INGENIERÍA Y DESARROLLO

ARTÍCULO CIENTÍFICO / RESEARCH ARTICLE

Reducing the computational cost of the authentication process in SET protocol

Reducción del coste computacional del proceso de autenticación en el protocolo SET

Cristina Satizábal*
Universidad Autónoma de Occidente (Colombia)
Rafael Martínez-Peláez**
Francisco J. Rico-Novella***
Jordi Forné***
Universidad Politécnica de Cataluña (España)

Grant and support information: This work has been supported by the Spanish public funded projects ARES (CONSOLIDERINGENIO-2010 CSD2007-00004) and ITACA (TSI2006-13409-C02-02), a graduate scholarship from CONACYT (Mexico) and the project "Modelo de Confianza PKI Aplicable a Entornos Distribuidos" from University of Pamplona (Colombia).

* Ph.D. in Telematics Engineering, Universidad Autónoma de Occidente (Colombia), Automatic and Electronic Department, Full Time Professor.
icsatizabal@uao.edu.co
Correspondence: Calle 45 N.° 86-47, apto. 201, torre 3, Cali (Colombia).

** Ph.D. Student, Universidad Politécnica de Cataluña (España), Department of Telematics Engineering, Researcher. *rafaelm@entel.upc.edu*

*** Ph.D. in Telecommunications Engineering, Universidad Politécnica de Cataluña (España), Department of Telematics Engineering, Associate Professor.
f.rico@entel.upc.edu; *jforne@entel.upc.edu*

Número 27
Enero-junio, 2010
ISSN: 0122-3461

Abstract

SET is a secure credit card payment protocol that provides a robust security model based on data integrity, data confidentiality and mutual authentication to deliver personal and financial information through Internet. However, the parties involved in the transaction must carry out a lot of cryptographic operations which can be a problem when these parties use mobile devices with low processing and storage capacities. This paper shows how the computational cost of the SET protocol can be reduced when another protocol called TRUTHC is used in conjunction with the Public Key Infrastructure (PKI). Results show that the total execution time can be reduced about 3% using TRUTHC from the customer point of view. This reduction is still the same in spite of the increase of path length.

Keywords: Bluetooth technology, certification path validation, computational cost, mobile payment, SET protocol, TRUTHC.

Resumen

SET es un protocolo seguro de pago, con tarjeta de crédito, que proporciona un modelo robusto de seguridad para entregar información personal y financiera a través de Internet, basado en la integridad de los datos, su confidencialidad y la autenticación mutua. Sin embargo, las partes involucradas en una transacción deben llevar a cabo diversas operaciones criptográficas, lo que puede ser un problema cuando se usan dispositivos móviles con baja capacidad de almacenamiento y procesamiento. Este artículo muestra como se puede reducir el coste computacional de SET, mediante el uso de otro protocolo llamado TRUTHC en conjunto con una Infraestructura de Clave Pública (PKI). Los resultados muestran que, usando TRUTHC, el tiempo total de ejecución puede ser reducido un 3% desde el punto de vista del cliente. Esta reducción se mantiene aunque aumente la longitud del camino de certificación.

Palabras clave: Coste computacional, pago móvil, protocolo SET, tecnología bluetooth, TRUTHC, validación de caminos de certificación.

Fecha de recepción: 21 de octubre de 2009
Fecha de aceptación: 19 de enero de 2010

1. INTRODUCTION

In mobile payment systems, customers can pay for products and services anywhere and anytime with the comfort offered by their mobile devices. In this scenario, security is important because transactions are achieved without any physical contact between the customer and merchant. Also, payment information is sent through an open environment, and somebody in the coverage area, with the appropriate equipment, can hear the communication and modify information. Thus, participants can be victims

of fraud. The use of certificates avoids the repudiation of a transaction, guarantees the integrity and origin of data through digital signatures, and allows establishing a secure channel for payments.

SET is a secure credit card payment protocol that involves the use of certificates and cryptographic operations to protect the information exchanged among the participants in the payment transaction. However, these cryptographic operations demand mobile devices with high processing and storage capacities in order to carry out the whole protocol.

Certification path validation increases complexity of SET protocol. Satizábal et al. [1] proposed TRUTHC (Trust Relationship Using Two Hash Chains) to improve the efficiency of certification path validation process, since this mechanism establishes an alternative trust relationship among different entities of a hierarchical PKI (Public Key Infrastructure) using hash chains instead of signature verification operations. In a previous work, it was evaluated the performance of TRUTHC in WTLS protocol using a general mobile payment scenario and compared it with a typical PKI [2].

In this paper, the computational cost of the customer and merchant in SET protocol is reduced using TRUTHC. In addition, the transmission time and the whole execution time of the protocol are calculated. Section 2 describes mobile payment security. Section 3 shows the characteristics of Bluetooth. Section 4 explains the operation of SET protocol. Section 5 defines WPKI, certification path validation and hierarchical architectures. Section 6 describes the operation of TRUTHC. Section 7 presents the calculation of the computational cost, the transmission time and the whole execution time of SET protocol. In addition, the whole execution time of a typical PKI is compared with the time of a PKI with TRUTHC. Finally, section 8 presents the conclusion.

