Wireless Pers Commun (2017) 94:713–724
DOI 10.1007/s11277-016-3646-5

An Electronic Transaction Mechanism Using Mobile Devices for Cloud Computing

Jen-Ho Yang[1]

Published online: 26 August 2016

Abstract In recent years, various electronic payment mechanisms have been proposed for cloud computing. However, the related works have high computation and communication costs so they are not really suitable for cloud environments. To solve this problem, we propose a new electronic transaction mechanism using mobile devices for cloud computing in this paper. The proposed mechanism uses the exclusive-or operation and one-way hash function to reduce the computation cost. In addition, it does not require a pre-shared key between the client and the vender for the authentication. Thus, the proposed mechanism has less communication cost. Compared with the related works, the proposed mechanism is more efficient and suitable for the electronic transactions using mobile devices in cloud computing.

Keywords Cloud computing · Mobile devices · Electronic transaction · Authentication · Anonymity

1 Introduction

With the development of network technologies, more and more traditional transactions are implemented on the Internet, which is so-called electronic commerce (e-commerce). In e-commerce, the electronic transaction (e-transaction) mechanism [1–4] is the most important tool because it can be applied to many applications, such as online shopping, online TV, and online music. On the other hand, cloud computing [5–7] is another popular network technology because it can store huge data for users in the remote server. That is, the cloud user does not need to speed lots of money to expand the storage space in their

✉ Jen-Ho Yang
jenhoyang@mail.knu.edu.tw

[1] Department of Multimedia and Mobile Commerce, Kainan University, No. 1, Kainan Rd., Luzhu, Taoyuan County 33857, Taiwan, ROC

own devices. In addition, the user's secret information can be protected in the cloud server. Due to the above reasons, many e-transaction mechanisms for cloud computing have been proposed [8, 9]. In recent years, more and more user utilizes the mobile device to accomplish all e-transactions on the Internet. Thus, the e-transaction mechanism for cloud computing become a popular research topic in e-commerce applications.

In 2013, Yang et al. [8] proposed an electronic payment system for cloud computing. There are five roles in their e-payment system: the client, the shop, the bank, the cloud registration center (CRC), and the trusted authority (TA). They designed the e-payment system in the secure cloud area, which contains the personal cloud and the public cloud. The personal cloud stores the user's registration and payment information, and the public cloud stores the shop's and the bank's transaction information. Before the e-transaction begins, the client, the shop, and the bank have to register to TA to get their certificates. Then, the transaction participants can use the certificates to prove their identities in the cloud environments. Because all transactions are performed in the secure cloud area, the user's payment and purchasing information can be well-protected in their e-payment system.

However, we find that Yang et al.'s e-payment system has some drawbacks. First, their system uses the certificate to authenticate each participant. This causes additional computation costs for each participant in their system. Second, each participant has to share a symmetric key with the other participant for authentication and encryption. If the client wants to purchase from several shops, then the client has to maintain many symmetric keys for different shops. This causes the key management problem for client. Third, their system still has large computation costs so it is not suitable for mobile users in the cloud environments.

To solve the above problems, we propose a new e-transaction mechanism using mobile devices for cloud computing in this paper. The proposed mechanism does not require any certificate for authentication. In addition, the user only needs to keep one secret value to transact with different venders. Besides, the proposed mechanism uses the one-way hash function [10, 11] and exclusive-or operation to design so the computation costs can be greatly reduced. According to the above advantages, the proposed e-transaction mechanism satisfies the security requirements of integrity, authentication, and non-repudiation. Compared with the related works, the proposed mechanism are more efficient and securer. Therefore, it is very suitable for the e-transaction using mobile devices in cloud computing.

2 Review of Yang et al.'s Mechanism

In this section, we review Yang et al.'s e-payment system. Their system has five participants: the client, the shop, the bank, the cloud registration center (CRC), and the trusted authority (TA). Besides, their system is divided into five phases: the registration phase, login phase, certificate issuing phase, withdrawing phase, and payment phase. The notations used in their system are shown in Table 1. Note that all the transaction steps are performed in the cloud. Then, the steps of the five phases in Yang et al.'s e-payment system are shown as follows.

