Mobile Payment Based on Transaction Certificate Using Cloud Self-Proxy Server

Soonhwa Sung, Eunbae Kong, and Cheong Youn

Recently, mobile phones have been recognized as the most convenient type of mobile payment device. However, they have some security problems; therefore, mobile devices cannot be used for unauthorized transactions using anonymous data by unauthenticated users in a cloud environment. This paper suggests a mobile payment system that uses a certificate mode in which a user receives a paperless receipt of a product purchase in a cloud environment. To address mobile payment system security, we propose the transaction certificate mode (TCM), which supports mutual authentication and key management for transaction parties. TCM provides a software token, the transaction certificate token (TCT), which interacts with a cloud self-proxy server (CSPS). The CSPS shares key management with the TCT and provides simple data authentication without complex encryption. The proposed self-creating protocol supports TCM, which can interactively communicate with the transaction parties without accessing a user's personal information. Therefore, the system can support verification for anonymous data and transaction parties and provides user-based mobile payments with a paperless receipt.

Keywords: Mobile payment, Mutual authentication, Transaction certificate token (TCT), Cloud self-proxy server (CSPS), Paperless receipt.

Manuscript received Aug. 28, 2016; revised Oct. 31, 2016; accepted Nov. 29, 2016.

Soonhwa Sung (corresponding author, shsung@cnu.ac.kr), Eunbae Kong (keb@cnu.ac.kr), and Cheong Youn (cyoun@cnu.ac.kr) are with the Department of Computer Science and Engineering, Software Research Center, Chungnam National University, Daejeon, Rep. of Korea.

I. Introduction

A mobile payment is any transaction that is executed via a mobile device and involves either the direct or indirect exchange of fiscal values between parties. An interesting aspect of mobile payments is that mobile phones can be used as payment devices under all types of payment circumstances. Optimists believe that the new world economy will see the transition of mobile devices from a simple communication device to a payment mechanism [1]–[4].

Recently, some mobile payment systems keep information about the transaction parties on the mobile devices or use it in a transaction without authentication, which makes these systems vulnerable to attack. Most of these payment protocols are designed to preserve the traditional flow of payment data (Client–Merchant–Merchant Bank), which is a transaction that is carried out between a client and merchant. Therefore, it is vulnerable to attacks such as transaction or balance modification by a merchant. This increases the risk that the user's account will be illegally accessed [5]–[12].

Current authentication systems for mobile payments do not support a protocol wherein a customer verifies the transaction parties such as the merchant and customer banks. In addition, the authentications of each transaction party suffer from a heavy certificate burden.

Hence, a new mobile payment system that is unlike previous payment systems is required to reinforce the security of user-based transactions and implement an efficient mobile payment method in the cloud environment.

Therefore, our scheme prevents the merchant from associating a financial institution with the message that originated from the client and supports a user-based payment transaction with a paperless receipt for a product purchase.

The remainder of this paper is organized as follows. Related work is presented in Section II, the Transaction certificate

mode (TCM) is presented in Section III, the mobile payment system using a cloud self-proxy server (CSPS) is detailed in Section IV, and the proposed method is evaluated in Section V. Section VI concludes the paper.

II. Related Work

Mobile phones have been involved in financial transactions such as mobile banking and mobile payment, both of which include the transmission of sensitive information [13].

In [14], mobile payment usage was analyzed in the technology adoption model, in which the adoption of a technology is based on its perceived usefulness and ease-of-use. In [15], users were concerned about their trust in network reliability and having their phone accessed if it was hacked, lost, or stolen. In [16], shopping histories and purchases on mobile phones were studied and it was found that users had few trust concerns while making transactions on their mobile devices over the Internet.

According to [17], [18], the authentication process only verifies names, so users require various other authentications for the authentication process.

Some mobile transaction mechanisms [19]–[21] have proposed eliminating the requirement of a shared secret between the shop and mobile network operator (MNO). Although the shop has no link with the MNO, a message digitally signed by the MNO is considered authentic and its contents are trusted by the shop. When dealing with signed data, one has to distinguish between data authenticity and trust in the message contents, as authentic data may not be true [22].

To address these issues, transaction protocols were based on the NFC cloud wallet model [20], [23]–[25], NFC payment application [21], and the scheme for secure cloud-based NFC transactions [19]. However, these systems have an insufficient architecture and protocol for cloud-based NFC payment applications.

In order to provide any-to-any security where users can conduct transactions securely from anywhere and at any time, mutual verification methods among transaction parties are necessary. In addition, users require a paperless receipt to be received for a product purchase carried out using their portable devices.

Therefore, easy and secure transaction certificates are required to provide a paperless receipt of a product purchase that is accessible by portable devices.

III. TCM

We assume that a user is not known to a merchant in our scheme and that the TCM manages the transaction certificate token (TCT) and interacts with the CSPS securely. Here, we assume that wireless communications are insecure and that there is an attacker. The attacker has the ability to intercept all messages communicated in the proposed scheme. It is assumed that a mobile device such as a mobile phone for mobile payments is permitted for the scheme. The mobile phone can support payment methods such as credit card-, debit card-, or account-based (bank transfer) transactions. Our scheme uses a mobile phone as a mobile device to investigate methods of secure mobile payment.