2. MOBILE PAYMENT SECURITY

Mobile payment is defined as the process of exchanging financial values between two parties (customer and merchant) using a mobile device to pay for products or services [3], [4].

Mobile payment protocols must offer robust security because the financial data are sent over wireless networks. In this sense, customers and merchants require mutual authentication, payment authorization, confidentiality, integrity and non-repudiation [5], as follows:

- *Authentication*: mobile payment systems must offer the option to authenticate each entity (mutual authentication).
- *Authorization*: mobile payment systems should request confirmation of the payment.
- *Integrity*: mobile payment systems must guarantee that the messages have not been modified.
- *Non-repudiation*: mobile payment systems should avoid refuting payments.
- *Privacy*: mobile payment systems must avoid eavesdroppers to have access to the messages.

3. BLUETOOTH TECHNOLOGY

Bluetooth is a wireless technology to interconnect mobile devices to each other, or with other devices via point-to-many or point-to-point communications. This technology transfers voice, data, and video in real time. The transmission area is omnidirectional and its transfer rate is 1Mbps. The maximum distance between the data origin (source) and the receiver is around 10m. Bluetooth technology is a key issue in mobile commerce because it enables mobile devices to pay for goods or services [6], [7].

Baseband layer

Bluetooth technology uses two types of links to establish a connection among devices: Synchronous Connection Oriented (SCO) and Asynchronous Connectionless Link (ACL). SCO link establishes a point-to-point connection and it is a symmetric dedicated link between two devices. On the other hand, ACL link establishes a point-to-multipoint connection and it is an asynchronous link among all the devices. The first type of link is a circuit switched connection between the master and slave, while ACL link is a packet switched connection among the master and all the slaves.

SCO link guarantees the delay and bandwidth to transmit an average quality of voice and music by the use of the Link Management Protocol (LMP). LMP performs the link configuration such as quality of service (QoS) [7].

On the other hand, there are two different ACL link packets (frames): 1) DMX, where the payload is encoded and 2) DHX, where the payload is unprotected. The value of ´X´ stands for the number of slots required to transmit the frame. DMX types are DM1, DM3 and DM5, which includes Forward Error Correction (FEC), Cyclic Redundancy Check (CRC) code and Automatic Repeat reQuest (ARQ). Table 1 summarizes the DMX and DHX link packet types' characteristics [6].

Table 1. Characteristics of ACL link packets

					Asymmetric max rate (Kbps)	
Type	**User Payload (bytes)**	**FEC**	**CRC**	**Symmetric max rate (Kbps)**	**Forward**	**Reverse**
DM1	0-17	2/3	YES	108.8	108.8	108.8
DM3	0-121	2/3	Yes	258.1	387.2	54.4
DM5	0-224	2/3	Yes	286.7	477.8	36.3
DH1	0-27	no	Yes	172.8	172.8	172.8
DH3	0-183	no	Yes	390.4	585.6	86.4
DH5	0-339	no	Yes	433.9	723.2	185.6

Physical layer

Bluetooth technology transmits and receives on the frequency band of 2.4 GHz. The band is divided into 79 MHz wide channels that are spaced 1 MHz. This layer utilizes Frequency Hopping Spread Spectrum (FHSS) as technique of transmission. FHSS can reduce the impact of jamming and interference caused by other systems. The transmission channel changes 1600 times per second. Bluetooth technology uses a slotted Time Division Duplex (TDD) scheme for duplex transmission, where each time slot is

625μs. Each slot corresponds to a different transmission (Figure 1). The master uses the even numbered time slots to transmit, while the slaves use the odd numbered time slots [6].

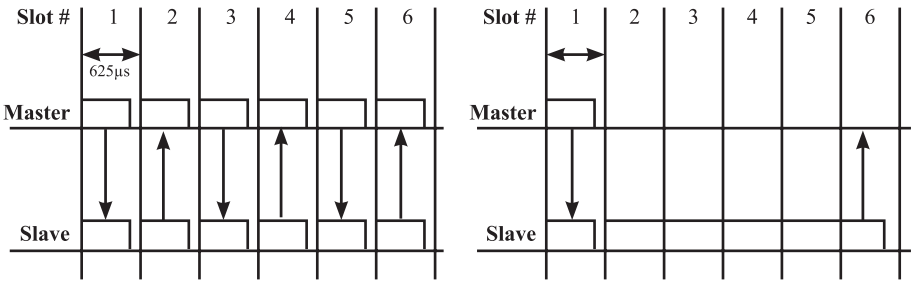

Figure 1. Transmissions of single-slot and multi-slot link packets

4. SECURE ELECTRONIC TRANSACTION (SET)

Financial institutions developed a secure credit card payment protocol over open networks, called SET [8]. This protocol provides a robust security model based on data integrity, data confidentiality and mutual authentication to deliver personal and financial information through Internet. For that reason, customer C and merchant M carry out several cryptographic operations that require CPU execution time and power consumption. Moreover, the transmission of the information increases the time of the payment transaction.

