *S. Zeadally*[2] *J.S. Cámara*[3]

[1]*Computer Science Deparment (FACYT), Universidad de Carabobo, Avenida Universidad, Sector Bárbula, Valencia, Venezuela*
[2]*Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC 2008, USA*
[3]*Computer Science Department, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, 28911 Leganés, Madrid, Spain*
*E-mail: szeadally@udc.edu*

**Abstract:** Vehicular *ad hoc* networks (VANETs) have attracted a lot of attention over the last few years. They have become a fundamental component of many intelligent transportation systems and VANETs are being used to improve road safety and enable a wide variety of value-added services. Many forms of attacks against VANETs have emerged recently that attempt to compromise the security of such networks. Such security attacks on VANETs may lead to catastrophic results such as the loss of lives or loss of revenue for those value-added services. Therefore making VANETs secure has become a key objective for VANET designers. To develop and deploy secure VANET infrastructures remains a significant challenge. The authors discuss some of the main security threats and attacks that can be exploited in VANETs and present the corresponding security solutions that can be implemented to thwart those attacks.

## 1 Introduction

Nowadays, transportation systems play an important part in our daily activities. However, recently, we have also witnessed several deficiencies as well as inefficiencies in many of our transportation systems that have led to the loss of lives, money and time. There are on-going efforts to develop and improve transportation systems to make them more intelligent and such systems are often referred as intelligent transportation systems (ITSs).

One transportation system that has recently attracted a lot of attention from both academia and industry is vehicular *ad hoc* networks (VANETs). They constitute the basis of ITS. They enable vehicles to actively communicate with each other and to better perceive the traffic situation (such as accidents and traffic jams) in their vicinity. VANETs enable vehicles to avoid problems either by taking desired actions or by alerting other drivers. Besides road safety enhancements, the advent of VANET also opens up opportunities for many VANET-related applications (such as internet access from a car) that have great potential in enhancing our travelling comfort. VANETs do not depend on fixed infrastructures and their nodes (known as mobile nodes) and may form networks on the fly for a variety of environments. Owing to the nature of these kinds of mobile nodes, VANETs can be challenged with frequent topology changes as well as physical threats (that create potential vulnerabilities for potential attackers). A successful attack on VANETs can have catastrophic results (such as the loss of lives) or may lead to financial losses (for payment services). Therefore securing VANETs is crucial to the design, implementation and operation of these networks.

In this work, we discuss some of the major security attacks that have been reported on VANETs recently. In Section 2, we present the corresponding security solutions that have been proposed to thwart those security attacks and vulnerabilities. The main security areas we focus on in this section include: anonymity, key management, privacy, reputation and location. Section 3 presents VANET security challenges that still need to be addressed by the research community. Finally, we summarise the major results and make some concluding remarks in Section 4.



doi: 10.1049/iet-com.2009.0191

## 2 Security attacks and solutions in VANETs

### 2.1 Anonymity

Anonymity is a critical concern in VANETs and aims to hide the physical identity of a node (typically a vehicle).

*2.1.1 Malicious Vehicle:* One of the most important security requirements of VANETs is privacy. To avoid being tracked, the use of randomly changing identities (also called pseudonym) is suggested. This can lead to a situation where a malicious vehicle $M$ can easily change its identity to node N without being punished. To isolate a malicious vehicle from exploiting the use of pseudonyms in VANETs, Liu *et al.* [1] proposed a probabilistic method which uses Bloom filters to record both dishonest and trusted nodes. Moreover, with this approach, the authors assume that each car has tamper-proof device (TPD) carrying out secure operations.

In Liu's proposed scheme, each vehicle periodically broadcasts feedback messages to neighbours within its transmission range. At the same time, every feedback message received is parsed and used to update its credit (a value that represents an honesty level) by TPD. The proposed scheme works as follows:

- Feedback structure: Two kinds of feedback messages exist in Liu's scheme: positive feedback which indicates that the identities included in the message are honest whereas negative feedback means the opposite. However, considering the communication overhead, the identities need to be compressed and the authors suggest the use of Bloom filters (Fig. 1) to represent $n$ elements to support membership queries. The basic idea is to allocate an $m$-bit vector $v$, with each bit set to be zero at first. Then $k$-independent hash functions are chosen, $h_1(x)$, $h_2(x)$, ..., $h_k(x)$. As illustrated in Fig. 1, for each member $a$ of the element, each bit of $v$ is set according to the results of $h_1(a)$, $h_2(a)$, ..., $h_k(a)$. When we check whether element $b$ is a claimed member of the filter, each bit of $v$ is compared with the results of $h_1(b)$, $h_2(b)$, ..., $h_k(b)$. As long as at least one bit (supposed to be 1) of $v$ is 0, it can be asserted that $b$ is not a member. However, if an element is accepted as a member, it may not actually be a real member (called false positive and represents an erroneous answers to queries). Fortunately by increasing the number of $k$, the allowable fraction of errors can be controlled below some level.

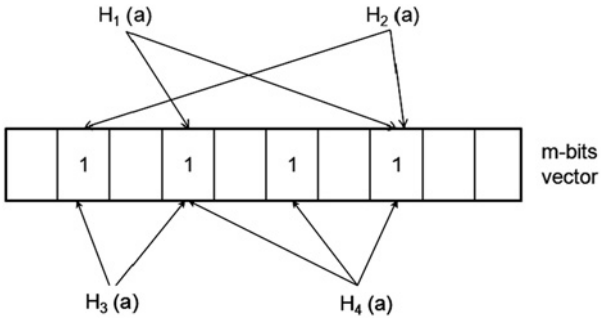


**Figure 1** *A bloom filter with four hash functions $H_i(a)$ [1]*

Once the member vector has been calculated, the vehicle should hand the data combined with the feedback type over to the TPD which attaches a timestamp and its own credit (which represents the value defined in the proposal that refers to the honest level of a car) to the vector and then signs it.

- Feedback broadcast: With this scheme, each vehicle is required to maintain two vectors representing the positive and negative filters. Once a vehicle has been detected or trusted, the filters are then updated according to the Bloom filters. Note that it is suggested that the two vectors should be reset after a certain period of time to take into account the existence of malfunctioning nodes. The TPD should guarantee that the feedback broadcasting frequency remains below a certain value to avoid being abused. Otherwise the TPD should decline requests or adds them to a waiting list.

- Credit update: After a feedback is received, each vehicle individually hands it over to its TPD which uses it to update its securely stored credit value. Vehicle can maliciously create false negative feedbacks to make other nodes get bad credits, or may even cooperate to send false positive feedbacks to benefit each other, the credit is securely evaluated by a mathematical model used in Liu *et al.*'s scheme. The credit calculated is attached to each message signed by the TPD, which can be used to determine if the message can be trusted.

### 2.2 Key management

Key management deals with the secure generation, distribution and storage of keys. For *ad hoc* networks, the current literature reports three main approaches for key management: key exchange, key agreement and key management infrastructure [2].

*2.2.1 Brute force attack:* VANETs extend the familiar concept of a computer network to vehicles travelling on roads. Typically, VANETs consist of several key components, the most important of which is road-side units (RSUs) positioned on the sides of roads and on-board units (OBUs) which vehicles are equipped with. The distribution of safety-related information (such as turn warnings, speed limit information and so on) is a major application of VANET. Since safety information may contribute to the survival of people driving the vehicles participating a VANET, security is of crucial importance to the system.

A simple solution to secure VANETs is the appropriate application of cryptographic algorithms and approaches that are already widely deployed to protect against traditional threats in computer networks. Since each RSU and OBU is a

potential node on the network, the number of keys used on such a network can be very large and the efficient management of the keys utilised by these cryptographic algorithms is a significant security challenge that must be addressed.

One key distribution solution was proposed by the vehicle safety communications consortium [3]. However, this proposal (called Straw Man) has one major deficiency because it does not provide a way to cryptographically differentiate OBUs and RSUs. This allows the possibility for an attacker to compromise the RSU by masquerading on the network as an RSU.

The above deficiency was then addressed by Langley *et al.* [4] who proposed a more secure method of authentication which requires the use of some unique identification for vehicles (required to determine if a given device has been authorised to use the VANET). However, the usage of a unique identifier is a potential violation of the privacy that the system must maintain for users of VANETs.

Since vehicles on the VANET do require privacy, it is important to devise a method to authenticate these vehicles while maintaining their anonymity. Vehicle identification number (VIN) can be used to identify vehicles in VANETs; they are by construction unique to each vehicle. However, if VIN were used for authentication, an attacker could eavesdrop on the network and determines the identity of a vehicle as it attempts to prove its identity to key distribution mechanisms when joining the network. Thus, VIN is not a usable, secure solution.

To deal with the VIN shortcoming, Langley *et al.*'s proposal generates some large random value and concatenate this value to the vehicle's VIN. The resulting value is then hashed using some hash algorithm (for example, SHA-1). The hashed value is then used as the vehicle's unique identifier. This value, by construction, is unique because of the properties of secure hash functions and the fact that VIN is unique to every vehicle. Furthermore, the appending of a random value provides some security against brute force attacks to determine the vehicle's identity. For this type of attack, VINs can be constructed and continually hashed until a match is found with the unique identifier used by the target vehicle. Since the hash input is concatenated with a large random value, the attacker must also correctly guess this random value to determine if the correct VIN was generated. This substantially decreases the probability that an attacker will be able to compromise the identity of a vehicle.

*2.2.2 Misbehaving and faulty nodes:* Vehicular network (VN) nodes (road-side infrastructure units and vehicles) can participate in network operations if each has a certificate issued by a Certification Authority (CA). Nevertheless, the possession of a certificate does not guarantee that its holder will provide correct information: a node can simply inject faulty data (e.g. alerts, warnings, coordinates) while complying with the implemented protocols.

The eviction of misbehaving nodes can be achieved by a typical approach: a revocation of a node's certificates which implies that messages from this node will be ignored after the revocation. However, the lack of an omnipresent road-side infrastructure, especially in the early deployment stages, and the large-scale deployment of VANETs prevent the application of traditional certificate revocation schemes [5]. Moreover, unless a node is revoked for administrative reasons (e.g. the vehicle owner did not renew its registration), it becomes difficult for the authority to obtain and validate sufficient evidence that a node is faulty or compromised. Thus, an additional challenge is the protection of non-misbehaving nodes until they obtain the revocation information regarding misbehaving nodes.

To address these problems, Raya *et al.* [5] proposed the combination of (i) infrastructure-based revocation protocols [the revocation of the trusted component (RTC) and revocation using compressed certificate revocation lists ($RC^2RL$)], (ii) a misbehaviour detection system enabling the neighbours of a misbehaving or faulty node to detect its deviation from normal behaviour, and initiates (iii) a local eviction of attackers by voting evaluators (LEAVE) protocol to safeguard the system operation until the attacker is revoked by the CA, partially or fully, based on the evidence LEAVE provides.

Rayas *et al.*'s scheme consists of the following basic components: (i) the centralised revocation of a node by the CA, (ii) the local detection of misbehaviour performed individually by each node and (iii) a distributed, localised protocol for the eviction of an attacker by its neighbouring nodes. The scheme, along with its basic components, are illustrated in Fig. 2.

Two methods are proposed for misbehaving node revocation initiated by the CA. The first one, RTC (Fig. 3), leverages on the presence of a TC unit on-board the vehicle. The CA determines that a vehicle $V$ must be revoked and, with the help of the road-side infrastructure, initiates a two-party end-to-end protocol with TCV, the trusted component of $V$. The CA instructs the TC to erase all cryptographic material (e.g. keys) it stores and halts TC operation upon completion of the protocol.

However, RTC is not robust against a sophisticated adversary that controls the communication link between the CA and the TC. If the CA fails while executing RTC (detected by the lack of an acknowledgment), it will revert to the distribution of the revocation information, namely, a CRL, to the VN. This way, the CA invalidates credentials before the end of their lifetime. But the size of CRLs will grow with the size of the VN and as a result, this approach is not scalable. To adapt this approach to the VN scale, Raya *et al.* [5] proposed the $RC^2RL$ protocol, with compressed CRLs (C2RLs) (using Bloom filter compression) being shorter than traditional CRLs.



*IET Commun.*, 2010, Vol. 4, Iss. 7, pp. 894–903
doi: 10.1049/iet-com.2009.0191

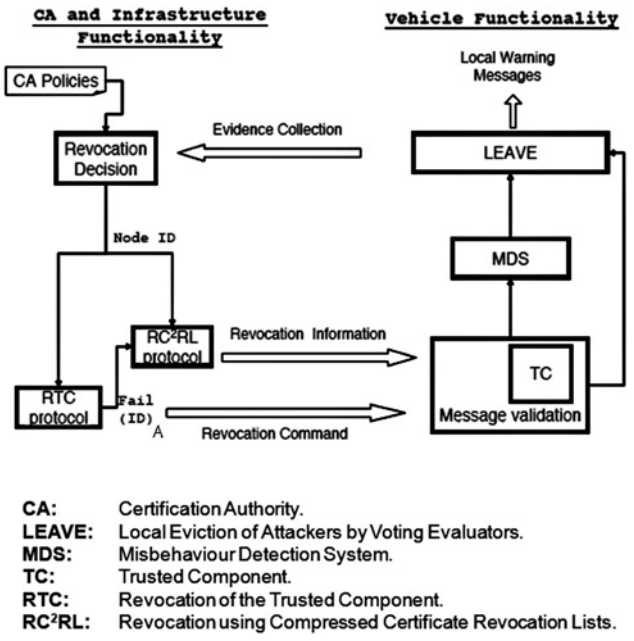


**Figure 2** *Detection and eviction scheme overview [5]*

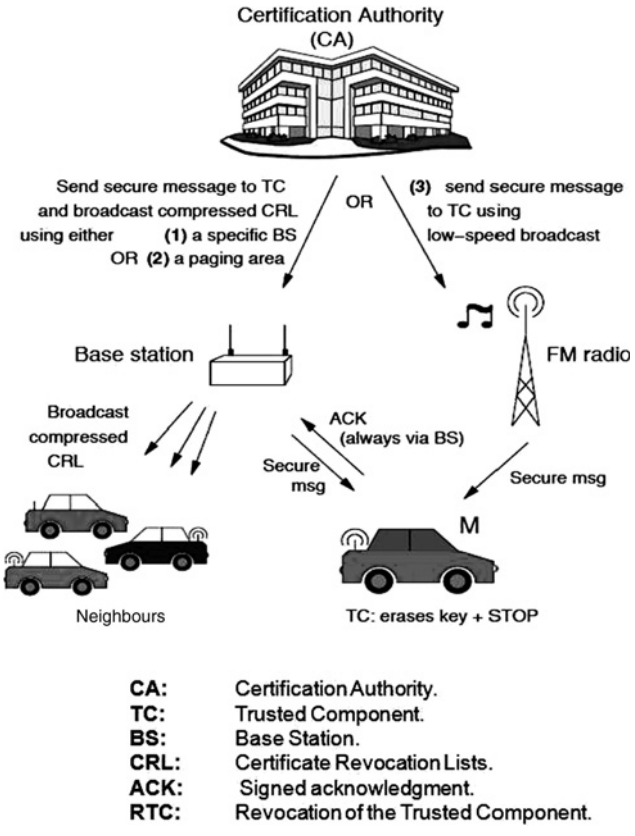


**Figure 3** *RTC and RC²RL [5]*

## 2.3 Privacy

Privacy is a key aspect in VANETs and refers to the ability of the drivers to protect sensitive information about them against unauthorised observers.

