Journal of
Computer Science & Systems Biology

Sridhar et al. J Comput Sci Syst Biol 2011, 4:3
DOI: 10.4172/jcsb.1000076

Research Article Open Access

Framework of M-Commerce using ID-Based Cryptography

Sridhar K[1]*, M. Sandeep[2] and D. Sagar[3]

[1]M.Tech Student, Sree Chaitanya College of Engineering, Karimnagar, India
[2]Asst Prof in CSE-Dept, VITS School Of Engineering and Technology, Karimnagar, India
[3]Assoc Prof in CSE-Dept, Sree Chaitanya College Of Engineering, Karimnagar, Indi

Abstract

Now a day's customers are more interested in value-added mobile applications. In order to attract more customers to such mobile applications, a solid, secure and robust trading model is necessity. This paper proposes such a secure trading model named Mobile Electronic Payment (MEP) for the mobile commerce (m-commerce) over wireless mobile networks, which applies the emerging ID-based cryptography for key agreement and authentication. Our MEP attempts to ease the computational cost, reduce the memory space requirement in mobile devices, and meet the requirements for secure trading: avoidance of excessiveness and double expenditure, fairness, user ambiguity and privacy. Our design is transparent to the bearer networks and is of low deployment cost. We expect that our MEP provides a viable trading architecture model for the future mobile applications.

Keywords: Mobile application; Security; Identity-based cryptography; Billing; M-commerce

Introduction

Most mobile applications come with the emergence of electronic trading (mobile commerce or m-commerce); hence good secure mobile trading model must be designed to attract more mobile users for doing business wirelessly. Thus, how to integrate the mobile applications with a secure trading model becomes an important design issue, which will significantly affect the success of any value-added mobile application. This is the major topic of this paper. Mobile applications can be categorized into session-based applications and event-based applications. In event-based applications, user's payment is reflected by one-time events. Examples include sending a message, querying traffic information, or purchasing a song. A session-based application consists of three phases: the session-setup phase, the communication phase and the session release phase.

A customer is charged for a session-based application based on either time spent or data volume transferred, e.g., VoIP-calling, video streaming, audio-streaming, or video-conferencing. There are a few payment models proposed in the literature

Which can be classified into two categories: the traditional payment model and the micropayment model? The examples of traditional payment models include the credit card platforms and the electronic cash platforms the traditional payment models allow only one payment in a payment transaction, which has been widely adopted for the event-based applications. Since a session based application usually requires multiple payments during the execution of this application, with the traditional payment model, it requires multiple payment transactions to complete a session-based application. This is inefficient because heavy signaling and computational overheads are introduced into the network. On the other hand, the micropayment models allow multiple payments in a payment transaction, which is considered more efficient than the traditional payment model. Thus, the micropayment models are often adopted for most of mobile applications. To secure transactions in the public-key cryptography (e.g., RSA) is adopted. Unfortunately, the public-key cryptography requires heavy computation and long execution time, which may not be a good solution in wireless mobile networks. Applied the symmetric-key cryptography such as Advanced Encryption Standard (AES) that is more efficient than the public-key cryptography in terms of computational cost and is more suitable for mobile devices. Unfortunately, the symmetric-key cryptography requires more frequent key establishments and updates to prevent the shared key from being compromised, and hence induces more communication cost due to key establishment and key updates.

Moreover, how to establish the shared key in wireless mobile networks for the symmetric-key cryptography is very challenging. Compared with fixed networks, mobile networks have lower bandwidth, longer transmission latency, and more unreliable connections, and mobile devices are restricted by limited memory size and low CPU computational capability. The installation of mobile applications on a mobile network should be quick and of low cost. To summarize, the following requirements should be addressed when designing a suitable trading mechanism on a mobile network. First, customers expect a robust, secure, and fair trading mechanism which can be applied in different mobile networks. Second, the trading mechanism should be light-weight (i.e., with low computational complexity and low communication overhead) so that it can be easily run on mobile devices. Third, user anonymity should be achieved, that is, users' purchasing behavior or preference should not be traceable by others. Finally, a trading mechanism should be of low implementation cost. In this paper, we design an application-level secure payment model, named Mobile Electronic Payment (MEP), for wireless mobile networks, which attempts to meet these requirements. It is based on a more general trading architecture model which combines both public-key cryptography and symmetric-key cryptography to overcome the disadvantages of both technologies. Specifically, we apply the emerging ID-Based Cryptography in the MEP to generate the public-private key pairs so that the certificate overheads among the network operator (denoted as **O**), the user (denoted as **U**), and the mobile application developer or content provider (denoted as **P**) commonly required in