We assume that a user plays a leading role in the cloud payment system (that is, the user is legitimate) because the user is the main agent of the transaction parties and is the first party verified by the Certificate Authority, which acts as a trusted third party (TTP).

Our scheme needs security for outsourcing and managing large amounts of data in cloud computing. To authenticate mobile payment processing, it would preferentially verify a payment transaction using fewer keys.

A user communicates with the server that directly operates the TCM without complex verifications, so the TCM communicates with a server and operates efficient mobile payments whenever a transaction occurs in the cloud environment. Because the TCM interacts with the CSPS, this makes the scheme secure against corrupt servers and provides more computational efficiency. The CSPS only utilizes keys associated with payment transactions. The general server knows the operation that was performed (for example, SELECT or UPDATE). However, the CSPS should be necessary only when the service provider requires a cipher key, and it does not know the operation is intended for a mobile payment.

1. TCM Overview

A TCM consists of a customer agent (CA), customer bank (CB), merchant agent (MA), and merchant bank (MB) as the transaction parties. The proposed TCT considers limited mobile phone resources; therefore, the TCT interacts with the CSPS whenever a transaction occurs and receives the simple information of transaction acceptance or rejection from the CSPS. The TCT interacts with the CSPS to avoid disclosing information to the customer and merchant server and to verify sensitive information for the mobile payment alone. It is a software token that acts as certificate that verifies the validity of transaction information for a mobile payment.

As illustrated in Fig. 1, the TCM includes a TCT that comprises TCT1 and TCT2, where the CB issues TCT1 and the MB issues TCT2 instead of a franchise terminal. We assume that the TCT issuer is secure.

The TCT issuer is managed by the TTP, which protects

ETRI Journal, Volume 39, Number 1, February 2017
https://doi.org/10.4218/etrij.17.0116.0608

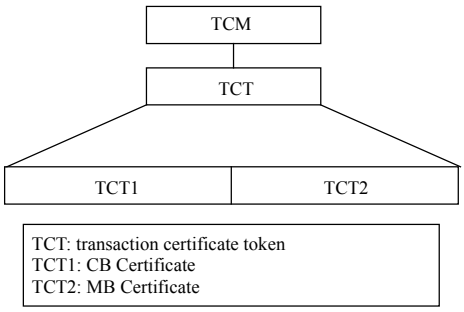

Fig. 1. TCM construction.

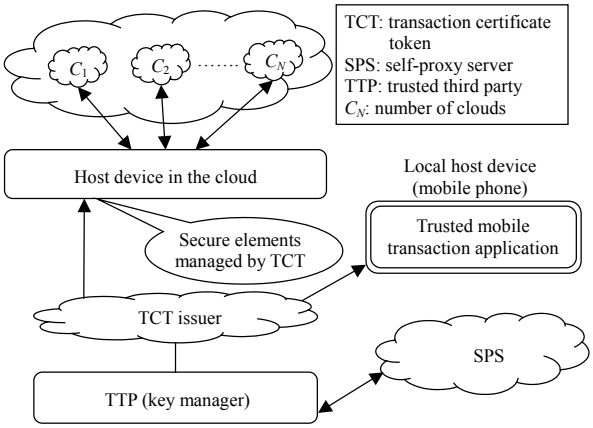

Fig. 2. TCT processing architecture for mobile payments.

sensitive information using secure keys. The TCT issuer sends an information processing system to the TCT for mobile payment processing.

After the remote cloud host communicates with the TCT, the TCT checks whether the remote cloud host is legitimate. To identify anonymous data from the remote cloud host and local host, the TCT issuer communicates with the TTP (the key manager, KM) to receive the keys that interact with the self-proxy server (SPS) in the cloud, as shown in Fig. 2.

The remote cloud host and local host device receive secure elements of the chosen cloud objects from the TCT issuer (see Fig. 2). The secure elements in the remote cloud host are managed by the TCT issuer. The TCT issuer generates the TCT after verifying secure keys from the TTP, which interacts with the CSPS. The local host (mobile phone) associates the TCT issuer with a trusted mobile transaction application when the CSPS and the TTP agree upon the identification of the TCT keys.

2. TCM Contributions

To overcome the inherent design weaknesses of mobile client-server environments and improve upon the secure key management of a mobile payment in a cloud environment, this paper proposes the TCM. In addition, our scheme needs security for outsourcing and managing large amounts of data in cloud computing. To authenticate mobile payment processing, it is better to verify a payment transaction using fewer keys.

Therefore, our scheme needs the TCM, including a TCT, to interact with the CSPS that executes the protocol with self-creating keys and encourages mutual authentication for transaction parties. The TCM is proposed to verify transaction parties and cloud objects in a simple manner. It provides efficient certificates and authentications of all payment transactions. It presents grounds for an argument about cloud financial services in a payment system.

The major contribution of the proposed TCM is that it makes it possible for point-of-sale (POS) transactions to use a certificate mode that enables interaction with the transaction parties directly without requiring affiliation with a local franchise. After the user confirms the product purchase list and cost with his/her phone payment, the payment is completed upon his/her agreement. Because of the TCM, the user can save the product purchase receipt of the POS transaction on the personal mobile phone by downloading it. Therefore, a signature step is not necessary after the mobile payment, including card verification, is complete, thereby rendering a paper purchase receipt obsolete.