*2.3.1 Malicious user:* VNs show great promise in improving safety and enabling other value-added services. Security and privacy are two important issues in the deployment of VNs. Privacy-preserving authentication is a key technique that addresses these two issues. To address the privacy problem, Xi *et al.* [6] proposed a random key-set-based authentication protocol that preserves user privacy under the zero-trust policy, in which no central authority is trusted with the user privacy (users have to rely on OBUs to provide privacy).

Xi *et al.*'s protocol takes advantage of symmetric random key set, in which every key is shared by a set of vehicles. Thus, when the key is used for authentication, an RSU cannot uniquely identify the OBU because the key may be provided by other vehicles who share the same key. The details of the proposed protocol are as follows (as shown in Fig. 4): first, the RSU announces its service by broadcasting certificates signed by the Key Distribution Center (KDC). When a vehicle decides to access the service, it sends an authentication request together with a set of key indices which were assigned from the key pool. The set of symmetric keys (called KSet) is shared between the vehicle and the RSU. A time information $T1$ is appended for message refreshness. All such information is encrypted by a session key $S$. The session key $S$ is encrypted with the RSU's public key $Pub_S$. This encryption protects the message integrity and prevents the keys from being disclosed to an outside observer. A set of keys is used (instead of one key) for authentication purposes because there is a high probability for the OBU to have one key shared by a large number of vehicles, which makes it difficult to identify a malicious vehicle if the key is reported as invalid.

Upon receipt of the authentication request from the vehicle, the RSU creates a challenge message by encrypting

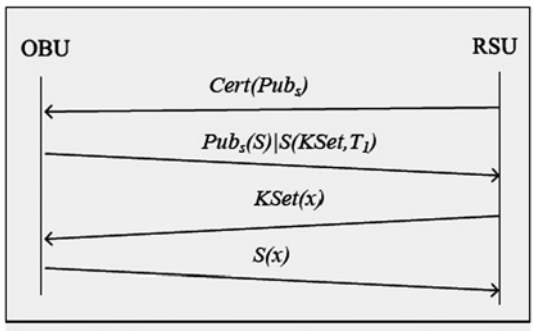


**Figure 4** *Protocol exchange diagram of privacy-preserving authentication protocol using symmetric key set [6]*

a random secret with the set of keys indicated in the request. The encryption should use cipher-block chaining mode with multiple encryption keys. The order to use the set of keys is the same as that defined in the authentication request. If the road-side server detects some invalid keys or revoked keys, the authentication will fail immediately. Otherwise, the challenge is sent back to the vehicle to verify the actual possession of those keys by the vehicle.

Once the vehicle has received the challenge message, it decrypts the message with the chosen keys and creates a response by encrypting the random secret with the session key. The response is sent back to the RSU which verifies the response by comparing the decrypted secret with its original secret. Upon successful verification, the RSU accepts the session key and the vehicle is authenticated.

The proposed protocol provides a way to link the origin of the attack to the attacker. This is due to the fact that access to the network requires authentication. Although the scheme provides some anonymity, the KDC can still narrow down the range of candidates for the attacker, especially when other details about the attacker are also provided. Generally, an attacker uses random key sets during an attack to prevent its physical trajectory from being identified. However, the attacker has only a limited number of keys. Consequently, using different keys during the attack increases the probability of exposing its unique identity.

*2.3.2 Traffic analysis attacks:* Considered as one of the serious threats to privacy in VANETs traffic analysis is a category of attacks against anonymity of communications [7]. Techniques to prevent traffic analysis attacks on the internet have been an active area of research as well. To address traffic analysis attacks, Cencioni *et al.* [8] proposed a vehicle-to-infrastructure communication privacy enforcement protocol (called VIPER) which is resilient to three kinds of traffic analysis attacks: message coding attack (if messages do not change their coding during transmission they can be linked or traced), message volume attack (the amount of transmitted data, i.e. the message length, can be observed. Thus, a global observer is able to associate a communication relation to a certain client and server) and timing attack (An opponent can observe the duration of a specific communication by linking its possible endpoints and waiting for a correlation between the creation and/or release event at each possible endpoint.). The intuition behind the VIPER protocol is to have vehicles not to send their messages directly to the RSU, but to have vehicles acting as mix nodes. The messages are encrypted via a public key crypto-system that allows reencryption (that is, a ciphertext can be encrypted again without first being decrypted, while requiring a single decryption on the recipient side) of messages. The mix is limited to nodes belonging to the same group, where a group is defined as the set of vehicles registered with a RSU. The combination of these techniques is as follows:

- Routing: When a vehicle needs to send a message to the RSU, it flips a biased coin, where faces are named: forward (with associated probability $p_f > 1/2$) and send (with associated probability $1 - p_f$). If the result of the flip is forward, then VIPER randomly selects another vehicle (henceforth referred to relay) from the same group it belongs to, and this vehicle becomes the intended recipient of the message. However, if the result of the coin flip is send, the RSU is the recipient. Fig. 5 shows an example of routing in VIPER.

To prevent the message volume attack, the batch size must be fixed: VIPER adds some dummy messages to the queue if the number of queued messages is smaller than the batch size. However, since it is possible that the number of messages to be sent could exceed the size of the queue, in these cases the authors assume that the excess messages are lost.

- Message encryption: To prevent an eavesdropper from learning the identity of the sender from the sender's field of the message, every message is encrypted. VIPER uses universal re-encryption, a public-key crypto-system based on ElGamal [9] that allows reencryption.

- Group membership notification: In VANETs, node mobility is high, and group membership can change frequently. To preserve the routing mechanism previously described, group members have to be notified about group changes. In VIPER this task is accomplished by the RSU, through the periodic dispatch of group notification messages (GNMs). That is, the RSU periodically broadcasts a message containing the identities of the vehicles currently belonging to the group (the vehicles currently registered with that RSU). To prevent an insider adversary (which is registered within the RSU and can decrypt a GNM) to learn the identities of the other group members, real identities have to be replaced with pseudonyms.

- Registration: Owing to the high mobility of the vehicles, it is possible that a vehicle (say, $V$) can leave an area served by an RSU $U$ and be into an area served by a different RSU $U'$. As a result, $V$ has to register within $U'$ to change group.

- RSU replies: As some messages require a reply from the RSU, the anonymity of $V$ has to be protected not only in

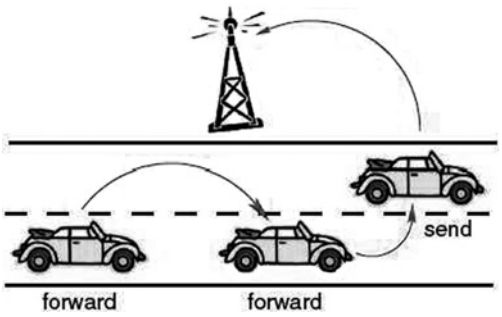


**Figure 5** *Example of routing in VIPER [8]*



*IET Commun.*, 2010, Vol. 4, Iss. 7, pp. 894–903
doi: 10.1049/iet-com.2009.0191

the request sending phase, but also in the reply transmission. To overcome this problem, Cencioni *et al.* [8] use hash message authentication code (HMAC) to allow a vehicle to determine if it is the recipient of the message without requiring message decryption. Both the RSU and $V$ share the secret value SV (they can exchange it secretly during the registration phase). Thus, every reply from the RSU to $V$ is then encrypted with the public key of $V$ and broadcasted together with a random value $r$ and the HMAC computed over the random value $r$ and the secret value $S_V$ that is, HMAC($S_V$, $r$). Upon receiving the triplet: random value, HMAC, and encrypted message (⟨$r$, hmac, enc_message⟩), the vehicle computes the value hmac! = HMAC($r$, $S_V$). If hmac! = hmac, then it is the intended recipient of the encrypted message, and can proceed to decrypt the message, otherwise it will not undertake this computationally expensive operation, since it is not the intended recipient.

The re-encryption operated by relay vehicles enables VIPER to be resilient to the message coding attack. Since the relay vehicle re-encrypts every relayed message using a secret encryption factor, the message coding changes at every relay making the tracking of the message impossible for the adversary. VIPER is resilient to the message volume attack because both the message and the batch size are fixed, while it is resilient to timing attack because of the mix function carried out by the relay vehicles. By forcing vehicles to transmit and receive messages at fixed data rates, it is also impossible for a local eavesdropper to track a message using different transmission and receiving intervals.

## 2.4 Reputation

Reputation is usually defined as the amount of trust inspired by a particular member of a community in a specific setting or domain of interest. Reputation systems are used to trust and encourage trustworthy behaviour. In VANETs, these kinds of systems can be used to defend against compromised nodes, and malicious ones.

*2.4.1 Malicious nodes:* The distribution of information about local traffic or road conditions is one of the emerging VANET applications. This can increase traffic safety and improve mobility. However, one of the main challenges is to forward event-related messages in such a way that the information can be trusted by receiving nodes.

One promising solution is the use of reputation system which can defend against compromised nodes. But conventional centralised trust establishment approaches are not suitable for use within distributed networks such as those envisioned for VANET environments. To address the above problem, Florian *et al.* [10] proposed the vehicle *ad hoc* reputation system (VARS), a completely distributed approach based on reputation. This reputation system makes use of direct and indirect trust as well as opinion piggybacking (a mechanism to append opinion about the message trustworthiness to the message by every forwarding node by which the message is distributed) to enable confidence decisions on event messages. Opinions on the trustworthiness of an event message are appended during message forwarding. Sender-based reputation levels influence these opinions.

The term direct trust is used for reputation information that is derived from experience (e.g. an announced event can be verified if recognised by a node) while indirect trust is transitive second-hand reputation provided by nodes of which reputation information is already known. Florian *et al.* [10] adjust thresholds for the confidence decision in relation to the relative position of the sender compared to the position of the deciding node. They distinguish between situations with respect to availability and quality of reputation information as well as familiarity of the area. These levels are called geo-/situation-oriented reputation levels.

*2.4.2 Illusion attack:* Illusion attack is a new security threat on VANET applications where the adversary intentionally deceives sensors on her/his own vehicle to produce wrong sensor readings [11]. As a result, the corresponding system reaction is invoked and incorrect traffic warning messages are broadcasted to neighbours, creating an illusion condition on VANET.

Lo *et al.* [11] argue that an attacker must create a virtual traffic event to produce an illusion attack. Two prerequisite conditions must be achieved by the attacker to create the virtual traffic event. The first condition for the attacker is to realise or create the prerequisite traffic situation on the road. Second, the false traffic warning messages should be generated and distributed by the attacker. The main difference between a fabricated message and an illusion attack is that the attacker in an illusion attack tries to achieve the prerequisite traffic situation first before distributing false messages.

The traditional message authentication and integrity check used in wireless networks are inadequate against the illusion attack, Lo *et al.* [11] proposed a new system architecture called the plausibility validation network (PVN) to check the raw data from sensors and further evaluate whether the incoming or generated message is valid or not.

The architecture of PVN (as shown in Fig. 6) uses two ways to obtain the application system input. One is to receive incoming messages from the wireless antenna whereas the other is to detect data reported by sensors. Input data are categorised by the data-type header. Detailed data information is stored in element fields sequentially after the type field in the header.

The PVN model is composed of a plausibility network (PN) checking module and a rule database. The PVM model works as follows: once a message is received from

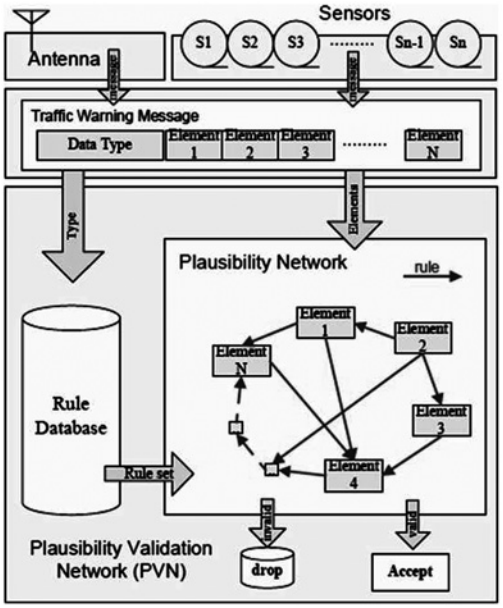


**Figure 6** *Architecture of PVN model [11]*

the antenna or generated from sensors, the system (based on the type of message) retrieves a predefined rule set (a series of validation procedures or rules to determine the value of a specific element field in a given message is reasonable or not) from the rule database for the PVN. In the PN checking module, each value in an element field of a given message is cross-verified with the values of other co-related element fields by referencing the predefined rules in rule set. Thus, a given message is trustworthy to application systems only if the verification processes of all element fields are successfully passed. A reasonable or trustworthy message is a valid message. Otherwise, the message is identified as invalid and will be dropped automatically by applications that implement the PVN model.

## 2.5 Location

Location refers to vehicle position in a vehicular *ad hoc* network (VANET). It is one of the most valuable pieces of information (used in geographic routing) and is often readily available through positioning services such as global positioning system (GPS).