Registration phase

Step 1. The client computes $H_3(PW)$ and sends $(ID_C, H_3(PW))$ to CRC

Step 2. CRC checks ID_C and stores $(ID_C, H_3(PW))$ in its database

Table 1 The notations of Yang et al.'s system

TA	Trust authority
ID_C	The identity of the client
ID_B	The identity of the bank
ID_S	The identity of the shop
Q_C, S_C	The public and private keys of the client
Q_B, S_B	The public and private keys of the bank
Q_S, S_S	The public and private keys of the shop
U_C, V_C	The certificate of the client
U_B, V_B	The certificate of the bank
U_S, V_S	The certificate of the shop
$F_P(\cdot)$	The trapdoor one-way hash function of the personal cloud
$F_B(\cdot)$	The trapdoor one-way hash function of the bank
s	The private key of TA
P_{pub}	The public key of TA
G_1	An additive group with order q
G_2	A multiplicative group with order q
P	A public element of G_1 with order q
e	Bilinear pairing satisfies e: $G_1 \times G_1 \rightarrow G_2$
$H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$, $H_4(\cdot)$	Four one-way hash functions satisfy: $H_1 : \{0, 1\}^* \rightarrow G_1$, $H_2 : \{0, 1\}^l$, $H_3 : \{0, 1\}^* \rightarrow Z_q^*$, and $H_4 : \{0, 1\}^* \rightarrow \{0, 1\}^1$
$E_K(\cdot)$, $D_K(\cdot)$	The symmetric encryption and decryption functions with the key K

Login phase

Step 1. The client chooses a random number r_c in Z_q^* and computes $M_1 = F_P(ID_C, r_c)$. Then, the client sends M_1 to the personal cloud for the user authentication

Step 2. The personal cloud decrypts $F_P(\cdot)$ to get ID_C and r_c and uses ID_C to obtain $H_3(PW)$ from the cloud database. Then the personal cloud chooses a random number r_s in Z_q^* and computes $M_2 = E_{H3(PW)}(r_s)$ and $Mac_1 = H_4(H_3(PW), r_c)$. Finally, it sends Mac_1 and M_2 to the client

Step 3. The client computes $H_4(H_3(PW), r_c)$ and checks if it is equal to Mac_1. If they are equal, then the client decrypts M_2 to get r_s and computes $Mac_2 = H_4(H_3(PW), r_c, r_s)$. Then, the client computes $M_3 = F_P(Mac_2)$ and sends it to the personal cloud. Finally, the personal cloud can decrypt M_3 to get Mac_2, and then the personal cloud checks if $H_4(H_3(PW), r_c, r_s)$ is equal to Mac_2. If they are equal, then the client is a legal cloud user

Certificate issuing phase

Step 1. TA chooses the system parameters G_1, G_2, e, $H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$, and $H_4(\cdot)$. Besides, TA also chooses a random number s in Z_q^* as its private key and computes $P_{pub} = sP$ as its public key. Finally, TA publishes $\{G_1, G_2, q, P, P_{pub}, e, H_1, H_2, H_3, E, D\}$

Step 2. The client sends the requirement message to TA to get his public key, private key the certificate. Then, TA computes $Q_C = H_1(ID_C)$ and $S_C = sQ_C$ as the client's public and private keys, respectively. Besides, TA chooses a random number y in Z_q^* and computes $U_C = E_s(ID_C \oplus y)$ and $V_C = s(H_1(U_C) + Q_C)$. Finally, TA stores (U_C, y) in its database and sends (Q_C, S_C, U_C, V_C) to the client in a secure channel

Step3. After receiving Q_C, S_C, U_C, and V_C, the client checks the validities of U_C and V_C. Note that the shop and the bank also use the above steps to get their certificates (U_S, V_S) and (U_B, V_B)

Withdrawing phase

Step1. The bank service server chooses two random numbers r and k in Z_q^* to compute $R = rP$, $Q_C = H_1(ID_C)$, kQ_C, $K_{BC} = H_2(e(S_B, kQ_C))$, and $M_4 = E_{KBC}(ID_B, U_B, V_B, R)$. Then, the server sends M_4 and kQ_B to the personal cloud