The number of cryptographic operations carried out by each entity during the transactions of SET protocol are summarized in Table 2. The process is as follows:

1. M discovers and associates the customer's device.
2. C initializes the protocol sending an Initial Request message to the merchant.
3. M sends to C his/her certificate ($CERT_M$), payment gateway's certificate ($CERT_{PG}$), and the TID (unique identifier of the transaction) signed with his/her private key ($\{TID\}K_M^{-1}$).

4. C verifies the certification path of M and the Payment Gateway (PG).

5. C sends to M the OI (Order Information); the PI (Payment Information); the hash value of the order information (H(OI)) and the hash of the payment information (H(PI)) encrypted with the symmetric key k1; k1 signed with his/her private key ($\{k1\}K_C^{-1}$); the hash of OI and PI encrypted with the symmetric key k2; k2 encrypted with the payment gateway's public key ($\{k2\}K_{PG}$); and his/her digital certificate ($CERT_C$). To obtain TID, C must carry out a verification operation with the public key of M ($K_{M)}$.

6. M verifies the certification path of C.

7. M verifies the integrity of the OI and PI. For that reason, M must carry out a verification operation with the public key of C (K_C) to obtain k1. Then, M uses k1 to decrypt (H(OI),H(PI)), makes a hash over OI and another hash over PI, and compares them with the decrypted values. In addition, M adds TID, Price and Date to the message, and signs the message with his/her private key (K_M^{-1}). Then, M encrypts the message with the PG public key (K_{PG}) and forwards it to PG.

8. PG decrypts the message using his/her private key (K_{PG}^{-1}) and verifies the integrity of OI and PI. To do this process, PG must carry out a verification operation with the public key of M (K_M). Then, PG must use its private key (K_{PG}^{-1}) to decrypt k2 and the public key of C (K_C) to verify the signature over k1. With these symmetric keys, PG decrypts (H(OI), PI) and (H(OI), H(PI)). Later, PG calculates a hash over OI and another over PI, and compares the results with the received values. Later, PG approves or rejects the transaction and sends the response message to M signed with its private key (K_{PG}^{-1}) and encrypted with the public key of M (K_M).

9. M decrypts the message and verifies the signature. M forwards the response signed with his/her private key (K_M^{-1}) to C. Finally, C verifies the signature and knows the status of the transaction.

Table 2. Cryptographic operations without authentication in SET protocol

Cryptographic operation	Customer	Merchant	PG
Encryption	1	1	1
Decryption	0	1	2
Signature creation	1	3	1
Signature verification	2	2	2
Symmetric encryption	2	0	0
Symmetric decryption	0	1	2
Hash function	2	2	2

5. WAP PUBLIC KEY INFRASTRUCTURE (WPKI)

WPKI [9] is an optimised extension of the typical PKI (Public Key Infrastructure) [10] for the wireless environment.

WPKI requires the same components used in a typical PKI: Certification Authorities (CAs), Registration Authorities (RAs), End Entities (EEs), PKI Directories (DIRs); and adds a new component, called PKI Portal. The PKI Portal is responsible for translating requests made by the WAP client to the RA and CA in the PKI. The PKI Portal will typically embed the RA functions and interoperate with the WAP devices on the wireless network and the CAs on the wired network (Figure 2).

The general model adopted in the current version of WPKI is, according to [9]:

- WTLS server and root CA certificates stored in the device will be according to WTLS certificate defined in [9].
- Client (WTLS and application) and root CA certificates stored in servers will be according to X.509 as profiled in [11].

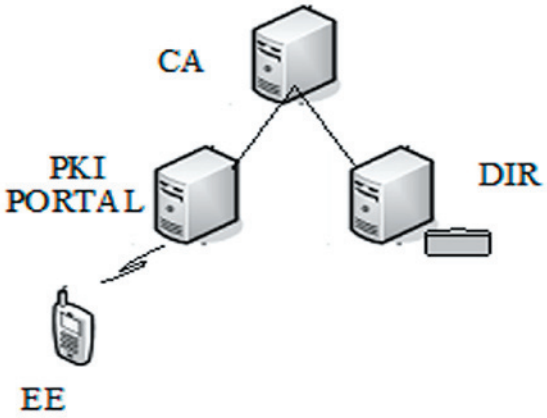

Figure 2. WPKI architecture

- Client (WTLS and application) and root CA certificates which must be sent over the air and/or stored in WAP client devices will be according to X.509 as profiled in [9].
- Storage of the certificate URL in the device, rather than the full client certificate, is the preferred model when X.509 format certificates would otherwise be expected to be transferred over the air. Storage of X.509 client certificates in the device is expected to be the exception, unless they are provisioned on the device through a WIM (Wireless Identity Module) [12].