*2.5.1 Forging positions and Sybil attacks:* The threat model assumed by Yan *et al.* [12] includes three types of attack: the position attack, the Sybil attack and the combination of the position attack and the Sybil attack. Position attacks can occur when the line of sight of radars is blocked. An attacker can launch a position attack by modifying position packets, replaying bogus position packets and dropping urgent position packets.

The Sybil attack is a well-known harmful attack in VANETs whereby a vehicle claims to be several vehicles either at the same time or in succession. The Sybil attack is a well-known harmful attack in VANETs whereby a vehicle claims to be several vehicles either at the same time or in succession. In addition, a Sybil attack refers to an attack where the vehicle's identity masquerades as multiple simultaneous identities. The Sybil attack is harmful to network topologies, connections, network bandwidth consumption, and there are some threats even related to human life. An example of a Sybil attack is shown in Fig. 7.

To prevent most of the position-related attacks and Sybil attacks reported until now, Yan *et al.* [12] proposed a novel solution that was motivated by the need to provide secure topology information in VANETs and to build a secure network for applications such as a congestion alert system. Based on the adage: 'Seeing of believing', the authors use on-board radar as the virtual eye of a vehicle. Although the eyesight is limited because of a modest radar transmission range, a vehicle can see surrounding vehicles and hear reports of their GPS coordinates. By comparing what is seen with what has been heard, a vehicle can corroborate the real position of neighbours and isolate malicious vehicles to achieve local security.

To prevent some variations of Sybil attacks, Yan *et al.* [12] proposed a solution where if a radar works such that it can detect the physical existence of a vehicle, this physical information can be used to improve the highly abstract information about the vehicle. The authors compute similarity among three kinds of data: radar detections, oncoming traffic reports and reports from neighbours. To average these similarities, each similarity has a weight. When radar works, radar detections are more trustworthy,

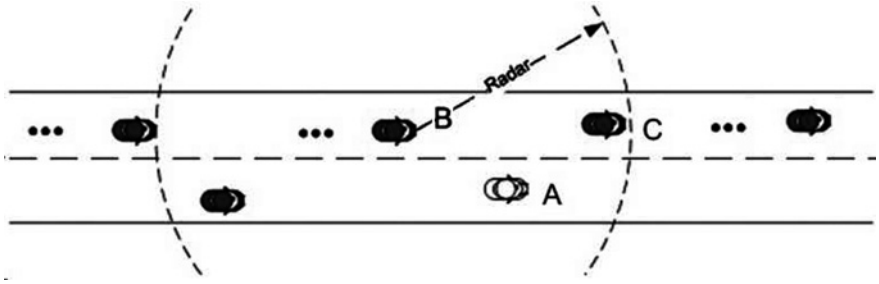


**Figure 7** *A possible Sybil attack*

*A* obtains *C*'s position $L_c$. *A* claims to victim *B* that its position is $L_c$, and that its ID is $ID_a$. *B* detects a vehicle is at $L_c$ then concludes that it is the position of *A* [12]



*IET Commun.*, 2010, Vol. 4, Iss. 7, pp. 894–903
doi: 10.1049/iet-com.2009.0191

therefore radar detections have a larger weight. When the radar does not work, reports from neighbours have a larger weight. The average position and velocity will be computed if the similarity is close. A history of the road map is maintained by storing these average positions and velocities over a period of time. When a query according to position needs to be made, vehicles rebuild the target vehicles map history virtually and make their decisions based on this map.

*2.5.2 Position cheating and false position disseminating:* VANETs have special requirements in terms of node mobility and position-dependent applications. These requirements are adequately met by geographic routing protocols. Geographic routing approaches are mostly based on the same principles: every node determines its current position by means of a positioning system such as a GPSs. The position is periodically broadcasted in beacon packets so that every node within the wireless transmission range is able to build up a table of neighbouring nodes including their positions. If a node has to forward a packet it selects the next hop from the neighbour table according to a predefined rule (e.g. it selects the node closest to the destination).

When a node disseminates wrong position data the routing process messages through the VANET is affected. Wrong position information may result from malfunction in the positioning hardware or may be falsified intentionally by attackers to reroute data. Malfunctioning nodes may degrade the performance of a system to some extent while rerouting of data through malicious nodes violates basic security goals such as confidentiality, authenticity, integrity or accountability.

Recently, Leinmüller *et al.* [13] addressed the security of geographic routing by proposing an approach based on the basic design space for position verification in VANETs (depicted in Fig. 8). Leinmüller *et al.* [13] focus on infrastructure-less autonomous position verification (subtree 1.2.1 in Fig. 13). With this approach each node judges the position claims independently of the other nodes. To perform this judgement, the approach relies entirely on position information that is transmitted in regular beacon messages, assuming that every node is able to determine its own position by using a positioning system (such as GPS or GALILEO).

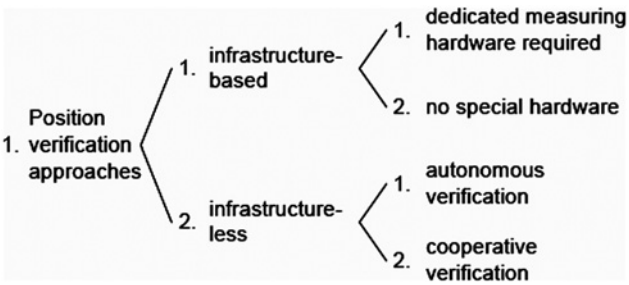


**Figure 8** *Position verification approaches design space [13]*

Although Leinmuller *et al.* have developed mechanisms to detect and mitigate the influence of falsified position information in geographic routing protocols without using special hardware (to measure signal strengths or time-of-flight) or preinstalled infrastructure networks, the selected mechanisms will not prevent malicious nodes entirely from using falsified position information because the system cannot detect all the fake positions because of the weaknesses of the proposal: the solution uses only hard thresholds that are not used in some conditions, and the GPS could be jammed or manipulated so the cars cannot effectively determine their own position and can send no or only false beacons.

However, the proposed mechanisms will drastically limit the choice of fake positions because these positions must lie within a node's wireless transmission range. As a result, the possibilities for attackers using faked positions are significantly reduced.

# 3 Security challenges in VANETs

Despite the various benefits offered by VANETs, they introduce several security challenges to the research community. These security challenges are mostly concerned with the trade-off between authentication and non-repudiation versus privacy during communication within VANET environments. Another security challenge is delay sensitivity because significant delays prohibit the use of security protocols that have high overheads or rely on multiple stages of full-duplex communication between nodes [14, 15].

The use of traditional authentication mechanisms to address some security threats (such as illusion attacks) without affecting privacy in VANETs poses a new challenge for those VANET applications that need to authenticate nodes in VANETs [6, 11]. The other challenge is the restriction introduced by the transitory nature of interactions in a VN to use reputation-based schemes [16].

Another VANET security challenge is the forwarding of event-related messages on very large *ad hoc* networks of highly mobile nodes in such a way that the information can be trusted by receiving nodes because by doing so we improve traffic safety and mobility [10].

All these challenges need to be addressed to provide a secure VANET infrastructure where nodes (vehicles) can communicate securely with each other and infrastructure units.

# 4 Summary and conclusions of this work

Securing VANETs is becoming increasingly important given ubiquitous deployment and adoption by the transportation

**Table 1** Summary of security solutions for attacks in VANETs

| Attack | Description | Solution |
|---|---|---|
| malicious vehicle | this attack occurs when a malicious vehicle *M* can easily change its identity to node N without being punished | to isolate a malicious vehicle with the existence of pseudonym in VANETs, Liu *et al.* [1] proposed a probabilistic method which use Bloom filters to securely evaluate the honesty of a node itself with the help of TPD |
| brute force attacks | in a brute force attack, an attacker works through all possible vehicle identification number (VIN) in order to determine and compromise the identity of a vehicle | to deal with the type of attack, Langley *et al.* [4] proposed a secure method of authentication which requires use of some unique identification for vehicles concatenated with some large random value and then hashed using some hash algorithm |
| misbehaving and faulty nodes | when a node injects faulty data while complying with the implemented protocols, we are in presence of a misbehaving node. A faulty node may attempt to disrupt the system stopping to respond, drop messages, or act arbitrarily | Raya *et al.* [5] proposed a framework for the identification and local containment of these kind of attacks which consists of the following basic components: (i) the centralised revocation of a node by the CA, (ii) the local detection of misbehaviour, performed individually by each node and (iii) a distributed, localised protocol for the eviction of an attacker by its neighbouring nodes |
| malicious users | in this type of attack, an attacker uses random key sets during the attack in order to prevent his physical trajectory from being easily identified | Xi *et al.* [6] proposed a random keyset-based authentication protocol that preserves user privacy under the zero-trust policy, in which no central authority is trusted with the user privacy (users have to rely on OBUs to provide privacy) |
| traffic analysis attacks | in a traffic analysis attack, an attacker listens and/or compromises certain parts of the VANET to match a message sender with the recipient | Cencioni *et al.* [8] proposed VIPER: a vehicle-to-infrastructure communication privacy enforcement protocol which is resilient to traffic analysis attacks. The intuition behind this protocol is to have vehicles not to send their messages directly to the RSU, but to have vehicles acting as mix nodes |
| malicious nodes | this attack redirects network traffic by altering control message fields or by forwarding routing messages with falsified values. Also, malicious nodes can launch denial of service attacks | Florian *et al.* [4] proposed VARS, a completely distributed approach based on reputation that make use of direct and indirect trust as well as opinion piggybacking to enable confidence decisions on event messages |
| illusion attacks | in this attack, a malicious attacker creates a specific traffic situation and sends false traffic warning messages to decoy other drivers believe that a traffic event occurred [11] | to resolve this security threat, Lo *et al.* [11] proposed a new system architecture [called the plausibility validation network (PVN)] to check the raw data from sensors and further evaluate whether the incoming or generated message is valid or not |
| forging positions and Sybil attacks | a forging position attack refers to a type of attack where the attacker misleads vehicle safety systems to display warnings to their drivers while the Sybil attack is a well-known harmful attack whereby a vehicle claims to be several vehicles either at the same time or in succession | to prevent these kind of attacks, Yan *et al.* [12] proposed a novel solution that uses on-board radar as the virtual 'eye' of a vehicle. Although the 'eyesight' is limited because a modest radar transmission range, a vehicle can see surrounding vehicles and receive reports of their GPS coordinates. By comparing what is seen to what has been heard, a vehicle can corroborate the real position of neighbours and isolate malicious vehicles |
| position cheating and false position disseminating | positions cheating refers to an attack where nodes disseminate false positions (which falsify their position data) to modify the geographic routing regarding both performance and security | Leinmüller *et al.* [13] addressed the security of geographic routing by focusing on infrastructure-less autonomous position verification, where each node judges the position claims independently of others without using special hardware of preinstalled infrastructure networks |



*IET Commun.*, 2010, Vol. 4, Iss. 7, pp. 894–903
doi: 10.1049/iet-com.2009.0191

sector. Many security challenges still remain to be solved to support and enable a highly secure VANET infrastructure and secure VANET communications.

In this work, we have analysed the attacks that VANETs can be subjected to. We have focused on the following security issues namely: anonymity, key management, privacy, reputation and location. Table 1 provides a summary of the security attacks we have identified that may be launched on VANETs and the corresponding security solutions reported in the recent literature to mitigate those attacks. We hope that this survey will enable VANET designers and developers to build more secure and robust VANET architectures, protocols and applications in the future.

## 5 Acknowledgments

We thank the anonymous reviewers for their constructive suggestions and remarks which helped to improve the quality and presentation of this paper.

## 6 References

[1] LIU B., ZHONG Y., ZHANG S.: 'Probabilistic isolation of malicious vehicles in pseudonym changing VANETs'. Seventh Int. Conf. on Computer and Information Technology (CIT 2007), 2007, pp. 967–972

[2] SUEN T.: 'Geographic ad-hoc routing with anonymous properties'. PhD thesis, The Florida State University College of Arts & Sciences, 2007

[3] Vehicle Safety Communications Consortium, Vehicle Safety Communications Project, Task 11: WAVE/DSRC Security Extension, Final Task Report, 2004

[4] LANGLEY C., LUCAS R., FU H.: 'Key management in vehicular ad-hoc networks'. IEEE Int. Conf. on Electro/Information Technology (EIT 2008), 2008, pp. 223–226

[5] RAYA M., PAPADIMITRATOS P., AAD I., JUNGELS D., HUBAUX J.: 'Eviction of misbehaving and faulty nodes in vehicular networks', *IEEE J. Sel. Areas Commun.*, 2007, **25**, (8), pp. 1557–1568

[6] XI Y., SHA K., SHI W., SCHWIEBERT L., ZHANG T.: 'Enforcing privacy using symmetric random key-set in vehicular networks'. Eighth Int. Symp. on Autonomous Decentralized Systems (ISADS 2007), 2007, pp. 344–351

[7] BERTHOLD O., FEDERRATH H., KÖHNTOPP M.: 'Project anonymity and unobservability in the Internet'. Tenth Conf. on Computer (CFP 2000), 2000, pp. 57–65

[8] CENCIONI P., DI PIETRO R.: 'A mechanism to enforce privacy in vehicle-to-infrastructure communication', *Comput. Commun.*, 2008, **31**, (12), pp. 2790–2802

[9] TSIOUNIS Y., YUNG M.: 'On the security of ElGamal based encryption'. First Int. Workshop on Practice and Theory in Public Key Cryptography (PKC'98), 1998, pp. 117–134

[10] FLORIAN D., LARS F., PRZEMYSLAW M.: 'VARS: a vehicle ad hoc network reputation system'. Int. Conf. on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2005), 2005, pp. 454–456

[11] LO N., TSA H.: 'Illusion attack on VANET applications – a message plausibility problem'. IEEE Globecom Workshops, 2007, pp. 1–8

[12] YAN G., OLARIU S., WEIGLE M.: 'Providing VANET security through active position detection', *Comput. Commun.*, 2008, **31**, (12), pp. 2883–2897

[13] LEINMÜLLER T., MAIHÖFER C., SCHOCH E., KARGL F.: 'Improved security in geographic ad hoc routing through autonomous position verification'. Third Int. Workshop on Vehicular Ad Hoc Networks (VANET 2006), 2006, pp. 57–66

[14] BURMESTER M., MAGKOS E., CHRISSIKOPOULOS V.: 'Strengthening privacy protection in VANETs'. IEEE Int. Conf. on Wireless & Mobile Computing, Networking & Communication (WIMOB 2008), 2008, pp. 508–513

[15] JAKUBIAK J., KOUCHERYAV Y.: 'State of the art and research challenges for VANETs'. Fifth IEEE Consumer Communications and Networking Conf. (CCNC 2008), 2008, pp. 912–916

[16] PARNO B., PERRIG A.: 'Challenges in securing vehicular networks'. Fourth Workshop on Hot Topics in Networks (HotNets-IV), 2005

Electron Commer Res
DOI 10.1007/s10660-010-9052-2

# Implementation and performance evaluation of a payment protocol for vehicular ad hoc networks

**Jesús Téllez Isaac · Sherali Zeadally · José Cámara Sierra**



**Abstract** *Vehicular ad hoc networks* (*VANETs*) are envisioned to support the development of a wide range of attractive applications such as payment services which require the design of payment systems that satisfy additional requirements associated with VANETs. The wide range of scenarios (with or without connectivity restriction) arising from vehicle-to-vehicle and vehicle-to-roadside communications have opened up new security challenges which must be considered by payment system designers to achieve the same security capabilities independent of the scenario where payment occurs. We propose and implement a new payment protocol (called KCMS-VAN protocol) for those scenarios where the client cannot communicate directly with the credit card issuer (the client's financial institution) for authentication. Our proposed protocol uses symmetric-key operations which require low computational power and can be processed much faster than asymmetric ones. We also present a performance evaluation of the proposed payment protocol and the results obtained demonstrate that optimal performance can be achieved with it.