***Corresponding author:** Sridhar K, M.tech Student, Sree Chaitanya College of Engineering, Karimnagar, India, E-mail: sridhark529reddy@gmail.com

Received April 25, 2011; **Accepted** July 30, 2011; **Published** July 31, 2011

Citation: Sridhar K, Sandeep M, Sagar D (2011) Framework of M-Commerce using ID-Based Cryptography. J Comput Sci Syst Biol 4: 050-054. doi:10.4172/jcsb.1000076

the traditional public-key cryptography can be eliminated. Then, from these public-private key pairs, we generate three symmetric keys *ku–o* (held by **O** and **U**), *ko–p* (held by **O** and **P**), and *ku–p* (held by **U** and **P**) to encrypt and decrypt the signaling messages exchanged among **O**, **U**, and **P**.

An important observation is that these three symmetric keys are established without actually exchanging them among the concerned parties, a unique feature of ID-based cryptography. To prevent the symmetric keys from being compromised, in each payment transaction, the three public-private key pairs (*kpub,o*, *kpri,o*) held by **O**, (*kpub,p*, *kpri,p*) held by **P**, and (*kpub,u*, *kpri,u*) held by **U** are used to generate the new symmetric keys. Our design keeps the *key freshness*2 and thus provides more robust security protection. Moreover, MEP supports both event-based and session-based applications and is suitable for the resource-constrained mobile devices because MEP attempts to alleviate the computational cost and reduce the memory space requirement in mobile devices. We expect that our MEP provides a viable trading model for the future mobile applications. The rest of this paper is organized as follows. In Section II, we briefly illustrate the general conceptual trading model and the basics of the ID-based cryptography. Section III presents the design of MEP. In Section IV, we elaborate on the features and computational overhead of MEP. Finally, Section V concludes our work

Preliminaries

General conceptual trading model

Fig. 1 illustrates the general conceptual trading model for mobile applications which consists of three major components: the network operator **O** the user (customer) **U** and the mobile applications/content provider **P**. The **P**s supply mobile applications to **U**s. The **O** provides network bearer services (e.g., the UMTS bearer services or the WLAN services) to **U**s, through which **U**s may use different kinds of mobile devices to access the applications. **P** and **O** may reside in different networks. For example, **O** is the operator of a cellular network, and **P** resides in the Internet. In this trading model, **O** has to be trusted by **U** and **P**.

Initially, **U** and **P** apply for accounts from **O**, and **O** maintains an account balance (Figure 1 (4)) for each account. The public- private key pairs, (kpub,o, kpri,o), (kpub,p, kpri,p), and (kpub,u, kpri,u), and certificates, co, cp, and cu, which are held by **O**, **U**, and **P**, respectively, are used to address the security issues such as the confidentiality and authentication. The certificate is used to verify the owner of a public key. The certificate uses a digital signature to bind a public key with an individual's identity information (e.g., telephone number or email address).

The public-private key pairs are used to encrypt and decrypt all the signaling messages exchanged among **O**, **U**, and **P**. Before **U** purchases a mobile application from **P**, it initiates a Payment Transaction among **O**,**P** and **U**. The creation process of a payment transaction consists of three phases:

The Withdrawal phase (Figure 1 (5)), the Payment phase (Figure 6), and the Deposit phase (Figure 1 (7)). The process begins at the Withdrawal phase where **U** obtains electronic means (e.g., the electronic tokens or the value- added smart card) from **O**. Then, the process enters the Payment phase. In the Payment phase, **U** issues the electronic means to **P**, which is known as "payment". Then **P** checks the validity of the electronic means. If it is valid, **U** is permitted to purchase a mobile application. The payment may be performed either once or many times, which depends on whether the application is event-based or session-based. For an event-based application, only one payment is made in this phase. For a session-based application, multiple payments may be executed. When the mobile application ends, the process gets into the Deposit phase. In this phase, **P** uses the electronic means obtained from **U** to exchange the payment with **O**, where **O** verifies the electronic means and deposits the payment into **P**'s account.

