Contents lists available at ScienceDirect

The Journal of Systems and Software

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

A mobile payment mechanism with anonymity for cloud computing

Jen-Ho Yang[a], Pei-Yu Lin[b,*]

[a] Department of Multimedia and M-Commerce, Kainan University, Taiwan
[b] Department of Information Communication, and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, 135 Yuan-Tung Rd., Chung-Li 32003, Taiwan

ARTICLE INFO

Article history:
Received 29 September 2014
Revised 30 April 2015
Accepted 9 July 2015
Available online 17 July 2015

Keywords:
Electronic transaction
Mobile payment
Cloud computing

ABSTRACT

In recent years, traditional transactions have been replaced by electronic transactions. To protect the security of the electronic transactions, various electronic payment (e-payment) mechanisms have been proposed. However, we find the previous e-payment mechanisms do not provide the non-repudiation requirement in the client side. Thus, a malicious client can easily deny the transaction and the merchant may not get the payment. In addition, these mechanisms have large computation and communication costs so they cannot be applied to the mobile payment for cloud computing. To solve the above problems, we propose a new mobile payment mechanism with anonymity for cloud computing in this paper. The proposed mechanism not only reduces the computation cost but also provides the non-repudiation requirement in the client side. Compared with the related works, the proposed mechanism is securer, fairer, and more efficient. Therefore, the proposed mobile payment mechanism is more suitable and practical for the cloud computing.

1. Introduction

With the rapid development in network technologies, traditional transactions have been replaced by electronic transactions (e-transaction) in recent years. In e-transaction applications, many people use mobile devices to deal the transactions, which is so-called electronic commerce (e-commerce). Because the e-commerce can be dealt anytime and anywhere, more and more people use it for selling, buying, and marketing. To protect the security of e-commerce, various electronic payment (e-payment) mechanisms have been proposed (Abad-Peiro et al., 1998; Chari et al., 2001; Kungpisdan et al., 2003; Wei et al., 2005; Martinez-Pelaez et al., 2010; Sun et al., 2010; Tsai et al., 2011). Generally speaking, a secure e-payment mechanism has to provide confidentiality, anonymity, integrity, fairness, and non-reputation. To accomplish the above requirements, how to design a secure and fair e-payment mechanism becomes an important issue for mobile commerce.

In 2012, Isaac and Zeadally (2012) proposed an anonymous secure payment mechanism in a payment gateway centric model. In their e-payment mechanism, clients and the merchant communicate each other through a payment gate. The clients do not communicate with the merchant directly so the client anonymity can be accomplished. In addition, they use the symmetric key cryptosystem (Menezes et al., 1996) to provide the confidentiality. Thus, the e-payment information can be well-protected. Besides, their mechanism also provides the integrity by using the message authentication code (MAC). Thus, the clients and the merchant can ensure that the information is not tampered during transferring via networks.

According to the above description, Isaac and Zeadally use the payment gateway to accomplish the confidentiality, anonymity, and integrity. We find that their mechanism satisfies the requirements of cloud computing because the transaction messages can be stored and protected in the payment gateway. That is, the payment gateway can be implemented as a cloud server for the e-payment in mobile commerce (Madlmayr et al., 2008; Pailles et al., 2010; Alpár et al., 2012; Pourghoumi and Ghinea, 2012a,b; Kounelis et al., 2012; Pourghomi et al., 2013, 2014; Moss, 2012; HCE, 2014; GSMA, 2014).

However, we also find that Isaac and Zeadally's mechanism does not provides the fairness and non-reputation requirements for the e-transaction. In their mechanism, the client generates the payment information anonymously to protect the payment privacy. This causes that the client can deny the transaction because the payment information cannot link directly to the client. Moreover, their mechanism uses the redundant symmetric key between the client and the merchant. This redundant key is unnecessary because all messages must be transmitted through the payment gateway. This causes the key management problem of the client. If the client wants to buy items from difference merchants, the client needs to keep many keys for different merchants. This also increases the computation and communication costs in this scenario. According to the above reasons, Isaac and Zeadally's e-payment mechanism has some problems if we want to apply it to mobile payment applications for cloud computing.

[*] Corresponding author. Tel.: +886 3 4638800x2130; fax: +886 3 4638277.
E-mail addresses: jenhoyang@mail.knu.edu.tw (J.-H. Yang), pylin@saturn.yzu.edu.tw, pagelin3@gmail.com (P.-Y. Lin).

http://dx.doi.org/10.1016/j.jss.2015.07.023
0164-1212/© 2015 Elsevier Inc. All rights reserved.

To solve the above problems, we propose a new mobile payment mechanism with anonymity for cloud computing in this paper. The proposed mechanism has the following advantages. First, the proposed model uses a payment gateway between the client and the merchant, and thus the client does not need to communicate with the merchant directly. That is, the client can transact anonymously to protect the payment privacy. Second, the client's bank generates the digital signature as the payment proof. Thus, the merchant can get the payment from the bank even if the malicious client denies the transaction. Third, we eliminate the redundant symmetric keys between the client and the merchant. The client does not need to maintain many keys for different merchants so the key management problem can be solved. Finally, we reduce the computation and communication costs in the user side so the proposed mobile payment mechanism is very suitable for cloud computing environments. According to the above advantages, the proposed mechanism provides the security requirements of confidentiality, anonymity, integrity, fairness, and non-reputation.