Step 2. The personal cloud computes $K_{CB} = H_2(e(S_C, kQ_B))$ to get (ID_B, U_B, V_B, R) from M_4. Besides, the personal cloud computes $Q_B = H_1(ID_B)$ to verify the bank's certificate (U_B, V_B) by checking if $e(V_B, P) = e(H(U_B) + Q_B, P_{pub})$. If the equation holds, then the personal cloud chooses two random numbers a and b in Z_q^* to compute $R' = aR + bP$, $h = H_3(m, R)$, $h = h'/a \ mod \ q$, and $M_5 = E_{KCB}(ID_C, ID_B, h, U_C, V_C)$. Then, the personal cloud sends M_5 to the bank service server

Step 3. The bank service server uses K_{BC} to get $(ID_C, ID_B, h, U_C, V_C)$ from M_5 and checks if $e(V_C, P) = e(H_1(U_C) + Q_C, P_{pub})$. If the equation holds, then the bank server computes $S = rQ_B + hS_B$ and $M_6 = E_{KBC}(ID_B, S)$ and sends M_6 to the personal cloud

Step 4. The personal cloud uses K_{CB} to get (ID_B, S) from M_6 and computes $S' = aS + bQ_B$. In addition, the personal cloud verify (R', S') by checking if $e(S', P) = e(Q_B, h'Ppub + R')$. If the equation holds, then the personal cloud obtains a valid e-cash (m, R', S')

Payment phase

Step 1. The personal cloud chooses a random number t to compute tQ_C and sends it to the shop service server

Step 2. The shop service server computes $K_{SC} = H_2(e(S_S, tQ_C))$ and $M_7 = E_{KSC}(ID_S, U_S, V_S)$. Then, the shop server sends M_7 to the personal cloud

Step 3. The personal cloud computes $K_{CS} = H_2(e(S_C, tQ_S))$ and $Q_S = H_1(ID_S)$ to verify (U_S, V_S) by checking if $e(V_S, P) = e(H_1(U_S) + Q_S, P_{pub})$. If the equation holds, then the personal cloud computes $M_8 = E_{KCS}((m, R', S'), U_C, V_C)$ and sends it to the shop service server

Step 4. The shop service server uses K_{SC} to get $((m, R', S'), U_C, V_C)$ from M_8 and checks if $e(V_C, P) = e(H_1(U_C) + Q_C, P_{pub})$. If the equation holds, then the shop server computes $Q_B = H_1(ID_B)$ and $h' = H_3(m, R')$. To verify the validity of the e-cash, the shop server checks if $e(S', P) = e(Q_B, h'P_{pub} + R')$. If the above equation holds, then the shop server ensures the payment is valid and successful

According to the above steps, we find that Yang et al.'s system uses the certificate to authenticate each participant. This causes large computation costs for each participant in the system. In addition, each participant in their system has to share a symmetric key with the others for authentication and encryption. This causes the key management problem for users. Besides, their system has large computation costs so it is not suitable for mobile user in cloud environments. To solve the above problems, we propose a new e-transaction mechanism in the next section.

3 The Proposed e-Transaction Mechanism

The proposed e-transaction mechanism has four participants: the trust authority (TA), the client, the vender, and the bank. Note that the client, the vender, and the bank have to register to TA before the transaction starts (Fig. 1). The operation flow of the proposed mechanism is shown in Fig. 2. The notations used in the proposed mechanism are listed in Table 2.

Registration phase

Step 1. The client sends ID_C, PW_C, and the registration request to TA. Then, TA computes $NID_C = h(ID_C \| x)$ and $A = h(NID_C \| PW_C)$ and stores ID_C, PW_C, and NID_C in its database. Finally, TA sends NID_C and A to the personal cloud

Step 2. TA computes $P = h(h\ (NID_C \| PW_C) \| ID_B)$ and sends P and NID_C to the bank service server

Step 3. TA computes $Q = h(h\ (NID_C \| PW_C) \| ID_V)$ and sends Q and NID_C to the vender service server

The steps of the registration phase are illustrated in Fig. 3.