**Keywords** Performance evaluation · Vehicular ad hoc networks · Payment protocol · Security · Implementation

J. Téllez Isaac (✉)
Computer Science Department, FACYT, Universidad de Carabobo, Av. Universidad, Sector Bárbula, Valencia, Venezuela
e-mail: jtellez@uc.edu.ve

S. Zeadally
Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC 20008, USA
e-mail: szeadally@udc.edu

J.C. Sierra
Computer Science Department, Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911, Leganés, Madrid, Spain
e-mail: sierra@inf.uc3m.es

Published online: 11 July 2010

## 1 Introduction

Vehicular Ad hoc NETworks (VANETs) are a form of Mobile Ad hoc NETwork (MANET) that aims to provide communications among nearby vehicles (also known as Inter-Vehicle Communication -(**IVC**)-) and between vehicles and nearby roadside base-stations (also referred to as Vehicle-to-Roadside Communication -(**VRC**)-).

The application space for vehicle-to-vehicle and vehicle-to-roadside communications is vast and opens up tremendous business opportunities and research challenges among which security is an important one. VANETs are envisioned to support the development of a wide variety of new attractive applications that can be broadly divided into two major categories [21, 24, 37]: (1) *Safety-related applications*, and (2) *Comfort applications*. Although the industry and academia have concentrated their research efforts on safety-related applications due to its importance for the automotive domain, it is expected that research on Comfort applications (that also offer great business opportunities) will continue to attract the attention of researchers and designers to develop non-safety VANET-based applications.

To enable payments in VANETs, it is necessary to build payment systems that satisfy the additional requirements associated with vehicular ad hoc networks. As mentioned previously, both vehicle-to-vehicle and vehicle-to-roadside communications open up new security challenges that must be considered by payment system designers to achieve the same security capabilities independent of the scenario where payments occur.

The above situation can be represented in the real word by the following example: a client is on the road and stops at a gas station to purchase gas or some others goods at the gas station store with some payment card. In both cases, if the client is not able to communicate with the card issuer (to authorize the payment) from the application unit (because of the absence of the infrastructure necessary such Road-Side Units (**RSUs**), or a HotSpot (**HS**)-), the client should still be able to perform the payment using the merchant's infrastructure.

An example of the above situation is the restricted connectivity scenario shown in Fig. 1 where a car (henceforth referred to as the client), with an On-Board Unit (OBU) and an Application Unit (AU) can only connect to the merchant during a payment transaction due to the lack of Internet access with its AU. This situation creates a potential security problem because the client cannot send any kind of messages directly to the issuer and has to do it through the merchant (who should not be able to change the content of the messages but must keep evidence of the payment). Note that in our scheme, the merchant takes an active role in the payment process because it acts as a proxy to allow the communication between the client and the card issuer.


**Fig. 1** A restricted connectivity scenario in VANETs

On the other hand, thinking in a VANET scenario with moving vehicles, we can found the following example in the real world which represents a restricted connectivity scenario: a client is driving on a road and wants to purchase a weather forecast using some payment card. This situation creates the same potential security problem than the aforementioned example. However, even in this kind of situations, the scheme proposed in this paper could be applied (according to the ideas claimed in the Carlink project, proposed by [26]). However, we can include any experimental that show that the proposed protocol performs well when the client is on the move because this study is beyond the scope of this paper.

In order to provide authentication in electronic payment systems (including mobile commerce), many methods have been considered but symmetric and asymmetric signature methods are chosen for authentication [10]. However, traditional asymmetric signature schemes make the signature computations very expensive and are not suitable for those portable devices (such as the ones typically attached to an OBU [7]) available in the market not based on the Texas Instruments TMS320C55x processors family [11].

The protocol we propose in this work is for the case where we do not have direct communication between the client and the card issuer. As a result, the above schemes are not suitable because the client has connectivity restrictions, and consequently, communication with other parties (such a Certification Authority, for verifying a certificate) is not possible during a payment transaction. Therefore, the use of symmetric signature scheme is required to satisfy the requirements of the protocol proposed in this work. Symmetric cryptography (which employs a shared key between two parties) provides (such as asymmetric cryptography) message confidentiality, message integrity, and party authentication, and represents an alternative in the construction of secure protocols for mobile payment systems. Symmetric-key operations do not require high computational power nor do they require additional communications steps (as in the case of protocols based on public-key infrastructures where public-key certificates have to be verified by a Certificate Authority).

To address the issue of direct communication between the client and the card issuer for authentication purposes, we designed and implemented a payment protocol that allows the client to send a message to the issuer through a merchant (who will not be able to decrypt the message). The proposed protocol, called the Kiosk Centric Model payment protocol for VANETs (henceforth referred to as the KCMS-VAN Protocol), employs symmetric-key operations in all engaging parties to reduce both, the setup cost for the payment infrastructure and the transaction cost. Furthermore, it supports both credit-card and debit-card transactions, protects the real identity of the client during the payment and should be used with a portable device attached to an AU.

We analyze the performance of the KCMS-VAN protocol with actual mobile phones and PDAs as the implementation platforms. This allows us to further demonstrate that our client-side application can be installed on multiple heterogeneous Java™-enabled memory-constrained portable wireless handheld devices.

The rest of the paper is organized as follows. Section 2 presents recent related works related to our research. In Sect. 3, we propose and describe the payment protocol for vehicle-to-roadside scenarios in VANETs. Section 4 presents the implementation of KCMS-VAN in detail. We present performance evaluation results of the proposed protocol in Sect. 5. Finally, make some concluding remarks in Sect. 6.

## 2 Related work

In recent years, several studies have been conducted to improve the security of mobile payment systems. Many of these efforts have also been dedicated to unify concepts and scenarios into frameworks that will be useful in the design of new electronic payment systems. Many of these studies have considered the following methods to provide authentication in electronic payment systems (including mobile commerce): username/password, symmetric and asymmetric cryptography, smart card, and biometric methods. Username/password does not offer enough security for m-commerce. Biometric approaches are not feasible at present and smart card would require an external device to read the card. Symmetric and asymmetric signature have been widely used for authentication purposes.

Most of the protocols proposed in recent years are based on the Full Connectivity scenario (where all the entities are directly connected one to another [8]) and employ asymmetric-key operations [5, 9, 12, 17, 36] whereas the remaining scenarios use symmetric-key operations which is more suitable for wireless networks [15]. Unfortunately, usage of those protocols is not possible in scenarios where direct interaction among two of its parties is not allowed due to the communication restriction imposed by the model (as in the case for the Kiosk Centric Model [8]). As a result, a new trend has emerged which is to develop mobile payment systems based on restricted connectivity scenarios, achieving the same security and performance levels as the Full Connectivity Scenario. Protocols proposed by Téllez et al. [27–32] constitute examples of mobile payment protocols suitable for scenarios with communication restrictions. However, such proposals are theoretical and do not capture practical performance issues we encounter with real systems. Thus in this work, we present an implementation of our proposed secure payment protocol and evaluate its performance.

## 3 Proposed secure payment protocol for vehicle-to-roadside scenarios in VANETs

### 3.1 The kiosk centric model payment protocol for VANETs (KCMS-VAN) model

The KCMS-VAN protocol was designed taking into consideration the model suggested by Abad-Peiro et al. [1] such that it can be applied by many different payment methods. Thus, based on the General Payment Model of Abad-Peiro et al. [1], our proposed secure payment protocol uses the following entities:

1. *Client:* a user who wants to purchase goods or services from the merchant. In our proposed protocol, the client is a userside entity equipped with an On-Board Unit (called **OBU**) and/or an Application Unit (called **AU** that may use the OBU's communication capabilities). An AU can be an integrated part of a vehicle and be permanently connected to an OBU or could be a portable device such as a Personal Digital Assistant (PDA), a mobile phone or a gaming device that can dynamically attach to and detach from an OBU [7].

2. *Merchant:* an entity that has products or services to offer or sell. This entity could be a computational one (such as a normal web server, a roadside computing station or an intelligent vending machine) or a physical one (such as a gas station that makes it possible to pay from within an AU) which the user can connect to using a short range link (using wireless technologies such as Wi-Fi or Bluetooth). Moreover, this entity connects with the Payment Gateway (an entity which provides the necessary infrastructure to allow a merchant to accept credit card and other forms of electronic payment) through a secure channel allowing the client to communicate with the issuer using this connection.
3. *Acquirer:* is the merchant's financial institution. It verifies the validity of the deposited payment instructions and manages the merchant's account including fund transfers.
4. *Issuer:* is the client's financial institution. It provides electronic payment instructions to the client to use in a payment and manages the client's account including fund transfers.
5. *Payment gateway:* an additional entity that acts as a medium between acquirer/issuer on the bank's private network side and the client/merchant at the Internet side [15].

The parties of the former model communicate with each other regarding fund transfers, using the following 3 primitive payment transactions:

– In *Payment*, the client transfers the value to the merchant.
– In *Value Subtraction*, the client requests that the payment gateway (on behalf of issuer) deducts the money from the client's account.
– In *Value Claim*, the merchant requests that the payment gateway (on behalf of acquirer) transfers money to the merchant's account.

The five entities in KCMS-VAN and their interactions are shown in Fig. 2. Note that there is no direct interaction between the client and the issuer. Moreover, the connection between the client and the merchant (denoted as the dotted arrow) is set up through a wireless channel (such as IEEE 802.15.1, Bluetooth, IEEE 802.11a/b/g).

Interaction between the merchant and the payment gateway (depicted as solid arrow in Fig. 2) should be reliable and secure against passive and active attacks. Therefore, the connection should be established through a secure wired channel using a well-known security protocol such as secure socket layer/transport layer security (SSL/TLS). Note that the issuer, the acquirer and the payment gateway operate in the bank private network. The security of the messages exchanged among them is beyond of the scope of this paper.

Before receiving payment services, the client must register with an Issuer. Generally, this registration can be done either personally at the issuer's premises or via the issuer's website. During the client's registration, the following steps are performed:

1. The Client shares his/her credit- and/or debit-card information (CDCI) with the issuer (who will not reveal it to any merchant).
2. The Issuer assigns several nicknames to the client. Those nicknames are known only to the client and the issuer [10].

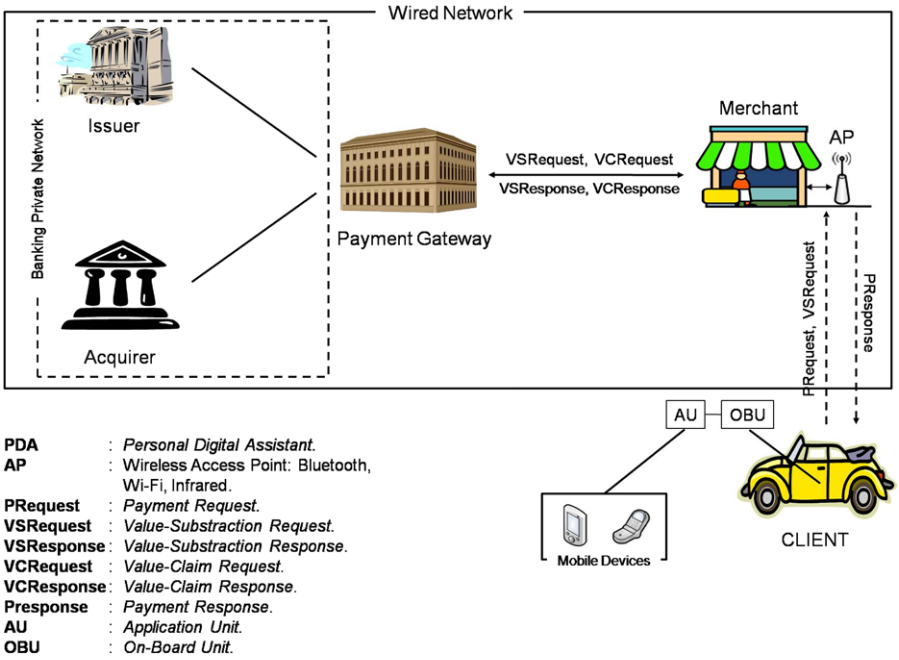


**Fig. 2** KCMS-VAN architecture

## 3.2 Notations

All the entities involved in our proposed payment protocol are called parties and communicate through wireless and wired networks. The symbols C, M, PG, I, A are used to denote the names of the parties Client, Merchant, Payment Gateway, Issuer and Acquirer, respectively. The following symbols are used to represent other messages and protocols:

- **$ID_P$:** The identity of party $P$ that contains the contact information of $P$.
- **$NID_C$:** Client's nickname, temporary identity.
- **TID:** Identity of transaction that includes time and date of the transaction.
- **$TST_P$:** Timestamp generated by $P$.
- **Stt:** The status of transaction (Stt = {Accepted, Rejected}).
- **OI:** Order information ($OI = \{TID, OD, h(OD,\text{Price})\}$) where $OD$ and Price are order descriptions and its amount.
- **TC:** The type of card used in the purchase process ($TC = \{\text{Credit, Debit}\}$).
- **TIDReq:** The request for $TID$.
- **MIDReq:** The request for $ID_M$.
- **$SEC_{A-B}$:** The master secret shared between parties $A$ and $B$.
- **$\{M\}_X$:** The message $M$ symmetrically encrypted with the shared key $X$.