As soon as the user's mobile phone touches the POS terminal, the transaction is verified and a purchase receipt is copied from the POS terminal to the mobile phone by some application of data synchronization. The purchase receipt is stored in the mobile phone so a user can manage it directly.

3. TCM Process

TCM supports the transaction parties that are authenticated by mutual trust. To reduce cloud key computing, CSPS deals with the keys for the verifications of a payment transaction. The CSPS provides as few keys as possible by interacting with the TTP for efficient computing.

After the TTP interacts with the CSPS, it sends the TCT issuer the permission, which generates the TCT (see Fig. 3). Confirming whether the host device is in the cloud or is a local host (mobile phone), the TCT issuer generates the TCT and sends it to each host device. Each host device communicates with users and sends them the TCT. Therefore, the TCM provides a mutual authentication solution for mobile payments that have a paperless receipt.

TCM Process:

1) The users of each host device require a certificate from a general server for mobile payment.
2) After receiving the query message, the TCM requires the TTP to verify the transaction parties.
3) After the TTP interacts with the CSPS, it sends the TCT

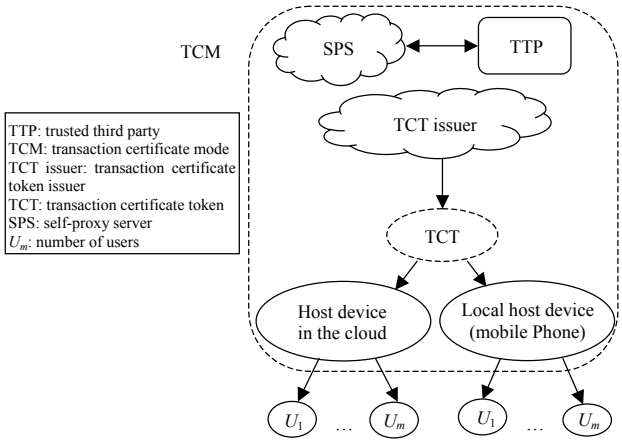

Fig. 3. TCM process diagram.

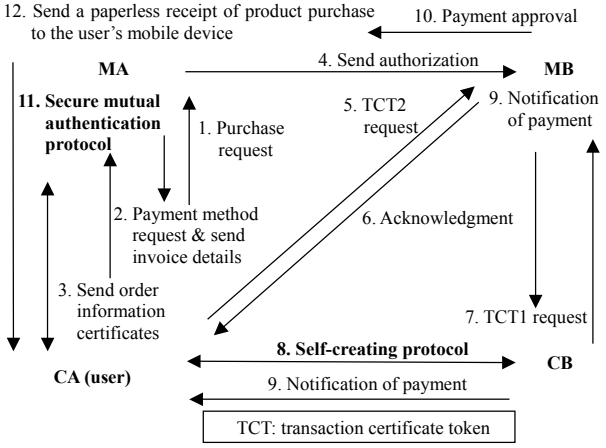

Fig. 4. Mobile payment protocol based on TCT.

issuer the permission, which generates the TCT.

4) The TCT issuer generates a TCT and sends it to each host device for trusted mobile transaction applications.
5) Each host device installs the TCT in the assigned secure software area.
6) After the TCT is calculated in the assigned secure software area, each host device responds to say whether the TCT is legitimate or illegitimate.

IV. Mobile Payment System Using a CSPS

The proposed system operates a mobile payment protocol with the TCT using a CSPS.

1. Mobile Payment Protocol Based on TCT

The TCM mobile payment protocol commences when the CA requests a purchase from the MA and terminates when the CA receives a confirmation of payment from the CB. Unlike current payment protocols, the CA and CB proceed to a secure mutual authentication protocol for a payment after the CA confirms a legal MB using the TCT. Therefore, the protocol supports user-based payments that are different from the current payment mechanism, which is biased in favor of the merchants. For a secure channel, the TCT supports a secure mutual authentication protocol between the CA and CB by the acknowledgement that enables communication between the CA and MB.

As shown in Fig. 4, the flow for payment transactions is as follows:

1) The CA requests a purchase from the MA.
2) The MA prepares an invoice and sends the merchant's encrypted banking information and certificate with the invoice details to the CA.
3) The CA sends the order information certificates to the MA.
4) The MA sends authentications to the MB.
5) The CA requests the TCT2 from the MB.
6) The MB acknowledges the TCT2 and sends it to the CA.
7) The MB requests the TCT1 from the CB.
8) The secure mutual authentication protocol is performed between the CA and CB.
9) The CA receives a notification of payment from the CB.
10) The MB sends payment approval to the MA.
11) The MA communicates with the CA by a secure mutual authentication protocol.
12) The MA sends a paperless receipt of product purchase to the user's mobile device if the MA agrees with the CA.

2. Secure Mutual Authentication Protocol

The secure mutual authentication protocol in Fig. 4 provides for authentication between the MA and the CA. It encourages a secure payment transaction as a general verification and identifies keys associated with the TCT.

The protocol consists of three phases: registration, login, and authentication. Table 1 explains the notation for the secure mutual authentication scheme.

Registration Phase:

User U_c submits ID_c and P_{wc} to S in order to register with server S. Afterwards, S performs the following tasks.

1) It calculates $V_c = H(ID_c, IMEI, Pri_s)$.
2) It calculates $A_c = H(ID_c, IMEI, Pri_s) \oplus P_{wc}$.
3) It stores $(ID_c, V_c, A_c, H(.))$ in the TCT.