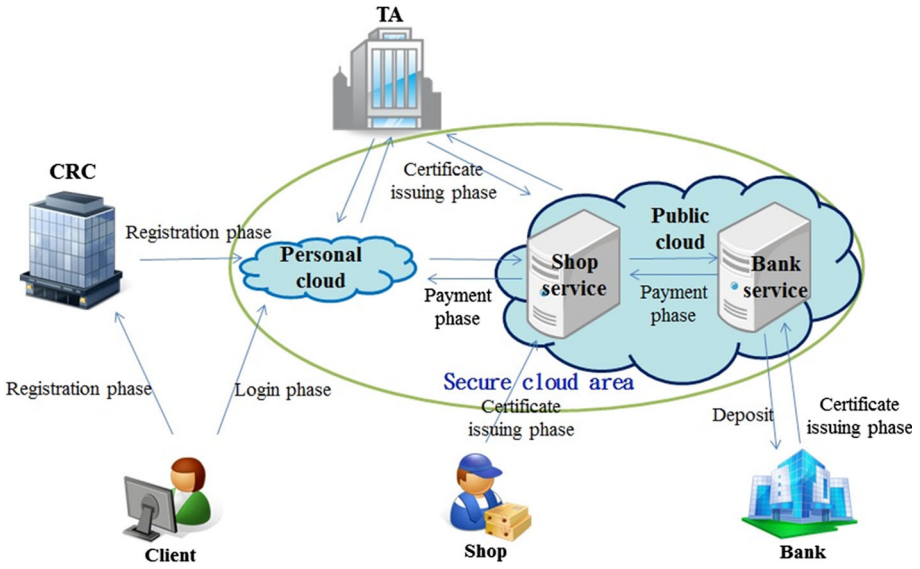

Fig. 1 The operation flow of Yang et al.'s e-payment system

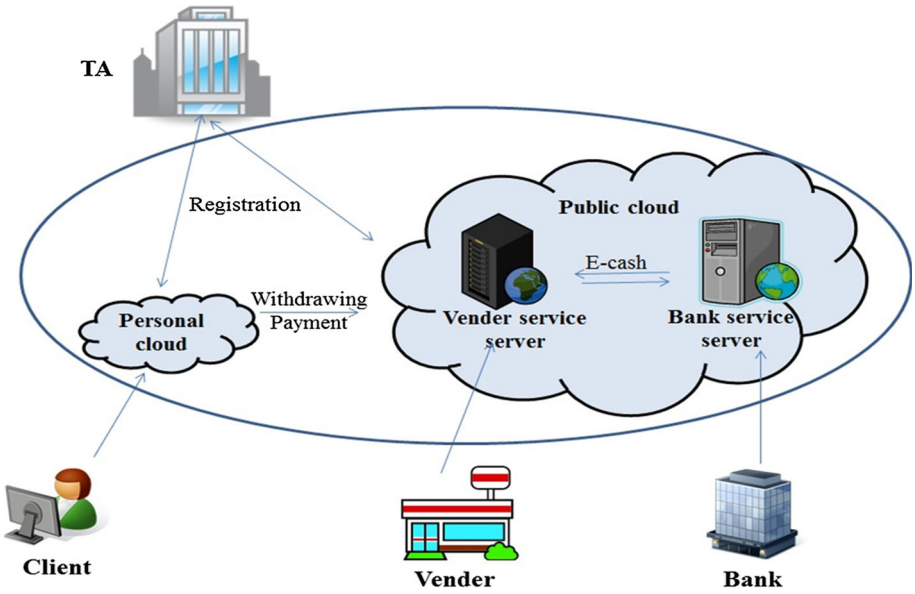

Fig. 2 The operation flow of the proposed e-transaction mechanism

Table 2 The notations of the proposed mechanism

ID_C	The identity of the client
PW_C	The password of the client
NID_C	The anonymous identity of the client
x	The secret key of TA
ID_B	The identity of the bank
ID_V	The identity of the vender
$\oplus$	Exclusive-or operation
$\|\|$	String concatenation operation
$h(\cdot)$	One-way hash function
TS_i	The timestamp generated by the participant i
m	The e-cash information includes the serial number and face value
s	The digital signature
$goods$	The good information includes the price
a	The total price of the goods

Withdrawing phase

Step 1. The personal cloud chooses a random number y to compute $D = h(P \,||\, TS_C) \oplus y$ and $G = h(y \oplus TS_C)$ and sends $\{NID_C, D, G, TS_C\}$ to the bank service server