**Table 1** Details of key generating technique

| | |
|---|---|
| Generating $KS_{C-M_i}$ | $KS_{C-M_1} = h(1\text{-bit-shift-of-}KS_{C-M})$,<br>$KS_{C-M_2} = h(2\text{-bit-shift-of-}KS_{C-M}), \ldots,$<br>$KS_{C-M_n} = h(n\text{-bit-shift-of-}KS_{C-M})$ |
| Generating $KS_{M-PG_k}$ | $KS_{M-PG_1} = h(1\text{-bit-shift-of-}KS_{M-PG})$,<br>$KS_{M-PG_2} = h(2\text{-bit-shift-of-}KS_{M-PG}), \ldots,$<br>$KS_{M-PG_n} = h(n\text{-bit-shift-of-}KS_{M-PG})$ |
| Generating $KS_{C-I_z}$ | $KS_{C-I_1} = h(1\text{-bit-shift-of-}(CDCI, KS_{C-I}))$,<br>$KS_{C-I_2} = h(2\text{-bit-shift-of-}(CDCI, KS_{C-I})), \ldots,$<br>$KS_{C-I_n} = h(n\text{-bit-shift-of-}(CDCI, KS_{C-I}))$ |

- $h(M)$**:** The one-way hash function of the message $M$.
- **MAC(X,K):** Message Authentication Code of the message $X$ with the key $K$.
- $KS_{A-B_t}$**:** The session key shared between parties $A$ and $B$, generated applying a hash function with $t$-bit cyclic shifting (either left shift or right shift) of $KS_{A-B}$. More details about this technique is given in the next section.

### 3.3 Session key generation technique

Two efficient key generation techniques are employed in KCMS-VAN to generate the sets of session keys used in transactions. In both of them, the main idea is to apply a hashing algorithm with one-bit cyclic shift of a master secret each time a session key is generated [16]. As a result, the performance of the protocol is increased due to the reduced frequency of the key update processes.

The key set $KS_{C-M_i}$ (with $i = \{1, \ldots, n\}$), is generated from the secret key $KS_{C-M}$ and stored in the client's device and merchant's terminal. The set $KS_{M-PG_k}$ (with $k = \{1, \ldots, n\}$), is generated from the secret key $KS_{M-PG}$ and stored both in the merchant and Payment gateway terminals. The set $KS_{C-I_z}$ (with $z = \{1, \ldots, n\}$), is generated from the key secret $KS_{C-I}$ and is stored in the client's device and the issuer's terminals.

The details of the generation of the different sets of session keys are shown in Table 1.

### 3.4 Our proposed kiosk centric model payment protocol for VANETs (KCMS-VAN)

The KCMS-VAN Protocol is composed by two sub-protocols: the *KCMS-VAN Merchant Registration Protocol (MRP)* and the *KCMS-VAN Payment Protocol (PP)*.

For the *KCMS-VAN Merchant Registration Protocol*, the client has to register with the merchant to send the master key $KS_{C-M}$. The protocol has to be executed every time the client wants to perform transactions with a merchant. The details of the protocol are shown as follows:

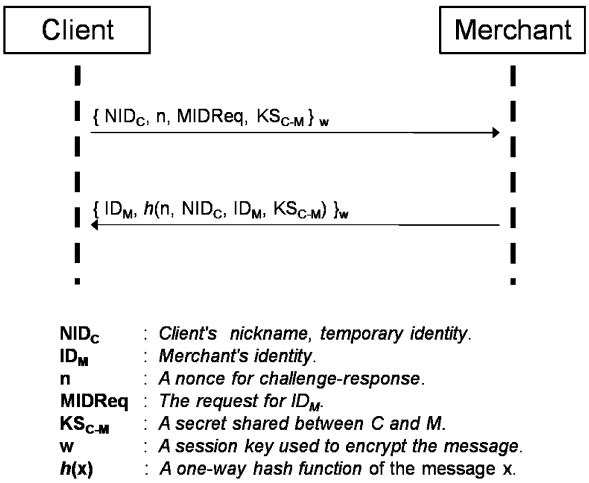


**Fig. 3** Merchant registration protocol message exchange

$$\mathbf{C} \rightarrow \mathbf{M}: \quad \{NID_C, n, MIDReq, KS_{C-M}\}_w$$
$$\mathbf{M} \rightarrow \mathbf{C}: \quad \{ID_M, h(n, NID_C, ID_M, KS_{C-M})\}_w$$

Client **C** generates $KS_{C-M}$ which is to be shared with the merchant and sends it with her/his nickname $NID_C$, a nonce $n$ for challenge-response and $MIDReq$ to **M**. After the merchant **M** receives the message, she/he sends $h(n, NID_C, ID_M, KS_{C-M})$ and the merchant's identity ($ID_M$). Note that both messages (from the client to the merchant and viceversa) are encrypted with the session key $w$. The transmitted messages between the client and the merchant during the *Merchant Registration Protocol* are shown in Fig. 3.

Once **C** and **M** have exchanged the necessary information, they can generate a new set of $KS_{C-M_i}$ using the same key generation technique. The client may then start the *KCMS-VAN Payment Protocol*.

For the *KCMS-VAN Payment Protocol*, the client purchases goods from the merchant and pays for them using her/his credit-card or debit-card. This protocol is formalized as follows:

$$(1)\ \mathbf{C} \rightarrow \mathbf{M}: \quad NID_C, i, TIDReq$$
$$\mathbf{M} \rightarrow \mathbf{C}: \quad \{TID, ID_M\}_{KS_{C-M_i}}$$

$$(2)\ \mathbf{C} \rightarrow \mathbf{M}: \quad \{OI, Price, NID_C, ID_I, TST_C, z, h(KS_{C-I_z}), VSRequest\}_{KS_{C-M_i}}, MAC[(OI, Price, NID_C, ID_I, TST_C, z), KS_{C-M_{i+1}}].$$

$$VSRequest = (MAC[(Price, h(OI), TST_C, TC, ID_M), KS_{C-I_z}], TC, TST_C)$$

$$(3)\ \mathbf{M} \rightarrow \mathbf{PG}: \quad \{VCRequest, ID_M, z\}_{KS_{M-PG_k}}, k, MAC[(ID_M, k, z, VCRequest), KS_{M-PG_{k+1}}]$$

$$VCRequest = (VSRequest, TST_M, h(OI), TID, Price, NID_C, ID_I)$$

(4) Under banking private network,

(4.1) **PG** → **I:** $NID_C, ID_M, VSRequest, TID, h(OI), z,$
$Price, h(KS_{M-PG_{k+1}})$

(4.2) **PG** → **A:** $Price, ID_M$

(4.3) **I,A** → **PG:** $VSResponse, Stt, h(Stt, h(OI))$

$$VSResponse = \{Stt, h(OI)\}_{KS_{C-I_z}}$$

(5) **PG** → **M:** *VCResponse*

$$VCResponse = \{Stt, VSResponse, h(Stt, h(OI), h(KS_{C-I_z}))\}_{KS_{M-PG_{k+1}}}$$

(6) **M** → **C:** *PResponse*

$$PResponse = \{VSResponse\}_{KS_{C-M_{i+1}}}$$

**Step 1:** The client **C** and merchant **M** exchange the information necessary to start the protocol by performing the following sub-step.

1-1: **C** sends his/her nickname ($NID_C$), the index $i$ (that will be used to generate the session key between the client and the merchant) and the request for the transaction identity (*TIDReq*) to **M**.

1-2: **M** receives the request and sends back its identity ($ID_M$) and *TID* to **C**, encrypted with $KS_{C-M_i}$.

**Step 2:** Client **C** creates a *Payment Request* (referred to as the General Payment Model as described in [1, 15]) in the following sub-steps.

2-1: A *Value-Subtraction Request* (called *VSRequest*) is created and it includes $MAC[(Price, h(OI), TST_C, TC, ID_M), KS_{C-I_z}]$, $TST_C$ and $TC$.

2-2: A new message is created which includes $C's$ nickname, $I's$ identity, *Price*, *OI* (used to inform **M** about the goods and prices requested), *VSRequest*, the index $z$, timestamp $TST_C$ read from $C's$ clock and $h(KS_{C-I_z})$ (used to prevent the payment from modifying the approval result in Step 4-5).

2-3: The message created in the previous sub-step (henceforth referred to as the *Payment Request*) is encrypted with the session key $KS_{C-M_i}$.

2-4: The *Payment Request* is sent to the merchant.

**Step 3:** The merchant **M** generates the *Value-Claim Request* (called *VCRequest*) by performing the following sub-steps.

3-1: The message received from **C** is decrypted to extract *OI* and $TST_C$.

3-2: The timeliness of the *Payment Request* is verified. If the check is successful, the following sub-steps will be performed.

3-3: The *VCRequest* is prepared, and contains the *VSRequest*, $TST_M$, *h(OI)*, identity of transaction, order's amount, $C's$ nickname, and $I's$ identity.
3-4: The *VCRequest*, the $M's$ identity and the index $z$, are encrypted with $KS_{M-PG_K}$.
3-5: The encrypted message is sent in sub-steps 3-4 to **PG** with $k$.

**Step 4:** Using the private network of the banking institution, the Payment Gateway (**PG**) performs the following sub-steps to verify the payment.
4-1: The *VCRequest* is decrypted to retrieve *VSRequest* and the others fields.
4-2: The timeliness of *VCRequest* is verified. If the check is successful, the following steps are executed.
4-3: The *VSRequest* and other important, such as: $h(OI)$, $TID$, $ID_M$, $Price$, $z$ and $h(KS_{M-PG_{k+1}})$ are forwarded to the issuer (**I**) where it is decided whether to approve or reject the transaction.
4-4: $ID_M$ and the requested price $Price$ are sent to confirm to the Acquirer **A** that the client is the party to whom the requested amount $Price$ will be transferred to.
4-5: The approved result (*Stt*) and Value-subtraction Response (called *VSResponse* and encrypted with $KS_{C-I_z}$) are received from the issuer **I**. It is worth noting that the *VSResponse* is prepared by the issuer after (a) checking the timeliness of *VSRequest* and the validity of the client's account, and (b) after transferring the total amount of *OI* to the merchant's account.

**Step 5:** The Payment Gateway **PG** generates the *Value-Claim Response* (called *VCResponse*) in the following sub-steps.
5-1: A *VCResponse* is created that includes *Stt*, *VSResponse* (which will be forwarded to the client **C**) and $h(Stt, h(OI), h(K_{C-I_z}))$.
5-2: The *VCResponse* is encrypted with $KS_{M-PG_{k+1}}$ and sent to **M**.

**Step 6:** Merchant **M** performs the following sub-steps.
6-1: The *VCResponse* is decrypted to retrieve *VSResponse* and other fields.
6-2: The merchant's own *OI* is compared with the received *h(OI)*. If they do not match, then the client performs Sub-step 6-3a, otherwise the client performs Sub-step 6-3b.
6-3a: A message is sent to the Payment Gateway to notify it of the response failure. The Payment Gateway then starts a recovery procedure or resends the message.
6-3b: The *Payment Response* (called *PResponse* and represents the result of the client's request) is created and includes the *VSResponse*.
6-4: The *PResponse* is encrypted with $KS_{C-M_{i+1}}$ and sent to **C**.

### 3.5 Security analysis

We analyze the security of our proposed secure Payment Protocol in this section. In the case of the KCMS-VAN protocol, the client uses a nickname $NID_C$ (a temporary

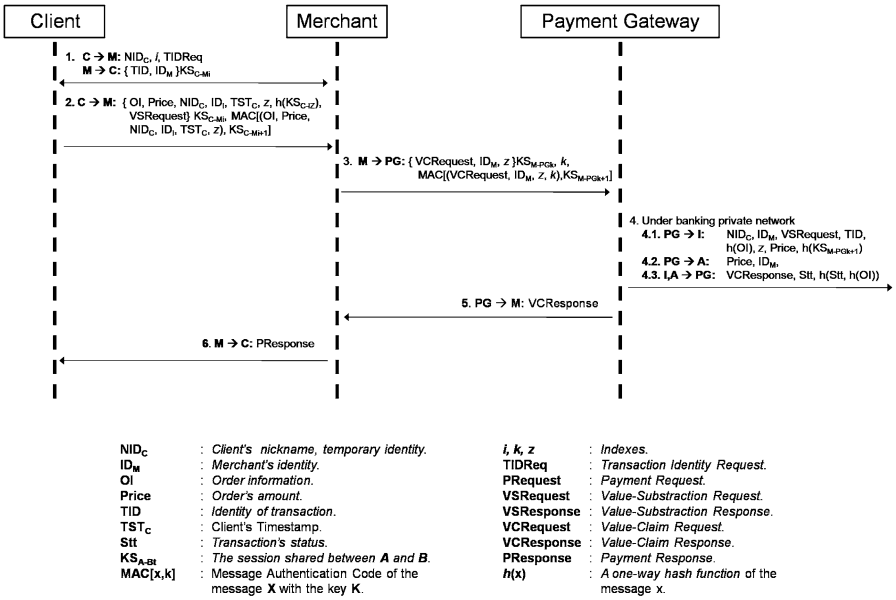


**Fig. 4** KCMS-VAN payment protocol message exchanges

identity known only to the client and the issuer) instead of his/her real identity. As a result neither the merchant nor the payment gateway can map the nickname to the client's true identity. This anonymity protects relevant information from third parties but not unrestrained anonymity [2].