Certification path validation

A certification path is a chain of public key certificates through which a user can obtain the public key of another one. The primary goal of the path validation is to verify the binding between the client and his/her public key. A trust anchor is the CA verification key used by the client application as the starting point for all certificate validation. The certification path length is equal to the number of certificates in the path that is the number of CAs in the path plus one. Since, the verifier knows and trusts the public key of his/her trust anchor, the trust anchor's certificate is not included in the path [13].

In general, the path validation process involves the following steps:

- *Discovering a certification path*: It is to set up a trusted path between the verifier's trust anchor and the target entity based on the trust relationship among the entities of the PKI.
- *Retrieving the certificates*: It is to retrieve each certificate in the path from the directories where they are stored.
- *Verifying the digital signatures*: It is to verify the validity of the digital signature of each certificate in the path. It involves:

    a. Decrypting the signed part of the certificate with its issuer's public key.

    b. Computing the hash of the certificate's content.

    c. If the result of steps a and b are the same then the signature is valid.

- *Verifying the validity of the certificates*: It is to determine if the certificates are expired or revoked. The certificates validity period is used to verify the expiration, while the revocation status depends on the revocation mechanism.

Hierarchical architecture

In a hierarchical architecture, all the users trust the same root CA (RCA), that is, all the users of a hierarchical PKI begin certification paths with the RCA public key [14]. In general, the root CA does not issue certificates to users but only issues certificates to subordinate CAs. Each subordinate CA may issue certificates to users or another level of subordinate CAs, if it is permitted by policies.

The certification paths are easy to build in a hierarchical PKI because the trust relationships are unidirectional and the longest path is equal to the depth of the tree plus one: a CA certificate for each subordinate CA plus the user's certificate.

6. TRUTHC

TRUTHC [1] establishes an alternative trust relationship among the entities of a hierarchical PKI through two hash chains: one links the secret seeds of the certification authorities and the other links the certificates of each path. This replaces the verification operations of a path validation process by hash operations, which reduces the computational cost from the verifier's point of view. Table 3 shows the notation used in TRUTHC.

Hash chain

A hash chain [15] is a list of values $y_1, y_2, ..., y_m$ linked together cryptographically, where m is the length of the chain. These chains are created by recursively computing a hash function H over a random seed x:

$$y_1 = H(x),\ y_2 = H(y_1),\ ...,\ y_m = H(y_{m-1})$$

Table 3. Notations

Notation	Meaning
K_X	Public key of X
K_X^{-1}	Private key of X
$H(I)$	Hash of information I
$CERT_X$	Certificate of entity X
Cnt_X	Content of $CERT_X$
Sig_X	Signature over Cnt_X
s_{RCA}	Random secret seed of the root CA
n_X	Secret seed of authority X
N_X	Encapsulated seed of authority X
L	Certification path length.
h_X	Integrity check value associated with authority X
SN_X	Serial Number of the certificate $CERT_X$

A hash function H is a transformation that takes a variable-size input x and returns a fixed-size string, which is called the hash value. H must be a one-way function, that is to say:

- Given a value x, it is easy to compute $H(x)$
- Given a value y, it is not feasible to obtain a value x such that $y = H(x)$

Thus, given a value y_i of the chain, it is unfeasible to compute the previous values. In addition, H can be collision-free, which means that it is computationally unfeasible to find any pair (x, z) such that $H(x) = H(z)$.

Issuing certificates

TRUTHC extends the typical certificate issuing process in a hierarchical architecture. The chaining relation, encapsulated seed, and protocol are:

Chaining Relation

$$n_{RCA} = H(s_{RCA})$$

$$n_{CA1} = H(n_{RCA}, SN_{CA1})$$

$$n_{CAi} = H(n_{CAi-1}, SN_{CAi}),\ 2 \le i \le L - 1$$

$$h_{CA1} = H(n_{RCA}, Cnt_{CA1})$$

$$h_{CAi} = H(h_{CAi-1}, n_{CAi-1}, Cnt_{CAi}),\ 2 \le i \le L - 1$$

$$h_U = H(h_{CAL-1}, n_{CAL-1}, Cnt_U)$$

Encapsulated Seed

$$N_{CAi} = \{n_{CAi}\}K_{CAi},\ 1 \le i \le L - 1$$

Protocol

$$CA_i \rightarrow CA_{i+1}: CERT_{RCA}, CERT_{CAi+1}, h_{CAi+1}, N_{CAi+1}$$

$$CA_{L-1} \rightarrow U: CERT_{RCA}, CERT_U, h_U$$

It is assumed that the hash function *H* is collision-free. Figure 3 shows the chaining relation of the certificates.