Login Phase:

In order to login to S, U_c provides ID_c and P_{wc} for the TCT. The TCT then carries out the following tasks.

1) It calculates $B_c = A_c \oplus P_{wc}$.
2) It calculates $B_c = A_c \oplus P_{wc}$.
3) It calculates $B_c = A_c \oplus P_{wc}$.
4) It checks whether $B_c = V_c$. If the test fails, the request is

Table 1. Notation.

Symbol	Description
U_c	User of client
S	Server
ID_c	Identity of the user
P_{wc}	Password of user
Pr_{ts}	Server's private key
$IMEI$	International mobile equipment identity
N_c	User's generated nonce
N_s	Server's generated nonce
S_k	TCT session key
$\oplus$	Exclusive OR
$\|\|$	Concatenation

rejected.

5) It calculates $C_1 = B_c \oplus N_c$.
6) It sends (ID_c, C_1) to server S.

Authentication Phase:

When S receives a login request (ID_c, C_1), it performs the tasks as detailed below.

1) It tests the format of ID_c. If the format is incorrect, the login request is rejected.
2) It calculates $B_s = H(ID_c, IMEI, Pri_s)$.
3) It calculates $C_2 = C_1 \oplus B_s$.
4) It calculates $C_3 = B_s \oplus N_c$.
5) It calculates $C_4 = H(C_1||C_3||S_k)$ where $S_k = H(B_s||C_2||N_s)$ is the common session key.
6) It sends $\{C_3, C_4\}$ to U_c to achieve unilateral authentication.

Upon receiving $\{C_3, C_4\}$ from S, U_c carries out the tasks as detailed below.

7) It calculates $C_5 = C_3 \oplus B_c$ and $C_6 = H(C_1||C_3||S_k)$ where $S_k = H(B_c||C_5||N_c)$ is the common session key.
8) It checks whether $C_6 = C_4$. If the check is passed, U_c authenticates S and unilateral authentication is complete; otherwise U_c rejects S.
9) It calculates $C_7 = H(B_c||C_5||N_c)$ and sends C_7 to S.

Upon receiving C_7 from U_c, S carries out the following tasks.

10) It calculates $C_8 = H(B_s||C_2||N_s)$.
11) It checks whether $C_8 = C_7$. If the values are equal, S authenticates U_c and mutual authentication is achieved [26].

3. CSPS

For the security of mobile payment data, key management should be carefully implemented. However, this is a challenging issue because of the large quantities of data in the cloud environment. To suggest better ways to protect the confidentiality of data, our scheme proposes the CSPS.

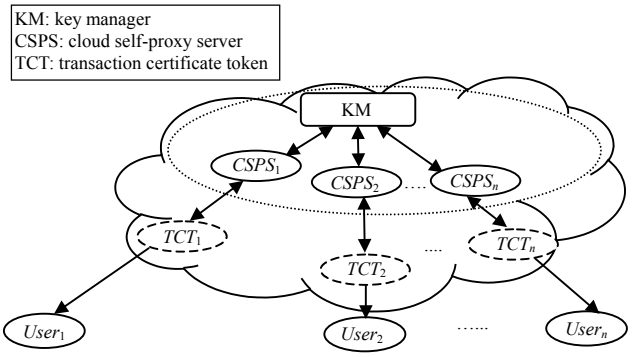

Fig. 5. Distributed CSPS in a cloud environment.

The CSPS provides proxy keys instead of local keys whenever a mobile transaction occurs in a cloud environment. It manages keys for each kind of cloud payment service using the distributed $CSPS_n$. The suggested protocol, called the self-creating protocol, provides the key optimization for the necessary cloud payment service. That is, the CSPS generates a proxy key by itself after verifying a local key is legitimate whenever a transaction occurs. A proxy key is generated only one time and operates many times for the same data in a cloud environment.

The distributed CSPS provides not only encryption and decryption keys but also immediate re-encryption keys for shared data. After communicating with the KM, it automatically receives the necessary keys from the KM using the self-created protocol. A distributed CSPS scheme is one solution where multiple proxy keys are automatically deployed in several clouds. Here, after the CSPS interacts with the TCT, the TCT sends a transaction approval to a user if the key of the transaction data is legitimate (see Fig. 5).

4. Self-Creating Protocol

The aim of the self-creating protocol is to minimize the number of keys for payment transactions with proxy keys using the CSPS.

To reuse the keys of the TCT for payment transactions, the self-creating protocol operates between the CA and CB.

In Fig. 6, a KM creates a proxy key α_n about each distributed server $CSPS_n$ and sends the key α_n to each distributed server $CSPS_n$ without identifying the servers. In addition, after the KM scans all n servers to encrypt the data, the inquiry q that the user created is saved by encrypting it in the cloud server along with each $CSPS_i$. After the encrypted data scans all n servers, it is then decrypted to proxy key α_n.

The flow of the self-creating protocol in Fig. 6 is as follows:

1) KM $\rightarrow$ $CSPS_i$: α_i.