The Confidentiality of messages transmitted in each transaction should be protected while in transit, specially in wireless networks where anybody can easily eavesdrop the transmitted messages. In our protocol, we protect those confidential messages by employing symmetric cryptography which uses a secret shared between two parties (called sender and receiver) that wish to communicate safely without revealing details of the message. Moreover, the encryption key allows receiver and sender to authenticate each other. We also use the Message Authentication Code (MAC) to maintain the integrity of important messages. The proposed scheme provides integrity and identity authentication.

Although, generally, in any transaction a party should not trust others unless they can provide a proof of trustworthiness [15], in our protocol we assume the trust relationship between the client and the issuer because the client has a credit- and/or debit-card issued by the issuer who will not reveal it to any other party.

Since $KS_{C-I_Z}$ could be generated only by the client or issuer but not by merchant, she/he is able to provide a non-repudiable evidence to prove to other parties that client has sent a message or requested merchant to perform transaction. Thus, Non-repudiation of transaction is ensured by $KS_{C-I_Z}$ in the proposed protocol.

In the following paragraphs, we adopt the realistic analysis method proposed by [3] to discuss two possible attacks against our proposed protocol:

– **Replaying attack:** If an intruder $E$ wants to impersonate a legal user by replaying the user's transmitting contents, the timestamp included in the transmitted message ensure the freshness of the message and avoids replaying attacks. Thus, our proposed protocol is secure against replaying attack.
– **Key guessing attack:** It is very difficult for any intruder $E$ to get information related to the secret key through analyzing intercepted data because the authentication key is dynamic (exploiting, on each transaction, different session keys based on the same master secrets). Then, our proposed protocol is secure against the key guessing attack.

## 4 System design and implementation of KCMS-VAN protocol

### 4.1 Environment settings

The various applications (client, merchant, payment gateway and issuer) in our KCMS-VAN implementation have been developed in Java [34], using JavaME [35] for the implementation of the client in mobile devices. Since the JaveME Mobile Information Device Profile (MIDP) version 2.0 does not have the necessary security support, we have used security APIs from [33], a light-weight API suitable for use in any environment (including the newly released JavaME).

In Table 2 we show the hardware configuration of the devices used to run and benchmark all the applications.

### 4.2 Chosen cryptographic operations

*Security* and *computational* requirements are two important aspects which should be considered in order to choose suitable algorithms for encryption and hash functions

**Table 2** Hardware specifications of system used for performance evaluation

| Protocol Party | Device | Features |
|---|---|---|
| Client | Nokia™ N95 | - 332 MHz Texas Instruments OMAP 2420 (ARM11-based).<br>- 160 MB of RAM.<br>- Symbian OS 9.2, S60 rel. 3.1. |
| | Palm™ T\|X | - Intel 312 MHZ ARM-based processor.<br>- 100 MB of RAM.<br>- Palm OS® Garnet 5.4.<br>- IBM's Java Virtual Machine. |
| Merchant<br>Payment Gateway<br>Issuer<br>Acquirer | Sony™ Vaio VGN-SZ450N | - Intel Core 2 Duo (2 GHz).<br>- 2 GB of RAM.<br>- Windows Vista Business.<br>- Sun's Java Virtual Machine. |

in our application. From past results discussed in [23] and [22], computational requirement is directly related to energy consumption of operating devices in that the higher computation the algorithm requires, the higher energy it consumes.

We present and discuss the cryptographic algorithms we used for our implementation bellow:

– **Symmetric-key algorithm:** The results presented in [23] and [22] shows that the *Advanced Encryption Standard* algorithm (called *AES* [20, 25]) requires less energy consumption and computation compared to the *Triple Data Encryption Standard* algorithm (called *3DES* [18, 19]). Therefore, AES is more suitable to operate on power-constrained mobile devices. But, comparing both algorithms in terms of security, 3DES with 112-bit key length provides the equivalent security of RSA public-key algorithm with 2,048-bit key length, while AES with 128-bit key length provides the security equivalent to RSA public-key algorithm with 3,072-bit key length [6].

  We deployed the AES algorithm (with 128-bit key) as the symmetric-key encryption algorithm for our implementation because it provides higher security, faster operation, and lower energy consumption compared to 3DES.

– **Hash function:** In our implementation, we chose MD5 algorithm [18] because it requires less computation and consumes less energy than *Secure Hash Algorithm version 1* (Called *SHA-1* [18]) as shown in citeravi2002 and [22]. Moreover, MD5 can produce the same length of output to the length of an AES key (128 bits).

– **Keyed-hash function:** The *Hashed Message Authentication Code with Message Digest 5* algorithm (called *HMAC-MD5* [14]) is used in our application to perform keyed-hash operation because the key length of HMAC-MD5 is equivalent to the length of each session key. Moreover, HMAC-MD5 is considered a secure keyed-hash algorithm available for wireless networks since no attacks have emerged on HMAC even when it is implemented with hash functions that are not weak collision-resistance [4].

### 4.3 The KCMS-VAN software at the client side

The KCMS-VAN protocol requires a software (henceforth referred to as the KCMS-VAN Wallet) at the client's side for purchase transactions. The client can obtain the above application by connecting to the issuer's web site to download it or sending a request to the issuer to receive it by mail. After the client has downloaded or received the KCMS-VAN wallet, she/he should install it on her/his mobile device. Note that during the installation process, the client is requested to set up her/his own password (henceforth referred to as *KWP* password) to protects unauthorized users to: a) open the program, b) authorize the payment transaction, or c) access client's information and key file. Figure 5 illustrates snapshots of the main screen and main menu of KCMS-VAN wallet on both Nokia™ N95 (top) and Palm™ T|X devices (bottom). The left screen shows the main screen of KCMS-VAN wallet whereas the right screen shows the main menu.

The KCMS-VAN wallet provides the following functionalities:

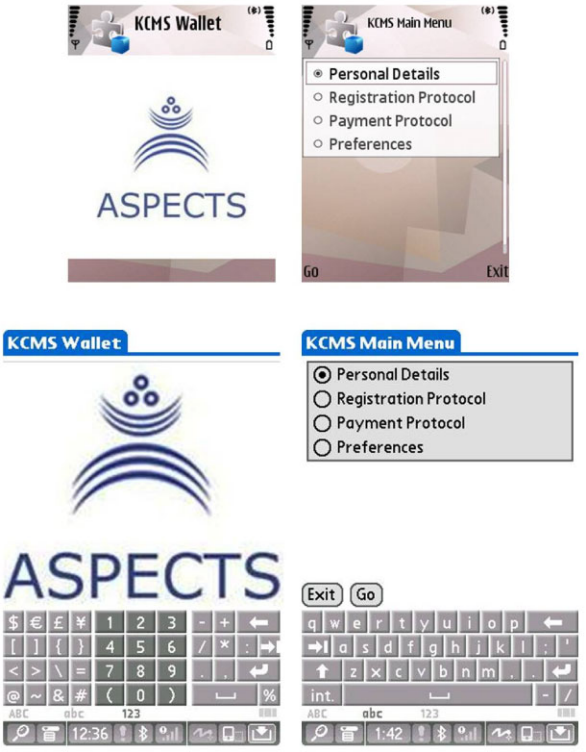


**Fig. 5** Snapshots of main screen and main menu of the KCMS-VAN wallet on both Nokia N95 and Palm T|X devices

- **Personal Details:** The Client's personal information is used in both *Merchant Registration* and the *Payment Fulfillment*. To prevent the client from being prompted for her/his information in both phases, the KCMS-VAN wallet allows client to store in a file (protected by the KCMS Wallet Password -*KWP*-) his/her personal information (such as name, contact information, issuer ID and credit-and/or debit-card information) for further uses. To achieve that, the KCMS-VAN wallet prompts the user to enter the above information and store it at the client's mobile device. Figure 6 illustrates the KCMS-VAN wallet prompting for the client to enter her/his details on both Nokia™ N95 and Palm™ T|X devices.
- **Key Generation:** To generate a new session key from the master key, the KCMS-VAN wallet call *generateItemofSet()* with two values: a master key and a random number *j*. Upon receiving the master key, its BigInteger representation is created and the number of zeros according to the value of *j* is added to the right to the master key, using the *shiftleft()* to perform the left-cyclic. Then, the new value will be put through a MD5 has function to produce a new 128-bit session key. The java code for *generateItemofSet()* is shown in Listing 1.

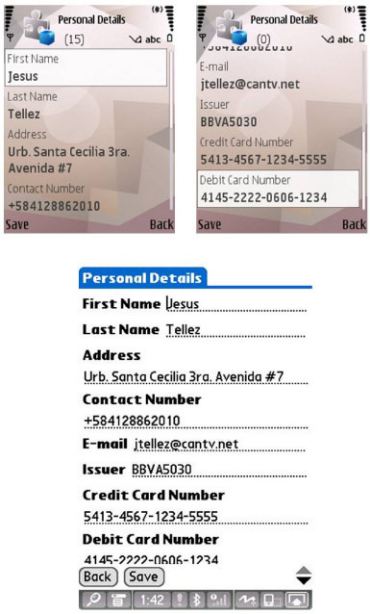


**Fig. 6** Snapshots of KCMS wallet personal details form on both Nokia N95 (*top*) and Palm TlX (*bottom*) devices

```
public String generateItemofSet(byte[] secret,int index)
   {
       String key_Base = new String(secret);
       String cadr = "", resul = "";
       BigInteger cadb;
       long valcar;
       char[] characters = key_Base.toCharArray();

       for (int i = 0;i < characters.length; i++)
          {
             valcar = characters[i];
             cadr   = cadr + Long.toString(valcar, 2);
          }

       cadb = new BigInteger(cadr, 2);
       cadb = cadb.shiftLeft(index);
       cadr = cadb.toString(16);
       resul = new String(this.hashfunc(cadb.toString(16),"MD5"));
       return (resul);
   }
```

**Listing 1** Java code for *generateItemofSet()*.

The above key generation technique can be directly applied to generate the sets of session keys $KS_{C-M_i}$ and $KS_{M-PG_k}$ from the key $KS_{C-M}$ and $KS_{M-PG}$, respectively. To generate the set of session keys $KS_{C-I_z}$, the credit- and/or debit-card information (CDCI) is treated as the BigInteger and added to the value of $KS_{C-I}$ before doing the *shiftleft()* operation. Figure 7 shows an example of the proposed session key generation technique.

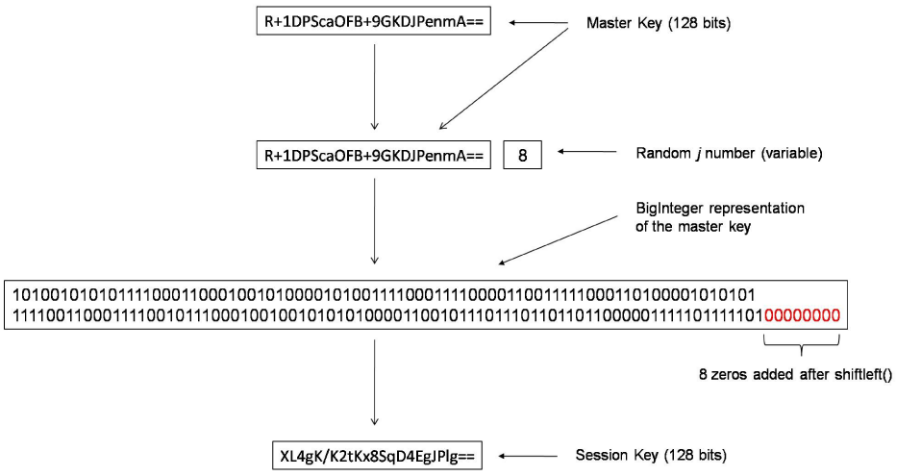


**Fig. 7** Example of the KCMS wallet key generation technique

```
public String generateAESkey(int skey)
    {
        byte[] AESkey = this.generateBytes(skey);

        return (new String(Base64.encode(AESkey)));
    }
```

**Listing 2** Java code for *generateAESKey()*.

– **Merchant Registration:** In order to make payments to a merchant, the client has to run the *Merchant Registration Protocol* to register with the merchant to share the master key $KS_{C-M}$. First, the client is prompted to introduce the *KWP* password to retrieve her/his personal information (such as name, address, contact number and email address) stored on the client's device. The KCMS-VAN wallet then generates the secret $KS_{C-M}$ and the session key by using *generateAESKey()* [13]. The java code used for *generateAESKey()* is presented in Listing 2.

After the keys have been generated, the KCMS-VAN wallet encrypts the client's personal information, the secret $KS_{C-M}$, a nonce and a request for the merchant's identity. All these data are encrypted with the session key $w$ using *AESEnc()*, before being sent to the merchant to register the client. The format of the client registration message is shown in Table 3 whereas the java code of *AESEnc()* is shown in Listing 3.

Upon receipt of the message, the merchant decrypts it using the *AESDec()* operation (see the java code in Listing 4) to retrieve the client's information including the secret $KS_{C-M}$. The merchant then encrypts, with the session key $w$ and using *AESEnc()*, her/his identity and $h(n, NID_C, ID_M, KS_{C-M})$. The encrypted message is sent to the client to confirm the registration.

**Table 3** Client registration message format

| Fields | Size (bits) |
|---|---|
| Client Data | Variable (Max. 1064) |
| $KS_{C-M}$ | 128 |
| Nonce | 128 |
| MIDReq | 48 |

```
public String AESEnc(String mkey,String StToEnc)
   {
        String resul = null;

        byte[] key    = Base64.decode(mkey.getBytes());
        byte[] input = StToEnc.getBytes();
        BlockCipher engineAES          = new AESLightEngine();
        BufferedBlockCipher cipherAES =
              new PaddedBlockCipher(new CBCBlockCipher(engineAES));
        cipherAES.init(true, new KeyParameter(key));
        byte[] cipherText =
              new byte[cipherAES.getOutputSize(input.length)];

        int outputLen =
         cipherAES.processBytes(input,0,input.length,cipherText,0);

        try
           {
               cipherAES.doFinal(cipherText, outputLen);
               resul = new String(Base64.encode(cipherText));
           }
        catch (CryptoException ce)
           {
               resul = "";
           }
        return(resul);
   }
```

**Listing 3** Java code for *AESEnc()*.

At the client side, the client retrieves the confirmation of her/his registration by decrypting the message using *AESDec()*. Then, the KCMS-VAN wallet stores the secret $KS_{C-M}$ in a key file protected with the *KWP* password. Once the registration is done, the merchant stores on her/his device the client's information together with the secret $KS_{C-M}$ and the key $KS_{C-M_i}$ (generated from $KS_{C-M}$).