The mobile electronic payment (Mep) platform

In this section, we present the MEP platform which follows the general trading model. When a new user **U** or a mobile application/ content provider **P** joins the MEP, the Key Distribution procedure (to be elaborated later) is executed to distribute **U** or **P** public-private key pairs denoted as (kpub,u, kpri,u) or (kpub,p, kpri,p), respectively. Then, **U** can purchase a mobile application from **P** by running a payment transaction. In a payment transaction, the signaling messages exchanged among **O**, **U**, and **P** are encrypted using three symmetric keys k_u–o (held by **O** and **U**), k_o–p (held by **O** and **P**), and k_u–p (held by **U** and **P**). The three symmetric keys are updated (by utilizing the public-private key pairs) at the beginning of every payment transaction. A payment transaction consists of three phases, the Withdrawal phase (where **U** obtains te tokens from **O**), the Payment phase (where **U** uses the tokes to purchase a mobile application from **P**), and the Deposit phase (where **P** redeems the obtained tokens from **O**).

In the following subsections, we first illustrate the key distri- bution procedure and then describe how a payment transaction is executed in MEP.

The key distribution procedure: The key distribution procedure generates public-private key pairs for **O**, **U** and **P**. The design of this procedure utilizes the IBC to eliminate the certificate overhead from binding one's ID with its public key. Figure 2 illustrates the message flow for this procedure with the following steps:

Step K1. **O** first generates a public-params set (K, G1, G2, pub,o,

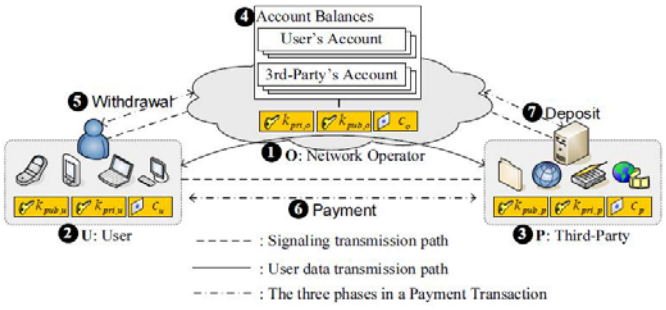

Figure 1: The general trading model for mobile applications.

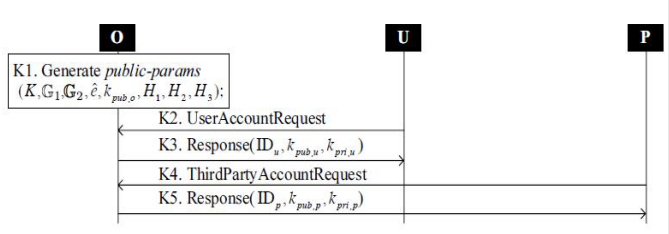

Figure 2: Message flow for the Key Distribution procedure.

H1, H2, H3) by the GENERATE-PARAMS algorithm as shown in Figure 3. The public-params set contains all parameters required in MEP. The usage of the parameters is listed in Table 1. Then **O** publishes the generated public-params set in a public place (e.g., website).

Algorithm 1

O selects a random number $S \in Z_K^*$, and derives its private key kpri,o by computing

$$kpri,o = S \cdot kpub,o \quad (1)$$

where "·" is defined in Property 2 in Section II-B. **O** keeps S and kpri,o confidential.

Step K2. **U** sends **O** the UserAccountRequest message to apply for a user account.

Step K3. Upon receiving **U**'s request, **O** selects an ID[3], IDu, for **U** and creates an account for **U**. Then **O** generates

U's public key kpub, u and private key kpri, u by

$$kpub, u = H1(IDu), \quad (2)$$

and

$$kpri, u = S \cdot kpub, u. \quad (3)$$

O sends kpub,u, kpri,u, and ID_u to **U** through the bearer network link. Since **U** is the customer of **O**, the bearer network4 is considered secure.

GENERATE-PARAMS

1: Generate the pairing parameters (K, G_1, G_2, ˆ);
2: Select an arbitrary generator for G_1 as the public key $k_{pub,o}$;
3: Choose a hash function H_1 : $\{0, 1\}^* \rightarrow G_1$;
4:Choose a hash function H_2: $G_2 \rightarrow \{0, 1\}^N$ for some integerN;
5: Choose a one-way hash function H_3: $\{0, 1\}^* \rightarrow \{0, 1\}^M$for some integer M (e.g., H_3can be SHA-1 and MD5);
6: **return** (K, G_1, G_2, e, kˆ $_{pub,o}$, H_1, H_2, H_3);

Figure 3: The Generate-Params algorithm.