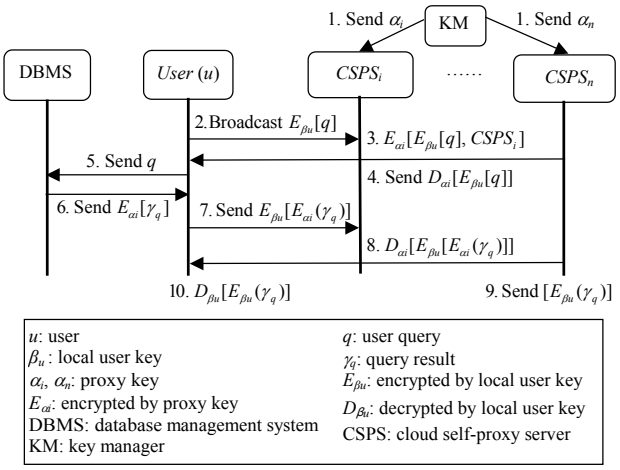

Fig. 6. Self-creating protocol using CSPS.

2) $u_i \rightarrow CSPS_i$: $E_{\beta u}[q]$.
3) for $i = 1, n$, $CSPS_i$: $E_{\alpha i}[E_{\beta u}(q), CSPS_i]$.
4) $CSPS_n \rightarrow u_i$: $E_{\beta u}[q]$.
5) $u_i \rightarrow$ DBMS: q.
6) DBMS $\rightarrow u_i$: $E_{\alpha i}[\gamma_q]$.
7) $u_i \rightarrow CSPS_i$: $E_{\beta u}[E_{\alpha i}(\gamma_q)]$.
8) for $i = 1, n$, $CSPS_i$: $D[E_{\beta u}[E_{\alpha_i - \sum_{j=1}^{i-1} s_j}(\gamma_q), CSPS_i]]$, where $\alpha_i - \sum_{j=1}^{i-1} CSPS_j(\gamma_q)$ represents the partial decrypted key with the first $i-1$ of the CSPS.
9) $CSPS_n \rightarrow u_i$: $E_{\beta u}[\gamma_q]$.
10) u_i: $D_{\beta u}[E_{\beta u}(\gamma_q)]$.

Therefore, even though a user withdraws his joining, the suggested system is flexible concerning key generation because of the self-creating protocol of the CSPS, which actively self-manages the key.

V. Analysis

This section discusses the security and efficiency of the proposed system.

1. Security Analysis

1) Session Key Perfect Forward Secrecy Attack

Session key perfect forward security means that even if the secret key of a user and server are leaked, the generated session key should be safe from the attacker. In the proposed system, even if the password of user P_{wc} and the server's private key Pri_s are compromised, the attacker cannot compute the TCT session key $S_k = H(B_s \| C_2 \| N_s)$ because he/she cannot derive B_s, C_2, and N_s from the authentication phase of the secure mutual authentication protocol.

2) Known-Key Attack

Known-key security means that a compromised past session key cannot be used to derive any further session keys. In the proposed system, the TCT session key $S_k = H(B_s \| C_2 \| N_s)$ is the result of a one-way hash function, which is not recomputed. Therefore, the attacker cannot obtain any further session keys.

3) Denial of Service Resulting from an Attack on the Server

A denial of service attack is one where an adversary updates the wrong verification information of another legitimate user so that the legal user cannot login to the remote server successfully. In this system, there is no risky information stored on the server because the TCT issuer from the TTP updates sensitive information on the TCT whenever a user performs transactions for a payment.

4) Mutual Authentication

The proposed system uses the registration, login, and authentication phase to achieve mutual authentication. The login and authentication phases compute from C_1 to C_8 so that the user authenticates with the server and mutual authentication is achieved.

The authentication protocol can be measured with respect to the following factors over the unreliable networks. Table 2 compares certain cryptographic security attributes of the proposed scheme with those of some relevant schemes.

In the proposed scheme, the authentication phase uses the common session key $S_k = H(B_s \| C_2 \| N_s)$ of step 5 and $S_k = H(B_c \| C_5 \| N_c)$ of step 7, and therefore, the scheme manages a session key.

User U_c submits P_{wc} to S in order to register with server S in the registration phase and changes P_{wc} for the TCT in the login phase because the server and the TCT issuer are managed independently. Moreover, our scheme does not have a clock synchronization problem nor needs an extra hardware device such as a card terminal contact for a mobile payment because the TCT uses CSPS. Similarly, it requires very low bandwidth because the TCT works without complex encryption and key generation, whereas the reference scheme [27] requires one scalar point multiplication operation and two short messages

Table 2. Comparison of security attribute functionality.

Scheme	[27]	[28]	[29]	[30]	Ours
Session key management	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	No	No	Yes	Yes
Password change	Yes	No	No	Yes	Yes
Clock synchronization problem	No	No	No	No	No
Extra hardware device	No	No	No	No	No
Bandwidth requirement	Low	Low	Low	High	Very low

on mobile stations for establishing each session after the initial one-time delegation key is verified.

2. Computational Complexity Analysis

To analyze computational complexity, Table 3 demonstrates the number of cryptographic operations involved for each party. There are three symmetric key encryptions and decryptions in the proposed protocol, that is, the number of XOR operations in the login phase is three, and the number of XOR operations in the authentication phase is three. There are two hash functions for the user in the registration phase and five hash functions for the server in the authentication phase. There are two keyed-hash functions for the server and no keyed-hash functions for the user. There are two key generations each for the user and server. Mutual authentications using the TCM may cause many hash functions for the server. Hence, it has more hash functions than other protocols. In another computation, the proposed protocol has been improved.