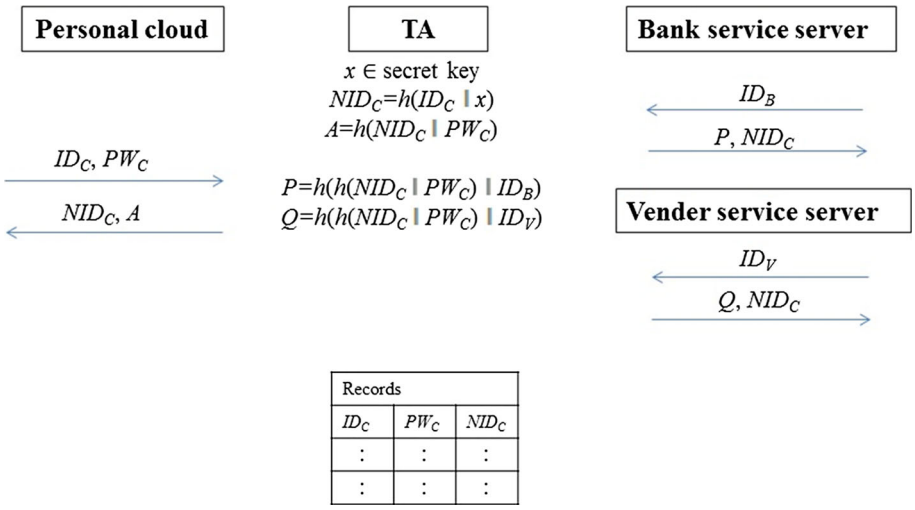

Fig. 3 Registration phase of the proposed mechanism

Step 2. The bank service server checks the validity of the timestamp TS_C. If TS_C is valid, then the server computes $y' = h(P \parallel TS_C) \oplus D$ and $G' = h(y' \oplus TS_C)$. The bank server also checks if $G = G'$ for the authentication. If the equation holds, then the bank server ensures the personal cloud is valid. Thus, the bank server computes $CB = h(y' \parallel G' \parallel TS_B \parallel TS_C)$ and $BC = h(CB \parallel TS_B)$ and sends (BC, TS_B) to the personal cloud

Step 3. The personal cloud checks the validity of the timestamp TS_B. If it is valid, the personal cloud computes $CB' = h(y \parallel G \parallel TS_B \parallel TS_C)$ and $BC' = h(CB' \parallel TS_B)$. The personal cloud also checks if $BC' = BC$ for the authentication. If the equation holds, then the personal cloud ensures the bank is valid. Then, the personal cloud performs any elliptic curve blind signature scheme (such as [11]) to create a blind message M for the e-cash information m. Finally, the personal cloud sends M to the bank

Step 4. The bank service server generates the blind signature S for M and sends S to the personal cloud

Step 5. The personal cloud computes the signature s from the blind signature S to get the signed e-cash

Payment phase

Step 1. The personal cloud computes $H = h(Q \parallel TS_C \parallel a) \oplus goods$ and $F = h(goods \oplus TS_C)$ and sends $\{NID_C, H, F, a, TS_C\}$ to the vender service server

Step 2. The vender service server checks the timestamp TS_C. If it is valid, then the vender server computes $goods' = h(Q \parallel TS_C \parallel a) \oplus H$ and $F' = h(goods' \oplus TS_C)$ and checks if $F = F'$. If the equation holds, then the vender server computes $CV = h(goods' \parallel F' \parallel TS_V \parallel TS_C)$ and $VC = h(CV \parallel TS_V)$ and sends (VC, TS_V) to the personal cloud

Step 3. The personal cloud checks TS_V. If it is valid, then the personal cloud computes $CV' = h(goods \parallel F \parallel TS_V \parallel TS_C)$ and $VC' = h(CV' \parallel TS_V)$. Besides, the personal cloud checks if $VC' = VC$. If the equation holds, then the personal cloud ensures the vender is valid and sends (m, s) to the vender service server

Step 4. Finally, the vender service server verifies (m, s) using the bank's public key. In addition, the vender server sends (m, s) to the bank service server for checking the amount of the payment and the double spending. If the bank server responds a legal payment message to the vender server, then the vender sends the goods to the client