Figure 8 illustrates the KCMS-VAN wallet during the *Merchant Registration* phase on both Nokia™ N95 (top) and Palm™ T|X devices (bottom). On the left screen, the KCMS wallet shows the client's details whereas the right screen shows the successful registration.

– **Payment Execution:** To make a payment to the merchant, the client is first prompted to enter the Order Description, the Product ID, the Price and the Type of Card to use (*Debit* or *Credit*). After filling the information, the KCMS-VAN wallet application generates the keys $KS_{C-M_i}$ and $KS_{C-I_z}$ based on the random numbers $i$ and $z$, respectively. Then, the client sends his/her nickname, the index $i$, and the transaction ID request (**TIDReq**). Upon receipt of the message, the merchant generates $KS_{C-M_i}$ (based on the $i$ value) and sends his/her identity $ID_M$) and the

```
public String AESDec(String mkey,String StToDec)
   {
       String resul = null;

       byte[] key    = Base64.decode(mkey.getBytes());
       byte[] input = Base64.decode(StToDec.getBytes());
       BlockCipher engineAES          = new AESLightEngine();
       BufferedBlockCipher cipherAES = new
              PaddedBlockCipher(new CBCBlockCipher(engineAES));
       cipherAES.init(false, new KeyParameter(key));
       byte[] cipherText =
              new byte[cipherAES.getOutputSize(input.length)];

       int outputLen =
       cipherAES.processBytes(input,0,input.length,cipherText,0);

       try
          {
              cipherAES.doFinal(cipherText, outputLen);
              resul = new String(cipherText);
          }
       catch (CryptoException ce)
          {
              resul = "";
          }
       return(resul);
   }
```

**Listing 4** Java code for *AESDec()*.

```
public String HmacMD5(String kyeenc, String st)
   {
       String resul = "";
       HMac hmac;

       hmac             = new HMac(new MD5Digest());
       byte[] macValue = new byte[hmac.getMacSize()];
       byte[] stbyt     = st.getBytes();
       hmac.init(new KeyParameter(Base64.decode(kyeenc)));
       hmac.update(stbyt, 0,stbyt.length);
       hmac.doFinal(macValue, 0);
       hmac.reset();
       resul = new String(Base64.encode(macValue));

       return resul;
   }
```

**Listing 5** Java code for *HmacMD5()*.

transaction ID (*TID*) to the client, encrypted with the session key $KS_{C-M_i}$ using *AESEnc()*.

The client then creates the *Value-Subtraction Request* (**VSRequest**) to be sent to the issuer. An authenticated hash of this message is computed using HMAC-MD5 algorithm with the key $KS_{C-I_z}$ by using the *HmacMD5()*. Note that the merchant will not able to decrypt or create the request since the merchant does not have the key $KS_{C-I}$ which is known only by the client and the issuer. The java code used for the *HmacMD5()* is presented Listing 5.

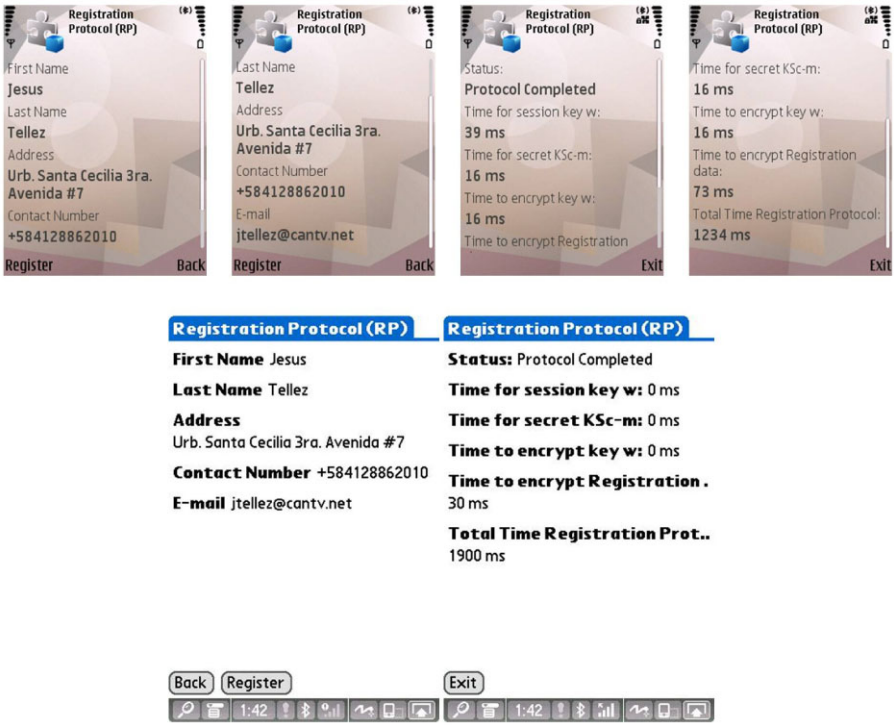


**Fig. 8** Screenshots of KCMS-VAN wallet during the execution of the merchant registration protocol phase on both Nokia™ N95 and ™ TlX devices

Once the **VSRequest** has been created, the client prepares the Payment Ordering request (called **PRequest**) to be sent to the merchant. This message includes **VSRequest**, the order information (**OI**), price, client nickname ($\mathbf{NID}_C$) and issuer identity ($\mathbf{ID}_I$). The encryption of *PRequest* with $KS_{C-M_i}$ using *AESEnc()* ensures its confidentiality.

Upon receipt of the request from the client, the merchant (who has the $KS_{C-M}$) decrypts the message using *AESDec()*. The merchant then combines the *VSRequest* received from the client and the amount payable by the client with the necessary information to create the *VCRequest*. This message is encrypted with $KS_{M-PG_k}$ before being sent to the Payment Gateway.

The Payment Gateway (PG) decrypts the message received using *AESDec()*. Then, PG sends the *VSRequest* to the issuer and the *VCRequest* to the acquirer. Upon receipt of the approval response, the PG prepares *VCResponse* (which includes *VSResponse* encrypted with $KS_{C-I_z}$) and encrypts it with $KS_{M-PG_k}$ using *AESEnc()*. The Payment Gateway then sends *VCResponse* to the merchant which in turn sends it to the client, encrypting $KS_{C-M_{i+1}}$ using *AESEnc()*. The message sent from the merchant to the client represents the Payment Ordering Response (*PResponse*). Note that although the flow information between the payment gateway, the issuer and the acquirer (step 4 of KCMS-VAN protocol) exists within a private banking network (beyond the scope of KCMS-VAN Protocol). We im-

**Table 4** Format of payment primitive transactions: *PRequest*, *VSRequest*, *VCRequest*, *VCResponse* and *VSResponse*

| Primitive | Fields | Size (bits) |
|---|---|---|
| Payment Request (*PRequest*) | *OI* | Variable |
| | Price | |
| | Client Data | |
| | Issuer Data | |
| | Timestamp | |
| | hmac(*VSRequest*) | 128 |
| Value-Subtraction Request (*VSRequest*) | Price | Variable |
| | Merchant Data | |
| | $h(OI)$ | 128 |
| Value-Subtraction Response (*VSResponse*) | $h(OI)$ | 128 |
| | Response (Yes/No) | Variable |
| Value-Claim Request (*VCRequest*) | VSRquest | Variable |
| | Price | |
| Value-Claim Response (*VCResponse*) | $h(OI)$ | 128 |
| | Response (Yes/No) | Variable |

plement the issuer and acquirer to have a better idea of the performance of the whole payment system even if it could be assumed that all transactions relevant to acquirer and issuer are successful within a limited time.

When the client receives the message, it retrieves the Payment Ordering Response using *AESDec()* to decrypt the message. The format of PRequest, VSRequest, VSResponse, VCRequest and VCResponse are shown in Table 4.

Figure 9 illustrates the KCMS-VAN wallet during the *Payment* phase on both Nokia™ N95 (top) and Palm™ T|X devices (bottom). On the left screen, the KCMS-VAN wallet shows the Order description whereas the right screen shows the successful payment.

## 5 Performance evaluation

### 5.1 Discussions of empirical results

#### *5.1.1 Execution time of merchant registration protocol*

In this section, we report on the empirical results obtained for our implementation of the KCMS-VAN protocol. In particular, we focus on the time taken to perform various parts of the KCMS-VAN protocol and the overall time taken to complete a payment transaction. The results were collected by performing 10 executions with different sets of data and the timings taken were done using the *getTime* method in the **Date** class of J2SE.

The average times required by the client and the merchant to perform the Merchant Registration Protocol are shown in Tables 5 and 6, respectively. Note that the time taken by the client to input the data has not been included into the time taken by

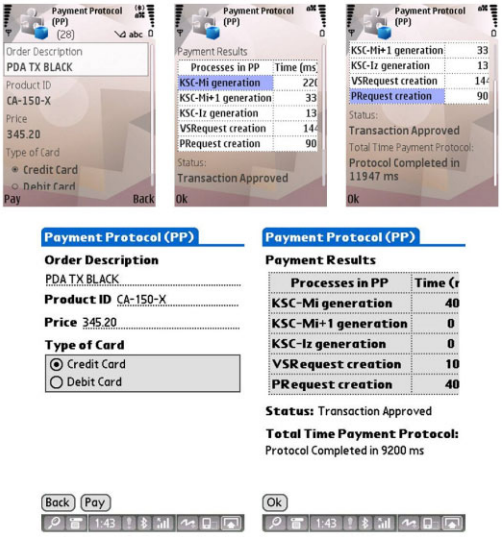


**Fig. 9** Screenshots of the payment phase on both Nokia™ N95 and Palm™ T|X devices

**Table 5** Time (ms) Taken at Client to perform the *Merchant Registration Protocol* (**TCRP**)

| Processes | Device | |
|---|---|---|
| | Nokia N95 | Palm T\|X |
| AESGen$_w$ | 39.90 | 1.00 |
| AESGen $KS_{c-m}$ | 28.80 | 1.00 |
| AESEnc Data | 81.90 | 5.10 |
| AESDec Data | 50.10 | 2.00 |
| **TCRP** | 1118.50 | 727.10 |

**Table 6** Time (ms) Taken at Merchant to perform the *Merchant Registration Protocol* (**TMRP**)

| Processes | Client's Device | |
|---|---|---|
| | Nokia N95 | Palm T\|X |
| AESEnc Data | $< 1$ | $< 1$ |
| AESDec Data | 6.40 | 5.90 |
| **TMRP** | 34.70 | 43.40 |

the application to perform the *Merchant Registration Protocol*. Moreover, when we tested our implementation by entering the maximum number of characters allowed into each individual field, we found that the maximum times taken by the client (1196 milliseconds for Nokia N95 and 740 milliseconds for the Palm T|X) and the merchant (35 milliseconds for Nokia N95 and 76 milliseconds for the Palm T|X) to perform the *Merchant Registration Protocol* are not much different from the average times calculated in Table 5 and Table 6.

**Table 7** Time (ms) taken at client (Nokia N95), merchant, payment gateway, and issuer on performing *Payment Protocol* (**TPP**)

| Processes | Nokia N95 | Merchant | PG | Issuer |
|---|---|---|---|---|
| $KS_{C-M_i}$ generation | 253.90 | 6.70 | – | – |
| $KS_{C-M_{i+1}}$ generation | 29.90 | $< 1$ | – | – |
| $KS_{C-I_z}$ generation | 15.00 | – | – | $< 1$ |
| $KS_{M-PG_k}$ generation | – | – | 6.80 | – |
| *VSRequest* creation | 145.30 | – | – | – |
| *PRequest* creation | 522.00 | – | – | – |
| *PRequest* decryption | 880.00 | – | – | – |
| *VCRequest* creation | 2.10 | – | – | – |
| *VCRequest* decryption | – | – | $< 1$ | – |
| *VSResponse* creation | – | – | – | 4.80 |
| **TPP** | 6842.20 | 2742.10 | 322.50 | 12.80 |

**Table 8** Time (ms) taken at client (Palm T|X), merchant, payment gateway, and issuer on performing *Payment Protocol* (**TPP**)

| Processes | Palm T\|X | Merchant | PG | Issuer |
|---|---|---|---|---|
| $KS_{C-M_i}$ generation | 66.00 | 5.70 | – | – |
| $KS_{C-M_{i+1}}$ generation | – | $< 1$ | – | – |
| $KS_{C-I_z}$ generation | – | – | – | $< 1$ |
| $KS_{M-PG_k}$ generation | – | – | 6.00 | – |
| *VSRequest* creation | 15.00 | – | – | – |
| *PRequest* creation | 653.00 | – | – | – |
| *PRequest* decryption | 1145.00 | – | – | – |
| *VCRequest* creation | 2.10 | 2.40 | – | – |
| *VCRequest* decryption | – | – | $< 1$ | – |
| *VSResponse* creation | – | – | – | 4.70 |
| **TPP** | 5657.00 | 2461.90 | 310.80 | 12.80 |

### *5.1.2 Execution time of payment protocol*

The average total time that the client has spent on performing the first transaction with the merchant (both the Merchant Registration Protocol and the Payment Protocol) was 7.96 seconds ($1.12 + 6.84 = 7.96$ seconds) using the Nokia N95 device compared to 6.39 seconds ($0.73 + 5.66 = 6.39$ seconds) when the palm t|x device was used. In next payment transactions, the total average time to complete each transaction will reduce to only 6.84 seconds on Nokia N95 device and 5.66 seconds on palm t|x device because the client does not have to execute the Merchant Registration Protocol. This minimal amount of time to complete a transaction reveals the potential of KCMS-VAN protocol to execute payment transactions in wireless environments with minimal delays.