Compared to the scheme proposed by Pourghomi and others [34], the proposed scheme uses a hash function seven times and key generation four times for the mutual authentication protocol, whereas the Pourghomi and others scheme uses symmetric key encryption nine times and key generation eight times for the transaction authentication protocol. Therefore, the proposed scheme results in a simpler transaction authentication. In contrast, the Pourghomi and others scheme incurs a high computational cost because of its complex key generations, several encryptions, and signature verifications. The scheme does not effectively manage the keys because it generates four keys (public, private, signing, and verification keys) for a MNO and generates eight keys for the whole scheme. In addition, the scheme utilizes two signature verifications with the signing and verification keys.

3. Performance Analysis

Figure 7 compares the proposed scheme with the previous cloud computing system with respect to processing delay time per unit time of the same size data. For a cloud simulation using Amazon CloudFront, inputs are set by the number of tasks, average processing time of a task, processing deviation, and the load of the task.

The scheme of [35] improved the computation and communication rate by grouping user tasks according to the processing ability of cloud resources, but it did not resolve the minimization of application task flow of the [36] and [37] schemes. The scheme of [38] designed a task flow to minimize the fixed price and time needed to use another cloud resource, but did not consider the resource load to match resources with the task. Therefore, on comparison with the scheme in [38], the proposed scheme has improved processing delay time due to cloud composition of resources for matching tasks.

Table 3. Comparison of security operations.

Cryptographic operations		Number of cryptographic operations				
Scheme		[31]	[32]	[33]	[34]	Ours
Symmetric key encryption/ decryption	U_c	5	3	3	4	3
	S	6	4	4	5	3
Hash function	U_c	2	2	2	N/A	2
	S	1	1	3	N/A	5
Keyed-hash function	U_c	N/A	2	1	N/A	N/A
	S	N/A	1	3	N/A	2
Key generation	U_c	N/A	2	2	2	2
	S	N/A	4	3	6	2

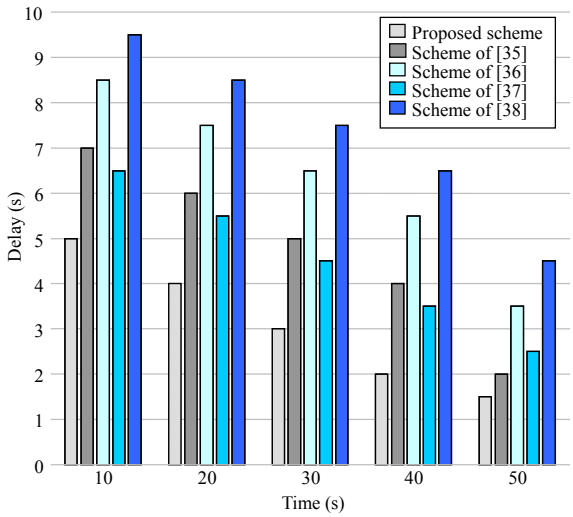

Fig. 7. Comparison of processing delay time.

VI. Conclusion

In comparison with those of previous studies, our contribution provides a user-based mobile payment model with a paperless receipt and provides insight into the security and efficiency of a mobile payment system with paperless receipts using a CSPS that supports a self-creating protocol.

We proposed the self-creating protocol for secure mobile payment in flexible transactions and a mutual authentication protocol between the CA and MA. The first protocol employs keys that have low cryptographic-computation requirements for all payment transactions because of the use of a proxy key. The second protocol, TCT, uses a software token for mutual authentication in mobile payment transactions.

To provide efficient keys for mobile payment transactions,

the CSPS interacts with the TCT and manages proxy keys for it.

Without a paper receipt for a mobile payment, the user-based mobile payment system requires less computation for each payment transaction because the CSPS manages the proxy key as well as a token code to avoid overlapping the key generation and withdrawal.

In addition, the transaction information flow is followed by a user-based transaction that does not require access to the user information by the merchant. This decreases the user's risk because sensitive transaction information cannot be copied and used later to access their account without authentication by the TCT.

Our performance analysis shows that the system requires less computation because the proxy keys of the CSPS are managed and robust against attacks such as a session key perfect forward secrecy attack or denial of service resulting from an attack on the server. In addition, the proposed system provides a paperless receipt for a product purchase without extra hardware devices and operates efficient verifications for the mobile transactions without complex keys because of the TCT and CSPS.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) and the Center for Women in Science, Engineering and Technology (WISET) grant funded by the Korean Government (Program for Returners into R&D by the Ministry of Science, ICT & Future Planning (MSIP)). This work was also supported by the Business for Cooperative R&D between Industry, Academy, and Research Institute funded by the Korea Small and Medium Business Administration in 2015 (Grant No. C0352986).

References

[1] E. Valcourt, J.M. Robert, and F. Beaulieu, "Investigating Mobile Payment: Supporting Technologies, Methods, and Use," *IEEE Int. Conf. Wireless Mobile Comput., Netw, Commun.*, Montreal, Canada, Aug. 22–24, 2005, pp. 29–36.

[2] S. Kungpisdan, B. Srinivasan, and P.D. Le, "A Practical Framework for MobileSET Payment," *IADIS Int. Conf. e-Soc.*, Lisbon, Portugal, June 3–6, 2003, pp. 321–328.