Unlike Yang et al.'s scheme [9], the proposed mechanism does not need the certificate for each participant's authentication. Thus, the computation cost can be reduced. In addition, each participant does not share the symmetric keys with other participant so the key management problem can be solved. Besides, the proposed mechanism uses the one-way hash functions and XOR operations so the computation cost is very low. On the other hand, any digital signature scheme can be easily applied to the proposed mechanism. This increases the flexibility of the proposed mechanism to implement the e-transaction in cloud computing environments. According to the above reasons, the proposed mechanism is efficient and flexible for the e-transaction using mobile devices in cloud computing (Figs. 4, 5).

4 Discussion

In this section, we present the security and performance analyses to show that the proposed mechanism is secure and efficient. The analyses are shown as follows.

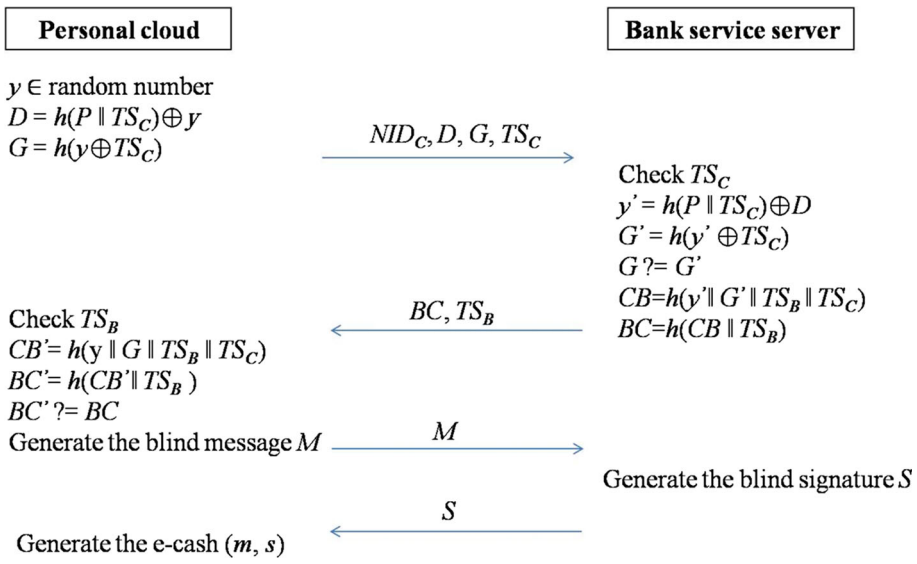

Fig. 4 Withdrawing phase of the proposed mechanism

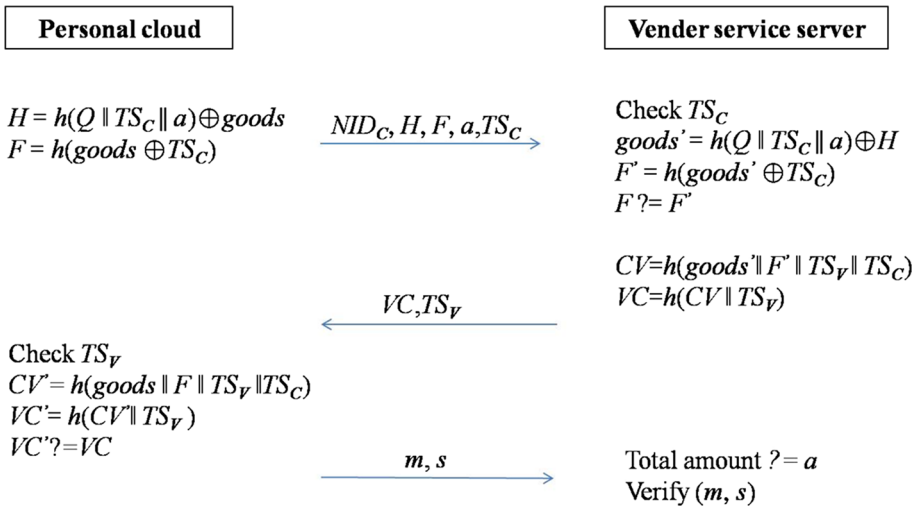

Fig. 5 Payment phase of the proposed mechanism

4.1 Security Analyses

We utilize some typical attacks on the proposed mechanism to analyze its security and show that the proposed mechanism is secure as follows.