The proposed mechanism is securer than the related works because the payment gateway is used to be the cloud server for saving the client's payment information. In addition, the proposed mechanism is fairer than the related works because the payment proof is generated by the client's bank. Thus, the client cannot deny the transaction. Besides, the proposed mechanism has less computation costs so it is more efficient for the mobile payment. According to the above descriptions, the proposed mechanism is securer, fairer, and more efficient than the related works. Therefore, the proposed mobile payment mechanism is more suitable and practical for cloud computing environments.

2. Review of the related work

In this section, we introduce Isaac and Zeadally's e-payment mechanism (Isaac and Zeadally, 2012). Their mechanism has five roles: the client, the merchant, the issuer (the client's bank), the acquirer (the merchant's bank), and the payment gateway. Note that all payment messages among the client, the merchant, the issuer, and the acquirer must be transmitted through the payment gateway. The notations used in their mechanism are shown in Table 1.

Table 1
The notations of Isaac and Zeadally's e-payment mechanism.

Notations	Descriptions
ID_p	The identity of the participant p
NID_C	The temporary identity of the client
TID	The identity of a transaction includes transaction time and date
TST_p	The timestamp generated by the participant p
Stt	The state of a transaction
OD	The order description
$Price$	The amount of the currency
OI	The order information ($OI = \{TID, OD, h(OD, Price)\}$)
TC	The type of card used in purchase process
$TIDReq$	The request of TID
$MIDReq$	The request for the merchant's identity
SEC_{A-B}	The secret shared between the participants A and B
$\{M\}_x$	The symmetric encryption with the message M using the symmetric key x
$h(M)$	The one-way hash function of M
$MAC(M, K)$	Message authentication code of M with the key K
KS_{A-B_i}	The session key shared between A and B, where i is i-bit cyclic shifting of KS_{A-B} for generating the next session key (Isaac and Zeadally, 2012)
$PRequest$	The payment request
$PResponse$	The payment response
$VSRequest$	The value-subtraction request
$VSResponse$	The value-subtraction response
$VCRequest$	The value-claim request
$VCResponse$	The value-claim response

The steps of Isaac and Zeadally's e-payment mechanism are described as follows.

Step 1: The client and the merchant exchange the necessary messages to start the mechanism through the payment gateway by the following sub-steps.

Step 1-1: The client sends NID_C, i, and $TIDReq$ to the payment gateway.

Step 1-2: The payment gateway forwards the above messages to the merchant.

Step 1-3: The merchant sends ID_p and TID encrypted by KS_{C-M_i} to the payment gateway.

Step 1-4: The payment gateway forwards the above message to the client.

Step 2: The client generates the payment requirement $PRequest$ by the following sub-steps.

Step 2-1: The client generates the value-subtraction request $VSRequest = (MAC, [(Price, h(OI), TST_C, TC, ID_M), KS_{C-I_Z}], TC, TST_C)$.

Step 2-2: The client generates $PRequest = \{NID_C,\ ID_I, Price, OI, z,\ VSReques\}_{KS_{C-M_i}}$, and $MAC [OI, Price, NID_C,\ ID_I,\ TST_C, z, h(KS_{C-I_Z}), KS_{C-M_{i+1}}]$.

Step 2-3: The client sends the payment request to the payment gateway.

Step 3: The payment gateway forwards $PRequest$ to the merchant.

Step 4: The merchant generates value-claim request $VCRequest$ by the following sub-steps.

Step 4-1: The merchant decrypts $PRequest$ to obtain OI, TST_C, and $VSRequest$.

Step 4-2: The merchant verifies the validity of TST_C. If it is valid, then the merchant generates $VCRequest = (VSRequest,\ TST_M,\ h(OI),\ TID,\ Price,\ NID_C,\ ID_I),\ \{VCRequest, ID_M, z, h(KS_{C-I_Z})\}_{KS_{M-PG_k}}$, k, and $MAC[(VCRequest, TST_M, z, h(KS_{C-I_Z})), KS_{M-PG_{k+1}}]$.

Step 5: The payment gateway verifies and approve the payment using the private network of the banks by the following sub-steps.

Step 5-1: The payment gateway decrypts $VCRequest$ and verifies the validity of TST_M. If it is valid, then the gateway sends NID_C, ID_M, $VSRequest$, TID, $h(OI)$, z, $Price$, and $h(KS_{C-I_Z})$ to the issuer. Besides, the gateway sends $Price$ and ID_M to the acquirer.

Step 5-2: The issuer checks the validity of the client by the messages from the payment gateway. If the client is valid, then the issuer approves the transaction.

Step 5-3: The acquirer checks $Price$ and ID_M and asks the issuer transfers the money to the merchant's account.

Step 5-4: The issuer generates $VSResponse = \{Stt, h(OI), h(KS_{M-PG_{k+1}})\}_{KS_{C-I_Z}}$ and sends $VSResponse, Stt, h(Stt, h(OI)), h(KS_{C-I_Z})$ to payment gateway.

Step 6: The payment gateway generates $VCResponse = \{Stt, h(Stt, h(OI), h(KS_{C-I_Z}))\}_{KS_{M-PG_{k+1}}}$ and sends it to the merchant.

Step 7: The payment gateway generates the payment response by the following sub-steps.

Step 7-1: The payment gateway generates $PResponse = \{VSResponse\}_{KS_{C-PG_{j+1}}}$ and sends it to the client.

Step 7-2: The client decrypts $PResponse$ to get $h(OI)$. Then, the client checks if the received $h(OI)$ is equal to his own $h(OI)$. If they are not equal, then the client sends the failure message to the payment gateway. Then, the payment gateway starts the recovery procedure or resends the message.