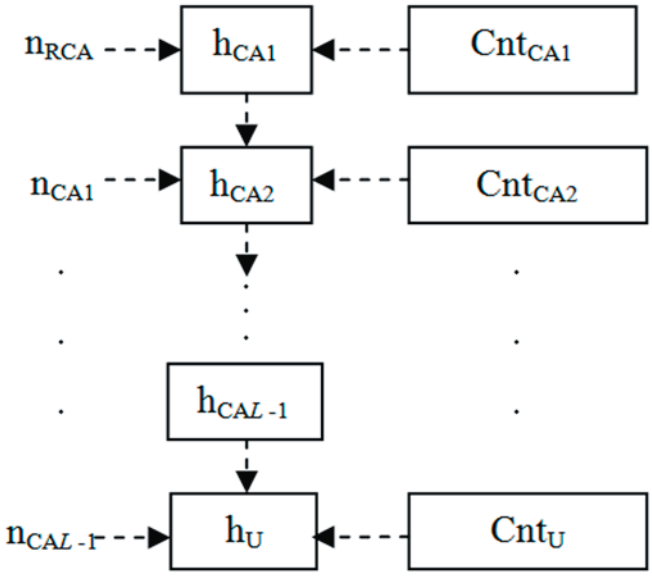

Figure 3. Chaining relation

Verifying certificates

TRUTHC includes a new TTP (Trusted Third Party) to the PKI called Verification Authority (VA). VA verifies the integrity of the certificates and the trust relationship among the entities of a certification path.

RCA issues the certificate $CERT_{VA}$, and sends to VA: the trust anchor´s certificate $CERT_{RCA}$, the certificate $CERT_{VA}$ and the secret seed n_{RCA} encrypted with the public key of VA, so that only VA can decrypt it. However, functions of VA can be carried out by RCA, since this authority can compute the seeds of its subordinated CAs.

Encapsulated Seed

$$N_{VA} = \{n_{RCA}\}K_{VA}$$

Protocol

$$CA \rightarrow VA: CERT_{RCA}, CERT_{VA}, N_{VA}$$

If user V wants to verify the signature of a message sent by user U, he/she needs to carry out the following steps:

- User V retrieves $CERT_U$ and h_U.
- User V sends VA: $CERT_U$.
- VA retrieves the other certificates of the certification path: $CERT_{CA1}$, $CERT_{CA2}$, ..., $CERT_{CAL-1}$.
- VA computes h'_U by using equations of the chaining relation in previous subsection and the secret seed n_{RCA}. Then, it returns h'_U in a signed response to user V.
- User V verifies the signature of the VA response. This implies to verify the signature of the VA certificate with PK_{RCA} and then the signature of the VA response.
- V compares h_U and h'_U. If h_U and h'_U are the same, user V can trust the integrity of the certificate $CERT_U$ and that it is part of the certification path of user U.
- User V verifies the signature of the message using the public key of user U, K_U, obtained from $CERT_U$.

7. EXECUTION TIME OF SET PROTOCOL

In this section, the efficiency of TRUTHC is evaluated, comparing the computational and communication cost of a typical PKI with the cost of a PKI with TRUTHC.

Scenario

A hierarchical PKI that involves the following entities is supposed (Figure 4):

- *Root Certification Authority (RCA)*: It could be a national or international certification authority, such as VeriSign.

- *Credit Card Certification Authority (CCCA)*: It could be an international credit card company such as VISA or MasterCard. CAs issue certificates to PGAs.

- *Payment Gateway Authority (PGA)*: It is the CA of some e-commerce application service provider. A PGA acts like mediator between the customer and merchant and issues certificates to users (customers and merchants).

- *Customer (C)*: Is a user that wants to obtain some object or service from a merchant.

- *Merchant (M)*: Is a user that offers its products or services to the customers.