**Table 9** KCMS-VAN wallet software size and its memory usage

| Device | Internal memory | Application size | % Memory used |
|---|---|---|---|
| Nokia™ N95 mobile phone | 163840 kb | 68 kb | 0.042 % |
| Palm™ T\|X handheld | 131072 kb | 132 kb | 0.100 % |

### *5.1.3 Application size*

as mobile devices have limited memory space, application size is an important issue which should be considered when developing applications for such devices. We focus our proposed payment protocol on the client's side because it is the only party in our scheme that uses a mobile device that interacts with the system.

Table 9 shows the KCMS-VAN wallet software size and its memory usage in the mobile devices used during our performance evaluation measurements.

## 6 Conclusions and further work

A protocol for secure on-line payments in a vehicle-to-roadside Restricted Scenario in VANETs (where the client cannot directly communicate with the issuer) was proposed in this research. Our protocol employs symmetric cryptographic techniques which allow clients to make purchases without disclosing private information and using a short range link (such as that provided by Bluetooth or Wi-Fi) to communicate with the merchant. Moreover, the chosen cryptography scheme has low computation requirements at both parties (since no public-key operation is required).

The empirical results from the performance evaluation results on the implementation have proven that payment transactions over a wireless network can be conducted by our proposed KCMS-VAN protocol. The chosen lightweight, secure cryptographic algorithms are suitable for our VANET application. Deploying such algorithms in KCMS-VAN results in the reduction of messages exchanged and computation costs at the client's mobile device. As we have seen from the implementation, a payment transaction by KCMS-VAN can be completed within average of 6.84 second using a Nokia N95 device and 5.66 second using palm T|X device.

The Client's KCMS-VAN wallet has a small size which makes it very suitable for memory-constrained mobile devices. The KCMS-VAN requires only 68 Kilobytes to be stored on a Nokia mobile phone and 123 kilobytes on a Palm Pilot 123 kb to be stored on a Palm Pilot.

In the future, we will explore the possibility to accommodate other cryptographic algorithms (such as elliptic-curve cryptography, digital signature scheme with message recovery using self-certified public keys) to our system. We also plain to reformulate and implement the proposed protocol into other restrictive connectivity scenarios in VANETs (such as a scenario where the merchant cannot communicate directly with the acquirer).

**Acknowledgements** We thank the anonymous reviewers for their useful comments that helped us to improve the quality and presentation of this paper. Sherali Zeadally was supported by an NSF award (No. 0911969) during this work whilst Jesús Isaac Téllez and José Cámara Sierra were supported in part by I-ASPECTS- Project (TIN2007-66107), but all ideas expressed represent the view of the authors.

## References


1. Abad Peiro, J. L., Asokan, N., Steiner, M., & Waidner, M. (1997). Designing a generic payment service. *IBM Systems Journal*, *37*(1), 72–88.
2. Asokan, N. (1994). Anonymity in Mobile computing environment. In *Workshop on mobile computing systems and applications* (pp. 200–2004).
3. Bella, G., & Bistarelli, S. (2005). Information assurance for security protocols. *Computers and Security*, *24*(4), 322–333.
4. Bellare, M. (2006). New proofs for NMAC and HMAC: security without collision-resistance. In *The 26th annual international cryptology conference (Crypto 2006)* (pp. 602–619).
5. Bellare, M., Garay, J., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G., Van Herreweghen, Els., & Waidner, M. (2000). Design, implementation and deployment of the *i*KP secure electronic payment system. *IEEE Journal on Selected Areas in Communication*, *18*(4), 611–627.
6. Certicom (2003). The next generation of cryptography. *Code and Cipher*: *Certicom's Bulletin of Security and Cryptography*, 1 (1).
7. Car2Car Communication Consortium (2007). *Overview of the C2C-CC system* (Technical Report version 1.0). Car2Car Communication Consortium.
8. Chari, S., Kermani, P., Smith, S., & Tassiulas, L. (2001). Security issues in M-commerce: a usage-based taxonomy. In *E-commerce agents* (pp. 264–282).
9. Hassinen, M., Hyppönen, K., & Haatajam, K. (2006). An open, PKI-based mobile payment system. In *Emerging trends in information and communication security, international conference (ETRICS'2006)* (pp. 86–100).
10. Hu, Z., Liu, Y., Hu, X., & Li, J. (2004). Anonymous micropayments authentication (AMA) in mobile data network. In *23rd Annual joint conference of the IEEE computer and communications societies (IEEE INFOCOM)* (pp. 46–53).
11. Hwang, R., Su, F., & Huang, L. (2007). Fast firmware implementation of RSA-like security protocol for mobile devices. *Wireless Personal Communications*, *42*(2), 213–223.
12. J. Hall, J., Kilbank, S., Barbeu, M., & Kranakis, E. (2001). WPP: a secure payment protocol for supporting credit-and debit-card transactions over wireless networks. In *International conference on telecommunications (ICT 2001)*.
13. Juntao, M. (2003). *Enterprise J2ME: developing mobile Java applications*. New York: Prentice Hall PTR.
14. Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: keyed-hashing for message authentication, RFC 2104.
15. Kungpisdan, S., Srinivasan, B., & Dunf Le, P. (2004). A secure account-based mobile payment protocol. In *International conference on information technology: coding and computing (ITCC'04)* (pp. 35–39).
16. Kungpisdan, S., Srinivasan, B., & Dung Le, P. (2003). Lightweight mobile credit-card payment protocol. In *4th International conference on cryptology in India (progress in cryptology—INDOCRYPT 2003)* (pp. 295–308).
17. Lei, Y., Chen, D., & Jiang, Z. (2004). Generating digital signatures on mobile devices. In *18th International conference on advanced information networking and applications (AINA'04)* (pp. 532–535).
18. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1997). *Handbook of applied cryptography*. Boca Raton: CRC Press.
19. NIST (1999). FIPS PUB 46-3 Data Encryption Standard (DES). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf.
20. NIST (2001). FIPS PUB 197 Advance Encryption Standard (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
21. Papadimitratos, P., Kung, A., Hubaux, J.-P., & Kargl, F. (2006). Privacy and identity management for vehicular communication systems: a position paper. In *Workshop on standards for privacy in user-centric identity management*.
22. Potlapally, N., Ravi, S., Raghunathan, A., & Jha, N. (2003). Analyzing the energy consumption of security protocols. In *2003 International symposium on low power electronics and design (ISLPED'03)* (pp. 30–35).
23. Ravi, S., Raghunalhan, A., & Potlapally, N. (2002). Securing wireless data: system architecture challenges. In *15th International symposium on system synthesis* (pp. 195–200).
24. Raya, M., & Hubaux, J.-P. (2005). The security of vehicular ad hoc networks. In *3rd ACM workshop on security of ad hoc and sensor networks (SASN'05)* (pp. 11–21).

25. Sanchez-Avila, C., & Sanchez-Reillol, R. (2001). The Rijndael block cipher (AES proposal): a comparison with DES. In *35th IEEE international Carnahan conference on security technology* (pp. 229–234).
26. Sukuvaara, T., & Pomalaza-RÃ¡ez, C. (2009). Vehicular networking pilot system for vehicle-to-infrastructure and vehicle-to-vehicle communications. *International Journal of Communication Networks and Information Security*, *1*(3), 1–10.
27. Téllez, J., & Sierra, J. (2007). A secure payment protocol for restricted connectivity scenarios in M-commerce. In *EC-Web* (pp. 1–10).
28. Téllez, J., & Sierra, J. (2007). Anonymous payment in a client centric model for digital ecosystems. In *Inaugural IEEE international conference on digital ecosystems and technologies (IEEE-DEST 2007)* (pp. 422–427).
29. Téllez, J., & Sierra, J. (2007). An anonymous account-based mobile payment protocol for a restricted connectivity scenario. In *DEXA workshops* (pp. 688–692).
30. Téllez, J., Sierra, J., Izquierdo, A., & Carbonell, M. (2006). Payment in a kiosk centric model with mobile and low computational power devices. In *Computational science and its applications (ICCSA 2006)* (Part V, pp. 798–807).
31. Téllez, J., Sierra, J., Izquierdo, A., & Torres, J. (2006). Anonymous payment in a Kiosk centric model using digital signature scheme with message recovery and low computational power devices. *Journal of Theoretical and Applied Electronic Commerce Research*, *1*(2), 1–11.
32. Téllez, J., Sierra, J., Zeadally, S., & Torres, J. (2008). A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks. *Computer Communications*, *31*(10), 2478–2484.
33. The Legion of the Bouncy Castle (2008). The Legion of the Bouncy Castle Java cryptography APIs version 1.4. http://www.bouncycastle.org/, 2008.
34. Sun Microsystem (2008). Java platform, micro edition (Java SE) v 1.6.0, API specification. http://java.sun.com/javase/index.jsp.
35. Sun Microsystem (2008). Java platform, micro edition (Java ME), API specification. http://java.sun.com/javame/index.jsp.
36. Wang, H., & Kranakis, E. (2003). Secure wireless payment protocol. In *International conference on wireless networks* (pp. 576–582).
37. Yousefi, S., Mousavi, M., & Fathy, M. (2006). Vehicular ad hoc networks (VANETs): challenges and perspectives. In *6th International conference on ITS telecommunications* (pp. 761–766).

**Jesús Téllez Isaac** is a System Engineer graduated at the Central Technological University (UNITEC), Venezuela, and he is a doctor candidate at the University Carlos III of Madrid in the Computer Science Department. He is an Associate Professor in the Computer Science Department of the University of Carabobo. He has served as a Technical Program Committee member for various international conferences. His research interests include Internet Security, performance evaluation of systems, mobile computing, mobile payment systems. *Jesús Téllez Isaac, Computer Science Department, FACYT, Universidad de Carabobo, Av. Universidad, Sector Bárbula, Valencia, Venezuela.*

**Sherali Zeadally** received his Bachelor's Degree in Computer Science from University of Cambridge, England, and the Doctoral Degree in Computer Science from University of Buckingham, England in 1996. He is an Associate Professor at the University of the District of Columbia. He currently serves on the Editorial Boards of over 15 international journals. He has been serving as a Co-Guest editor for over a dozen special issues of various peer-reviewed scholarly journals. He is a Fellow of the British Computer Society and a Fellow of the Institution of Engineering Technology, UK. His research interests include computer networks including wired and wireless networks, network and system security, mobile computing, ubiquitous computing, RFID, performance evaluation of systems and networks. *Sherali Zeadally, Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC, 20008, USA.*

**José Cámara Sierra** is an Associate Professor in the Computer Science Department of the University Carlos III of Madrid. He holds a Ph.D. in Computer Science and an M.Sc. in Business Administration. His research work focuses on Internet Security. He has participated in numerous research projects, and had published articles in various journals in the area of security. *José María Sierra Cámara Universidad Carlos III de Madrid, Department of Computer Science, Avda. de la Universidad, 30, Leganés (Madrid), 28911, España.*

Gmail by Google

**Jesús Augusto Téllez Isaac <jtellez@gmail.com>**

# FW: 2011 IET Premium Award - Communications - GREAT NEWS !

**Zeadally, Sherali <szeadally@udc.edu>** **Fri, Aug 19, 2011 at 12:56 PM**
To: "jtellez@gmail.com" <jtellez@gmail.com>

THE GREAT NEWS !!!!!

this should be a UNIVERSITY PRESS RELEASE at your University....

Make sure you send me a copy when your University releases the press release on this highly prestigious award....

Kind regards
Sherali

---

From: Deleay,Linda [LDeleay@theiet.org]
Sent: Friday, August 19, 2011 4:38 AM
To: Zeadally, Sherali
Subject: 2011 IET Premium Award - Communications

Dear S. Zeadally,
I have pleasure in confirming that you have been awarded the IET Premium Award for Communications for your paper, below:

Published in IET Communications
Received on 16th March 2009
Revised on 10th June 2009
doi: 10.1049/iet-com.2009.0191
In Special Issue on Vehicular Ad Hoc and Sensor Networks
ISSN 1751-8628
Security attacks and solutions for vehicular ad hoc networks
J.T. Isaac1 S. Zeadally2 J.S. Ca´mara3
1Computer Science Deparment (FACYT), Universidad de Carabobo, Avenida Universidad, Sector Ba´ rbula, Valencia, Venezuela
2Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC 2008, USA
3Computer Science Department, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, 28911 Legane´s, Madrid, Spain

Process
The editor-in-chief of each IET journal contacted the editorial board members inviting nominations for the best paper published in 2009 - 2010. This process started in January 2011. Any paper nominated by an editorial board member was short-listed. The editor-in-chief reviewed this shortlist and selected the best one, based on the recommendations of the board and their own subject expertise.

Prize
The award is worth £500. As the corresponding author, you will also receive a certificate. You may wish to divide the prize money with your co-authors, but the payment will be made directly to you. Please complete the attached Bank Details form and return it to me by email.

I am also pleased to invite you and a guest to attend the 2011 IET Ambition and Achievement Awards Ceremony at the Intercontinental Hotel<http://www.intercontinental.com/intercontinental/en/gb/locations/LONHB?sicreative=6119984843&dp=true&sicontent=0&sitrackingid=229087469&cm_mmc=Google-PS-IC_UK-_-G%20B-EMEA-Mkt-GBR-%5BE%5D-_-Exact-_-the%20intercontinental%20park%20lane&siclientid=1937>, Park Lane, London, on the afternoon of Wednesday, 9 November 2011. Registration will be held from 12.30pm, and the ceremony will commence at 1.00pm. The ceremony will close at approximately 5.00pm. Dress for the event is lounge suits.
The 2011 Awards Ceremony will celebrate the success of some of the profession's highest achievers – recognising excellence at all levels. During the event, you will have the opportunity to meet with other IET award winners, as well as guests from the engineering community and the sponsors of the 2011 Awards.
The certificate associated with your award will be presented during the Awards Ceremony. Please let me know if you are able to join us for the ceremony. If you are unable to attend, the certificate can be posted to you, in which case, please provide your postal address.
Please do not hesitate to contact me if you have any questions.

I look forward to hearing from you.

Best wishes,

Linda

Linda Deleay
Awards and Prizes Manager
The IET

www.theiet.org

T: +44 (0)1438 765694
M:+44 (0)7921 483625
Michael Faraday House, Six Hills Way, Stevenage, SG1 2AY, United Kingdom
P Please consider the environment before printing this email

---



**bank information form Premium Awards 2011.doc**
701K