4.1.1 Impersonating Attack

Assume that an attacker impersonates the personal cloud to withdraw the e-cash from the bank service server. Then, the attacker randomly generate A'' and y'' to compute $P'' = h(A'' \| ID_B)$, $D'' = h(P'' \| TS_C) \oplus y''$ and $G'' = h(y'' \oplus TS_C)$, and he sends $\{NID_C, D'', G'', TS_C\}$ to the bank service server. However, the attacker cannot be authenticated by the bank server because of the following reason: After receiving the messages from the attacker, the bank server computes $y' = h(P \| TS_C) \oplus D''$ and $G' = h(y' \oplus TS_C)$ and checks if G' is equal to G''. Nevertheless, G' is not equal to G'' because of $P'' \neq P$ and $y'' \neq y'$. Thus, we have $G'' = h(y'' \oplus TS_C) \neq G' = h(y' \oplus TS_C)$. According to the above analysis, the proposed mechanism can prevent the impersonating attack since the attacker does not know the authentication information $A = h(NID_C \| PW_C)$.

4.1.2 Server Spoofing Attack

Assume that an attacker wants to pretend that he is a bank server, and then he generates the forged messages: $CB'' = h(y'' \| G'' \| TS_B \| TS_C)$ and $BC'' = h(CB'' \| TS_B)$ and sends (BC'', TS_B) to the personal cloud. However, this attack cannot succeed because of the following reason: After receiving (BC'', TS_B), the personal cloud computes $CB' = h(y \| G \| TS_B \| TS_C)$ and $BC' = h(CB' \| TS_B)$ and checks if $BC' = BC''$ for the authentication. Nevertheless, the personal cloud will find that $BC' \neq BC''$ because $BC' = h(CB' \| TS_B) \neq BC'' = h(CB'' \| TS_B)$. Therefore, the person cloud will know these messages are sent by the attacker. That is, the server spoofing attack is infeasible for the proposed mechanism.

4.1.3 Insider Attack

Assume that a malicious client tries to obtain TA's secret key x, then he may computes x from his $NID_C = h(ID_C \oplus x)$. However, computing x from $NID_C = h(ID_C \oplus x)$ is impossible because x is protected by a one-way hash function. Similarly, the bank server cannot obtain the user authentication information $A = h(NID_C \parallel PW_C)$ from $P = h(h\ (NID_C \parallel PW_C) \parallel ID_B)$. This is because that A is also protected by the one-way hash function. Therefore, the insider attack is impossible for the proposed mechanism.

4.1.4 Outsider Attack

Assume that an attacker wants to obtain the secret value P, then he intercepts $D = h(P \parallel TS_C)$ from the communications between the personal cloud and the vender service server in the withdrawing phase. Then, the attacker tries to obtain P from $D = h(P \parallel TS_C)$-y. However, it is impossible because the attacker does not know the random number y. In addition, P is also protected by the one-way hash function. According to the above reason, the outsider attack is infeasible for the proposed mechanism.

4.1.5 Replay Attack

Assume that an attacker wants to impersonate a client, then he intercepts $\{NID_C, H, F, a, TS_C\}$ from the communications between the personal cloud and the vender service server. Thus, the attacker generates a fake timestamp TS_C' and re-sends $\{NID_C, H, F, a, TS_C'\}$ to the vender to impersonate a legal client. However, this attack is impossible because $F = h(goods \oplus TS_C)$ contains the real timestamp TS_C. The vender server will find that TS_C' is not the same with TS_C. That is, the proposed mechanism can prevent from the replay attack.

4.1.6 Anonymity

The client uses an anonymous identity NID_C to perform the transaction with the vender in the proposed mechanism. Thus, the vender does not know who the client is. That is, the client's privacy can be protected. If the disputation occurs, then the vender can send NID_C to TA for the judgment. TA can trace the real identity of the malicious client in its database. Therefore, the proposed mechanism can provide the anonymity for the user to protect the buying privacy in the e-transaction.