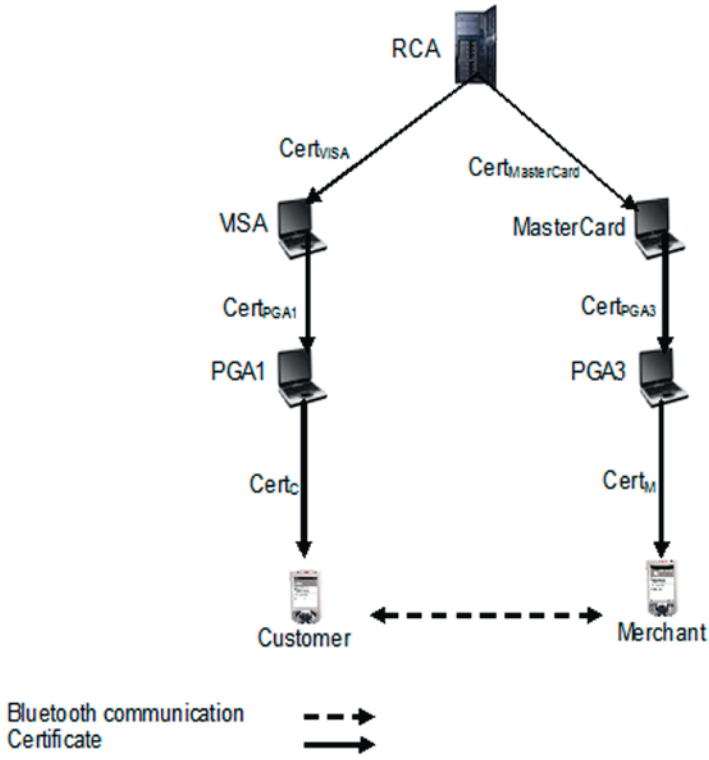

Figure 4. PKI structure

Computational cost

- *Number of cryptographic operations*

Table 2 shows the number of cryptographic operations carried out during SET protocol, without considering the operations of the mutual-authentication process.

When customer C carries out SET protocol with the merchant M (Figure 4), it must verify the certification path of M and PGA3 (step 4), and the merchant verifies the certification path of C (step 6). Thus, they carry out a mutual-authentication.

In a typical PKI, customer and merchant must obtain the certificates of the CCCA, the PGA and the user that are part of the path, and carry out three hash and three signature verification operations. Table 4 shows the number of cryptographic operations carried out by each entity during the whole SET protocol in a typical PKI.

Table 4. Cryptographic operations with authentication of SET protocol in a typical PKI

Cryptographic operation	Customer	Merchant	RCA	PGA3
Encryption	1	1	0	1
Decryption	0	1	0	2
Signature creation	1	3	0	1
Signature verification	5	5	0	2
Symmetric encryption	2	0	0	0
Symmetric decryption	0	1	0	2
Hash function	5	5	0	2

On the other hand, in a PKI with TRUTHC, it is supposed that RCA carries out the functions of VA. Therefore, RCA must calculate two h'_U values, carry out hash operations and sign the messages that must be sent to customer and merchant with these values. Then, C and M must verify the signature

of the response received from RCA. The number of cryptographic operations carried out by the customer, merchant and RCA in a PKI with TRUTHC during the mutual-authentication process are shown in Table 5. Table 6 shows the number of cryptographic operations carried out by each entity during the whole SET protocol in a PKI with TRUTHC.

Table 5. Cryptographic operations during authentication using TRUTHC

Entity	OP_{SIG}	OP_{VER}	OP_{HASH}
Customer	0	1	1
Merchant	0	1	1
RCA	2	0	10

Table 6. Cryptographic operations with authentication of SET protocol using TRUTHC

Cryptographic operation	Customer	Merchant	RCA	PGA3
Encryption	1	1	0	1
Decryption	0	1	0	2
Signature creation	1	3	2	1
Signature verification	3	3	0	2
Symmetric encryption	2	0	0	0
Symmetric decryption	0	1	0	2
Hash function	3	3	10	2

- ***Evaluation of Computational Cost***

For the implementation of the cryptographic operations in each device, Python 2.4 with Crypto library is employed. All the entities use RSA public key algorithm with a key size of 1024 bits.

The following devices are considered: PDA with 200MHz ARM920T processor and 64MB, running Linux 2.4 operating system, laptop with 800MHZ AMD Turion processor and 2GB, running Linux, and PC with 3GHz Pentium IV

and 1 GB, running Linux. It is assumed that the customer and merchant use PDAs, TTPs (e.g. bank or payment gateway) use laptops, and RCA use a PC.

The execution times obtained with the PDA are: encryption/decryption operations performed using RSA algorithm take 0.0263s and 1.8990s, respectively; a signature creation requires 1.8973s and signature verification requires 0.0263s; encryption/decryption operations using DES algorithm take 0.0010s. It is used SHA-2 as hash function and its execution time is 0.0006s.

In the laptop, the execution time obtained for encryption/decryption operations using RSA algorithm is 0.0004s and 0.0036s, and the execution time of signature creation/verification operations is 0.0014s and 0.0004s, respectively. The encryption/decryption operations using DES algorithm requires 0.00001s. It is used SHA-2 as hash function and its execution time is 0.00001s.

In the PC, the execution time obtained for encryption/decryption operations using RSA algorithm is 0.0016s and 0.000007s, and the execution time of signature creation/verification operations is 0.0015s and 0.0016s, respectively. The encryption/decryption operations using DES algorithm requires 0.000007s. The hash execution time using SHA-2 is 0.00001s.

Equation (1) is used to compute the computational cost (COST) of the cryptographic operations carried out by the customer, merchant and trusted third parties in the mobile payment protocol. It is denoted the number of public key encryption/decryption operations with OP_{ENC} and OP_{DEC} respectively, the number of signature/verification operations with OP_{SIG} and OP_{VER}, the number of symmetric key encryption/decryption operations with OP_{SYM} and the number of hash operations with OP_{HASH}. The execution time for each operation is defined with T_X where ´X´ denotes the type of operation.

$$\begin{aligned} COST = (OP_{ENC}*T_{ENC}) + (OP_{DEC}*T_{DEC}) + (OP_{SIG}*T_{SIG}) + \\ (OP_{VER}*T_{VER}) + (OP_{SYM}*T_{SYM}) + (OP_{HASH}*T_{HASH}) \end{aligned} \quad (1)$$

Table 7 shows the computational cost per entity in a typical PKI and PKI with TRUTHC.