[3] M. Ding and C. Unnithan, *Mobile Payments (mPayments) – an Exploratory Study of Emerging Issues and Future Trends*, Deakin University, 2003. Accessed May 2016. http://www.idea-group.com

[4] K. Pousttchi, "Conditions for Acceptance and Usage of Mobile Payment Procedures," *Proc. M-Business Conf.*, Vienna, Austria, June 2003, pp. 201–210.

[5] J.T. Isaac and J.S. Camara, "An Anonymous Account-Based Mobile Payment Protocol for a Restricted Connectivity Scenario," *Int. Workshop Database Expert Syst. Appl.*, Regensburg, Germany, Sept. 3–7, 2007, pp. 688–692.

[6] S. Kungpisdan, B. Srinivasan, and P.D. Le, "A Secure Account-Based Mobile Payment Protocol," *Proc. Int. Conf. Inform. Technol.: Coding Comput.*, Las Vegas, NV, USA, Apr. 5–7, 2004, pp. 35–39.

[7] L. Tommi and P. Mika, "Mobile Banking Innovators and Early Adopters: How They Differ from Other Online Users?," *J. Financial Service Marketing*, vol. 13, no. 2, Sept. 2008, pp. 86–94.

[8] M.R. Rieback, B. Crispo, and A. Tanenbaum, "Is Your Cat Infected with a Computer Virus?," *Annu. IEEE Int. Conf. Pervasive Comput. Commun.*, Pisa, Italy, Mar. 13–17, 2006.

[9] S. Kamouskos, "Mobile Payment: a Journey through Existing Procedures and Standardization Initiatives," *IEEE Commun. Surveys Tutorials*, vol. 6, no. 4, 2004, pp. 44–66.

[10] J.T. Isaac and J.S. Camara, "Anonymous Payment in a Client Centric Model for Digital Ecosystem," *IEEE Dig. EcoSyst. Technol. Conf.*, Cairns, Australia, Feb. 21–23, 2007, pp. 422–427.

[11] S. Nambiar, C.T. Lu, and L.R. Liang, "Analysis of Payment Transaction Security in Mobile Commerce," *Proc. IEEE Int. Conf. Inform. Reuse Integr.*, Las Vegas, NV, USA, Nov. 8–10, 2004, pp. 475–480.

[12] H. Sun et al., "A Novel Remote User Authentication and Agreement Scheme for Mobile Client-Server Environment," *Int. J. Appl. Math. Inform. Sci.*, vol. 7, no. 4, 2013, pp. 1365–1374.

[13] B. Jenkins, "Developing Mobile Money Ecosystems," International Finance Corporation and Harvard Kennedy School, Washington, DC, USA, 2008.

[14] J. Ondrus and Y. Pigneur, "Towards a Holistic Analysis of Mobile Payments: a Multiple Perspective Approach," *Electron. Commerce Res. Applicat.*, vol. 5, no. 3, 2006, pp. 246–257.

[15] N. Mallat, "Exploring Consumer Adoption of Mobile Payment-A Qualitative Study," *J. Strategic Inform. Syst.*, vol. 16, no. 4, Dec. 2007, pp. 413–432.

[16] S. Hillman et al., "Soft Trust and mCommerce Shopping Behaviors," *Proc. Int. Conf. Human-Comput. Interaction Mobile Devices Service*, San Francisco, CA, USA, Sept. 21−24, 2012, pp. 113–122.

[17] L. Nguyen, "The Missing Link: Human Interactive Security Protocols in Mobile Payment," *Proc. Int. Workshop Security*, Kobe, Japan, Nov. 22–24, 2010.

[18] S.M. Shedid, M. El-Hennawy, and M. Kouta, "Modified SET Protocol for Mobile Payment: An Empirical Analysis," *Int. J. Comput. Sci. Netw. Security*, vol. 10, no. 7, 2010, pp. 289–295.

[19] W. Chen et al., "NFC Mobile Transactions and Authentication Based on GSM Network," *Int. Workshop Near Field Commun., IEEE Comput. Soc.*, Monaco, Apr. 20, 2010, pp. 83–89.

[20] P. Pourghomi and G. Ghinea, "Managing NFC Payments

Applications Through Cloud Computing," *Int. Conf Internet Technol. Secured Trans.*, London, UK, Dec. 10–12, 2012, pp. 772–777.

[21] P. Pourghomi, M.Q. Saeed, and G. Ghinea, "A Proposed NFC Payment Application," *Int. J. Adv. Comput. Sci. Applicat.*, vol. 4, no. 8, Mar. 2013, pp. 173–181.

[22] P. Urien and S. Piramuthu, "Towards a Secure Cloud of Secure Elements Concepts and Experiments with NFC Mobiles," *Int. Conf. Collaboration Technol. Syst.*, San Diego, CA, USA, May 20–24, 2013, pp. 166–173.

[23] G. Tor-Morten, P. Pourghomi, and G. Ghinea, "Towards NFC Payments Using a Lightweight Architecture for the Web of Things," *Comput. J.*, vol. 97, no. 10, 2015, pp. 985–999.