4.2 Performance Analyses

In the subsection, we present the performance analyses of the proposed mechanism and the related works [9, 12]. Table 3 illustrates the computation costs of the proposed mechanism and the related works as follows.

In Table 3, BP, PM, SY, HA, and XR are bilinear pairing of ECC, point multiplication of ECC, symmetric en/decryption, one-way hash function, and exclusive or computations, respectively. According to [10], the magnitude comparison of the above computation costs can be denoted as BP > PM > SY > HA > XR in practice. Table 3 shows that the computation cost of the proposed scheme is much less than those of the related works [9, 12]. Therefore, the proposed mechanism is more efficient than the related works.

Table 3 The computation costs of the related works

Costs	Schemes		
	[9]	[12]	The proposed mechanism
Bank	1BP + 4PM + 4SY + 2HA	1PM + 2SY	1PM + 4HA + 2XR
Vender	2BP + 3PM + 2SY + 2HA	1PM + 2SY + 1HA	2PM + 4HA + 2XR
Client	8BP + 6PM + 5SY + 7HA	6PM + 4SY + 1HA	3PM + 8HA + 4XR

5 Conclusions

In this paper, we propose an e-transaction mechanism using mobile devices for cloud computing. The proposed mechanism has low computation and communication costs because it uses lightweight operations and less pre-shared keys to design. In addition, the client only needs to keep one secret value to transact with different venders so the key management problem can be solved. Moreover, the proposed mechanism provides the anonymity for clients to protect their buying privacy. According to the above reasons, the proposed mechanism is more efficient and suitable for the e-transactions using mobile devices in cloud computing.

Acknowledgments This work was supported by Ministry of Science and Technology of Taiwan under the Grants Most 105-2410-H-424-005.

References

1. Martinez-Pelae, R., Rico-Novella, F. J., & Satizabal, C. (2010). Study of mobile payment protocols and its performance evaluation on mobile devices. *International Journal of Information Technology and Management, 9*(3), 337–356.
2. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Network and Computer Applications*, *34*(1), 1–11.
3. Dimitrios, Z., & Dimitrios, L. (2012). Addressing cloud computing security issues. *Future Generation Computer Systems*, *28*(3), 583–592.
4. Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., et al. (2014). Security and privacy for storage and computation in cloud computing. *The International Journal of Information Sciences, 258*, 371–386.
5. Xun, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 28*(1), 75–86.
6. Zhang, S., Zhang, S., Chen, X., & Wu, S. (2010). Analysis and research of cloud computing system instance. In *Proceedings of the second international conference on future networks* (pp. 88–92).
7. Zhang, S., Zhang, S., Chen, X., & Huo, X. (2010). Cloud computing research and development trend. In *Proceedings of the second international conference on future networks* (pp. 93–97).
8. Xu, J. S., Huang, R. C., Huang, W. M., & Yang, G. (2009). Secure document service for cloud computing. In *Proceedings of the 1st international conference on cloud computing* (*CloudCom*'09), LNCS 5931 (pp. 541–546).
9. Yang, F. Y., Shu, C. W., Chang, H. N., & Wu, C. (2013). An electronic payment system for cloud computing. *International Journal of Advanced Information Technologies, 7*(1), 60–68.
10. Stallings, W. (1999). *Cryptography and network security: principles and practice* (2nd ed.). Englewood Cliffs: Prentice Hall.
11. He, D., Chen, J., & Zhang, R. (2011). An efficient identity-based blind signature scheme without bilinear pairings. *Computers & Electrical Engineering, 37*(4), 444–450.

12. Yang, J. H., Chang, Y. F., & Chen, Y. H. (2013). An efficient authenticated encryption scheme based on ECC and its application for electronic payment. *Information Technology and Control, 42*(4), 315–324.

Jen-Ho Yang Received the B.S. degree in computer science and information engineering from I-Shou University, Kaoshiung in 2002, and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Chiayi County in 2009. Since 2009, he has been an associate professor with the Department of Multimedia and Mobile Commerce in Kainan University, Taoyuan. His current research interests include electronic commerce, information security, cryptography, authentication for wireless environments, digital right management, and fast modular multiplication algorithm.