Table 7. Computational cost per entity (COST)

	COST (s)	
	Typical PKI	**PKI with TRUTHC**
Customer	2.0601	2.0053
Merchant	7.7527	7.6989
PGA3	0.0098	0.0098
RCA	0	0.0031

Transmission time

- ***Link packet characterization***

A realistic characterization of the traffic is quite difficult due to the lack of an exact knowledge of data's length transmitted during the protocol operation. For that reason, it is set the length of data transmitted among different entities in each step of SET protocol.

It was decided to use DH5 link packets in the analysis. The maximum size of the DH5 payload is 339 bytes and each ACL link packet fits into 5 slots. When a link packet arrives to the Bluetooth baseband layer its payload consists of three parts: 1) an IP header with 20 bytes; 2) a TCP header with 32 bytes; 3) a data with variable length. In addition, L2CAP (Logical Link Control and Adaptation Protocol) adds 4 bytes for channel identification and link packet length [6], [7]. The format of ACL link packets is shown in Figure 5. Therefore, the maximum length of data is 283 bytes.

It is assumed an ideal channel without link packet lost and a multi-slot packet transmission.

In order to initialize a new communication, the master device uses the inquiry procedure (Inq_p) and page scheme (Pag_s) to discover and establish a new communication with slave devices. The average time for the inquiry phase is 0.71s and 0.64s for the page phase [7].

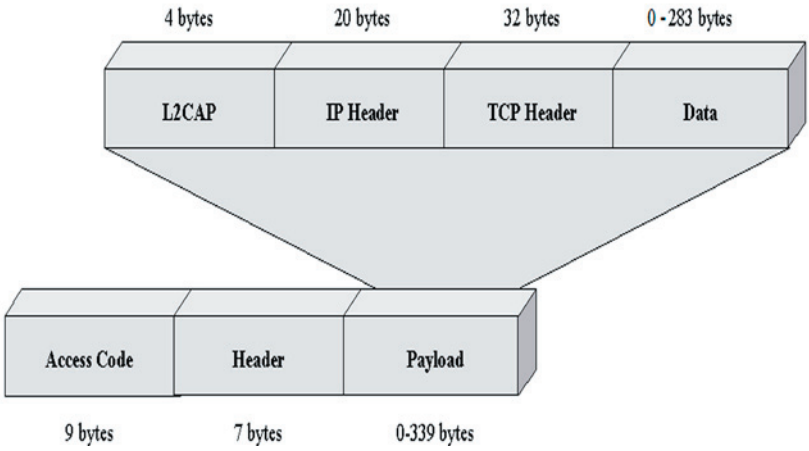

Figure 5. ACL link packet structure

It is introduced some notation adopted in this paper. For the total size of the data is used *Lp* and U denotes the maximum size of data field of each DH5 link packet (283 bytes). In addition, N_{DH5} is used as the number of ACL link packets required to transmit the data, calculated using equation (2) :

$$N_{DH5} = Lp / U \tag{2}$$

Furthermore, the total number of slots used to transmit the ACL link packets is indicated with S_T, taking into account the empty slots. This is calculated using equation (3); *DH5* is the number of slots required to transmit a DH5 link packet (5 slots), that is a multi-slot packet.

$$S_T = (N_{DH5} * DH5) + N_{DH5} - 1 \tag{3}$$

Now, the delay (*D*) caused by the information processing can be determined on: TCP layer (TCP_D), IP layer (IP_D), L2CAP layer ($L2CAP_D$) and Baseband layer ($Base_D$) using equation (4). Their values are 1µs, 1µs, 1ms and 1ms, respectively, according to [16].

$$D = TCP_D + IP_D + L2CAP_D + Base_D \quad (4)$$

The whole transmission time (T_T) to transmit each message is calculated using equation (5), where T_{SLOT} is the duration of each slot (625μs).

$$T_T = (S_T * T_{SLOT}) + D \quad (5)$$

Table 8 summarizes the transmission time of each message exchanged among the entities in SET protocol.

Table 8. Transmission time per message

Message	*Lp* (bytes)	ACL link packets N_{DH5}	Slots S_T	T_T (s)	Whole time (s)
1 Inq_p & Pag_s	-	-	-	1.35	1.691
2 C to M	96	1	5	0.0051	
3 M to C	11392	41	245	0.1551	
4 C	500	2	11	0.0089	
5 C to M	8060	29	173	0.1101	
6 M	500	2	11	0.0089	
7 M to PG	2524	9	53	0.0351	
8 PGA3 to M	424	2	11	0.0089	
9 M to C	344	2	11	0.0089	

Total execution time

In this section, the total cost required by all the entities to complete SET protocol is calculated, from the addition of the computational cost and the transmission time, determined in the previous sub-sections. Thus, the total execution time required is equal to $COST_T + T_T$. Table 9 shows the total execution time for each entity to carry out the SET protocol including the authentication process in a typical PKI and a PKI with TRUTHC.