Parameter	Usage	Parameter	Usage
K	The order of G_1 and G_2	$K_{pub,o}$	The O's public key
G_1	The cyclic group with operation "+"	H_1	The hash function used to derive one's ID to its public key
G_2	The cyclic group with operation "×"	H_2	The hash function used to derive the output of the Bilinear Pairing function $\hat{e}$ to a symmetric key
$\hat{e}$	The Bilinear Pairing function	H_3	The hash function used to generate the electronic means

Table 1: The Usage of the Parameters In *public-params* Set.

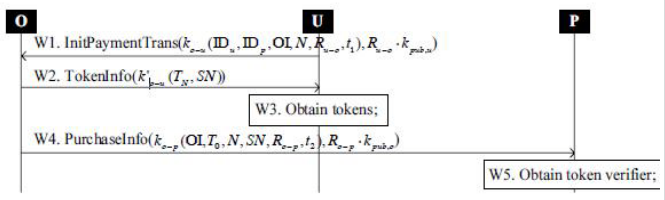

Figure 4: Message flow for the Withdrawal phase of a payment transaction.

Steps K4 and K5. The two steps are similar to Steps K2 and K3, respectively. **P** applies a third-party account by sending the Third Party Account Request message to **O**. **O** selects an ID, ID_p, creates an account for **P**, and generates **P**'s public key kpub, p and private key kpri, p by

$$kpub,p= H1(ID_p) \quad (4)$$

and

$$kpri, p = S \cdot kpub, p \quad (5)$$

O sends kpub,p, kpri,p and ID_p to **P** through the secure link between **O** and **P**.

Payment transaction in MEP

In this section, we describe the execution of a payment transaction in MEP for **U** to purchase a mobile application from **P**. Following the general trading model, a payment transaction in MEP consists of three phases: the Withdrawal phase, the Payment phase and the Deposit phases, which are described below.

Withdrawal phase: In this phase, **U** obtains the electronic means (i.e., the tokens) from **O** Figure 4 illustrates the Message flow for this phase with the following steps. To simplify our description, we use k(D) to denote that the data D is encrypted by the symmetric key k with an efficient symmetric-key algorithm.

Step W1. By browsing **P**'s website,**U** selects a mobile application, gets **P**'s ID, ID_p, and obtains the Order Information (OI) containing the ID and the data unit price of the mobile application. Then, **U** randomly selects an integer Ru−o From Z_K^* and generates the symmetric key ku−o by computing

$$k_u-o= H_2(\hat{} (R_u-o \cdot kpub,o, kpri, u)). \quad (6)$$

Then **U** sends an InitPaymentTrans message to **O** to initiat a payment transaction, where

$$k_u-o(ID_u, ID_p, OI, N, R_u-o, t_1) \text{ and } R_u-o \cdot kpub, u$$

are carried in the message. The first parameter $k_u-o(ID_u, ID_p, OI, N, R_u-o, t_1)$ contains the necessary information for **O** to generate the tokens for **U**. N is the amount of data units **U** will purchase and t1 is the current system time, which is used to prevent message replay and impersonation attacks. The second parameter Ru−o · kpub,u will be used by **O** to derive the symmetric key k_u-o (see Step W2) and authenticate **U**. Note that k_u-o is the same as ku−o so that **O** can decrypt the $ku-o(ID_u, ID_p, OI, N, Ru-o, t1)$ parameter.

Step W2: Upon receipt of the Init Payment Trans message, **O** will perform the following tasks:

(i) **O** extracts the second parameter Ru−o kpub,u from the Init Payment Trans message, and uses this parameter and **O**'s private key kpri,o to derive the symmetric key k_u-o^{as}

$$k_u-o^{=}H_2(\hat{} (R_u-o \cdot kpub,u, kpri,o)). \quad (7)$$

Then **O** uses k_u-o^{to}decrypt $ku-o(ID_u, ID_p, OI, N, Ru-o, t1)$, and **O** obtains the ID*u*, ID*p*, OI, *N*, *Ru−o*, and *t*1.

(ii) To authenticate the sender of the Init-PaymentTrans message, **O** verifies whether Ru−o · H1(ID$_u$)(where Ru−o and ID$_u$ are obtained in (i)) is equal to the second parameter $R_u-o \cdot kpub,u$. If they are not equal (i.e.,$R_u-o \cdot H_1(ID_u)=R_u-o$ kpub,u), the sender is illegal, and the phase quits without sending extra messages. If they are equal (i.e.,$H_1(ID_u)$ =kpub,u), the sender is authenticated and then **O** checks whether the

difference between t1 and the local clock time is within an acceptance windows to prevent from message replay and impersonation.