[24] P. Pourghomi and G. Ghinea, "Ecosystem Scenarios for Cloud-Based NFC Payments," *Int. Conf. Manag. Emergent Digit. EcoSyst.*, Neumunster Abbey, Luxembourg, Oct. 28–31, 2013, pp. 113–118.

[25] M.Q. Saeed et al., "Mobile Transactions over NFC and GSM," *Int. Conf. Mobile Ubiquitous Comput., Syst., Services Technol.*, Siem Reap, Cambodia, Jan. 9–11, 2014, pp. 118–125.

[26] S. Sung et al., "User Authentication Using Mobile Phones for Mobile Payment," *Int. Conf. Inform. Netw.*, Siem Reap, Cambodia, Jan. 12–14, 2015, pp. 51–56.

[27] K.K. Sathish, R. Sukumar, and M. Karthiyayini, "An Asymmetric Authentication Protocol for Mobile Hand Held Devices Using ECC Over Point Multiplication Method," *Int. J. Adv. Res. Comput. Science Technol.* vol. 2, no. Special 1, Jan. 2014, pp. 393–399.

[28] K.R.C. Pillai and M.P. Sebastian, "Elliptic Curve Based Authenticated Session Key Establishment Protocol for High Security Applications in Constrained Network Environment," *Int. J. Netw. Security Its Applicat.*, vol. 2, no. 3, July 2010, pp. 144–156.

[29] X. Li, F. Wen, and S. Cui, "A Strong Password-based Remote Mutual Authentication with Key Agreement Scheme on Elliptic Curve Cryptosystem for Portable Devices," *Int. J. Appl. Math. Inform. Sci.*, vol. 6, no. 2, 2012, pp. 217–222.

[30] S.K. Nayak, S. Mohapatra, and B. Majhi, "An Improved Mutual Authentication Framework for Cloud Computing," *Int. J. Comput. Applicat.*, vol. 52, no. 5, Aug. 2012, pp. 36–41.

[31] T.S. Fun et al., "A Lightweight and Private Mobile Payment Protocol by Using Mobile Network Operator," *Int. Conf. Comput. Commun. Eng.*, Kuala Lumpur, Malaysia, May 13–15, 2008, pp. 162–166.

[32] J.T. Isaac and S. Zeadally, "An Anonymous Secure Payment Protocol in a Payment Gateway Centric Model," *Procedia Comput. Sci.*, vol. 10, 2012, pp. 758–765.

[33] S. Manav and T. Shashikala, "Software Tokens Based Two Factor Authentication Scheme," *Int. J. Inform. Electron. Eng.*, vol. 2, no. 3, May 2012, pp. 383–386.

[34] P. Pourghomi, M.Q. Saeed, and G. Ghinea, "A Secure Cloud-Based NFC Mobile payment Protocol," *Int. J. Adv. Comput. Sci. Applicat.*, vol. 5, no. 10, 2014, pp. 24–31.

[35] L. Guo, G. Shao, and S. Zhao, "Multi-Objective Task Assignment in Cloud Computing by Particle Swarm Optimization," *Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Shanghai, China, Sept. 21−23, 2012, pp. 1–4.

[36] S. Pandey et al., "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," *IEEE Int. Conf. Adv. Inform. Netw. Applicat.*, Perth, Australia, Apr. 20–23, 2010, pp. 400–407.

[37] A. Verma and S. Kaushal, "Bi-Criteria Priority based Particle Swarm Optimization Workflow Scheduling Algorithm for Cloud," *Pro. Recent Adv. Eng. Comput. Sci.*, Mar. 6−8, 2014, pp. 1–6.

[38] S. Selvarani and G.S. Sadhasivam, "Improved Cost-based Algorithm for Task Scheduling in Cloud Computing," *Comput. Intell. Comput. Res.*, Tamilnadu, India, Dec. 28–29, 2010, pp. 1–5.

Soonhwa Sung received her PhD in 2005 from the Department of Computer Engineering, Chungnam National University, Daejeon, Rep. of Korea. From 2000 to 2005, she taught in the Department of Computer Web Information, Daeduk College, Daejeon, Rep. of Korea. From 2002 to 2005, she taught in the Department of Computer Engineering, Chungnam National University, From 2006 to 2011, she was a visiting professor at Chungnam National University. She was a visiting professor at Chungbuk National University, Cheongju, Rep. of Korea, in 2012. She is currently a researcher at the Software Research Center (SOREC), Chungnam National University. Her research interests include mobile information security, mobile payment security, user authentication system, cloud security, and sensor network security.

Eunbae Kong has been a professor at Chungnam National University, Daejeon, Rep. of Korea, in the Department of Computer Science and Engineering since 1988. He received his BS and MS degrees in computer science and statistics from Seoul National University, Rep. of Korea, in 1978 and 1981, respectively, and his PhD in computer science from Oregon State University, Corvallis, USA, in 1995. His research interests include machine learning and bioinformatics.

Cheong Youn is a professor at Chungnam National University, Daejeon, Rep. of Korea, in the Department of Computer Science and Engineering. He received his BS degree in Physics from Seoul National University, Rep. of Korea, in 1979, his MS degree in computer science from Illinois State University, Normal, USA, in 1983, and PhD in computer science from Northwestern University, Evanston, IL, USA, in 1988. From 1988 to 1993, he worked at Bell Communications Research, Piscataway, NJ, USA, as a senior researcher. His research interests include software engineering and object-oriented modeling.