Table 9. Transmission time per entity

Message	Typical PKI			PKI with TRUTHC		
	T_T (s)	COST (s)	Total execution time	T_T (s)	COST (s)	Total execution time
C	0.1241	2.0601	2.1842	0.1241	2.0053	2.1294
M	1.5580	7.7527	9.3107	1.5580	7.6989	9.2569
PGA3	0.0089	0.0098	0.0187	0.0089	0.0098	0.0187
RCA	0	0	0	0	0.0031	0.0031

8. CONCLUSIONS

SET is a complex payment protocol that uses certificates to carry out the mutual authentication process between customer and merchant. This increases the number of cryptographic operations that entities must carry out and therefore the computational cost of the protocol.

This paper presents a comparison between a typical PKI and a PKI with TRUTHC (Trust Relationship Using Two Hash Chains) in a mobile payment scenario. The computational, transmission and whole execution time of SET protocol is calculated. According to the obtained results, TRUTHC reduces the computational cost 2,66% for customer and 0,66% for merchant using RSA algorithm. On the other hand, the total execution time is reduced 2,51% for customer and 0,58% for merchant. This reduction implies a slight increase in the computational cost of RCA (0,0031s), but this is not a big problem, since RCA has more processing capacity than customer and merchant. In addition, if certification path length increases, the total execution time for costumer and merchant is the same in a PKI with TRUTHC compared with a typical PKI where an increase in path length implies more cryptographic operations for customer and merchant. The computational cost of RCA increases with path length but this entity carries out only hash operations where computational cost is low.

Acknowledgements

This work has been supported by LOGOS Research Group of University of Pamplona (Colombia)

References

[1] C. Satizábal, R. Páez, and J. Forné, "PKI trust relationship using hash chains", presented at International Conference on Advances in the Internet, Processing, Systems and Interdisciplinary Research (IPSI'05), France, 2005.

[2] C. Satizabal, R. Martínez-Peláez, J. Forné, and F. Rico-Novella, "Reducing the computational cost of certification path validation in mobile payment", *Lecture Notes in Computer Science*, vol. 4582, pp. 280-296, 2007.

[3] J. Gao, K. Edunuru, J. Cai, and S. Shim, "P2P-Paid: A peer-to-peer wireless payment system", presented at Second IEEE International Workshop on Mobile Commerce and Services (WMCS'05), Germany, 2005.

[4] S. Nambiar, C.-T. Lu, and L.-R. Liang, "Analysis of payment transaction security in mobile commerce", presented at IEEE International Conference on Information Reuse and Integration (IRI'04), USA, 2004.

[5] H. v. d. Heijden, "Factors affecting the successful introduction of mobile payment system", presented at 15th Bled Electronic Commerce Conference eReality: Constructing the eEconomy, Slovenia, 2002.

[6] R. Bruno, M. Conti, and E. Gregori, "Bluetooth: Architecture, protocols and scheduling algorithms", *Cluster Computing*, vol. 5, pp. 117-131, 2002.

[7] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison", *IEEE Wireless Communications*, vol. 12, pp. 12-26, 2005.

[8] Mastercard and VISA, "SET Secure Electronic Transaction specification", *Business Description 1.0*, 1997.

[9] Wireless Application Protocol Forum WAPForum, "Wireless Application Protocol Public Key Infrastructure Specification", Open Mobile Alliance WAP-210-WAPArch-20010712-a, 2001.

[10] International Telecommunication Union ITU-T, "Recommendation X.509: Information Processing Systems - Open Systems Interconnection - The Directory : Authentication Framework (Technical Corrigendum)", 2000.

[11] R. Housley, W. Polk, W. Ford, and D. Solo, "RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile", IETF Network Working Group, 1999.

[12] Wireless Application Protocol Forum WAPForum, "Wireless Identity Module Part: Security", Specification WAP-260-WIM-20010712-a, 2001.

[13] R. Housley, W. Polk, W. Ford, and D. Solo, "RFC3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", IETF Network Working Group, April, 2002.

[14] W. T. Polk and N. E. Hastings, "Bridge Certification Authorities: Connecting B2B Public Key Infrastructures", NIST, 2000.

[15] L. Lamport, "Password Authentication with Insecure Communication", *Communications of the ACM*, vol. 24, pp. 770-772, 1981.

[16] N. Johansson, M. Kihl, and U. Körner, "TCP/IP Over the Bluetooth Wireless Ad-hoc Network", presented at IFIP-TC6 / European Commission International Conference on Broadband Communications, High Performance Networking, and Performance of Communication Networks, 2000.