(iii) If the authentication is successful, **O** will Then generate the tokens for **U**. Suppose that each data unit consumes one token, and N tokens are required for **U**. Let T_N, T_N−1, T_N−2, . . . , T_1denote the N tokens. Initially, **O** selects a random number as the token root T_N. Then **O** executes the GENERATE-TOKENS algorithm (see Figure 5) with arguments T_N and N to generate N tokens. After executing the algorithm, **O** obtains the tokens T_N−1, T_N−2, . . . , $T_1$and a token verifier T0. The token verifier T0 will be used by to make sure that the tokens are sent from **U** in the Payment phase. Each token indicates the data unit price of the mobile application. Then **O** deducts the cost for N tokens from **U**'s account.

(iv) The payment transaction is assigned an unique serial number SN by **O**. Then **O** sends **U** the TokenInfo message carrying k_u−o (TN , SN).

Algorithm 2

Step W3. Upon receipt of the TokenInfo message, **U** uses k_u−o to decrypt the message and obtains T_N and SN. Then, uses the token root T_N to generate N tokens by executing the GENERATE-TOKEN algorithm. Note that due to the lightweight feature of the hash function H the tokens are generated efficiently.

Step W4. **O** selects a random integer Ro−p from Z_K* and generates the symmetric key ko−p as

$$k_o\text{−}p = H_2(\hat{}\ (R_o\text{−}p \cdot kpub, p, kpri, o)) \quad (8)$$

Step W5. Upon receipt of the PurchaseInfo message, **P**extracts the second parameter Ro−p · kpub,o from themessage, and uses this parameter and **P**'s private key kpri,p to derive the symmetric key k_o−p[as]

$$k_o\text{−}p = H_2(\hat{}\ (R_o\text{−}p \cdot kpub,o, kpri,p)). \quad (9)$$

Note that from Proposition III-B1, we know k_o−p=ko−p. **P** uses k_o−p to decrypt the first parameter k_o−p (OI, T_0, N, SN, R_o−p, t_2). Similar to Step W2. (ii), **P** calculates R_o−p· kpub,o (R_o−p is obtained from the first parameter and kpub,o is obtained from the public-params set) and checks whether the result is equal to the second parameter R_o−p· kpub,o carried in the PurchaseInfo message. If they are not equal, the sender is illegal, and the where **P**'s public key kpub,p is obtained by kpub,p= H1(ID_p). Then, **O** sends **P** a PurchaseInfo message to notify that **U** wants to purchase the mobile application. The parameters carried in the message contain ko−p (OI, T0, N, SN, Ro−p, t2) and Ro−· kpub,o, where t2 is the current system time used to prevent from message replay and impersonation and the second parameter R_o−p ·kpub,o will be used by **P**\ to derive the symmetric key k_o−p (to be elaborated in next step). Then, using SN as the index, **O** stores the information (containing ID_u, ID_p, N T_N, and ko−p) required in Deposit phase into its database. **Payment Phase:** In the Paymentphase, **U** uses the tokens to purchase a mobile application from **P**. This phase may consist of one or more payments. We assume that **U** pays Ji tokens in

GENERATE-TOKEN(T_N, N)

1: **for** i ← N − 1 **downto** 0

2 **do** T_i ← $H_3(T_{i+1})$

3: **return** T_{N-1}, T_{N-2}, . . . , T_0

Figure 5: The GENERATE-TOKEN algorithm.

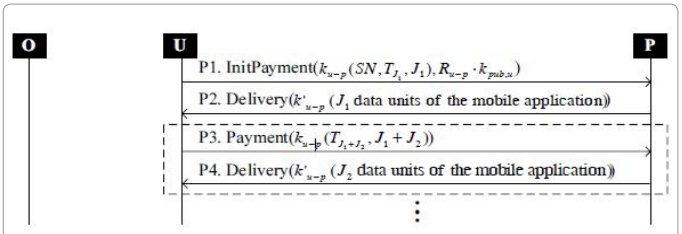

Figure 6: Message flow for the Payment phase of a payment transaction.

the ith payment. Figure 6 illustrates the message flow for the Payment phase with the following steps.

Step P1. **U** randomly selects an integer Ru−p from Z_K* and generates the symmetric key

$$ku\text{−}pby\ k_u\text{−}p = H_2(\hat{}\ (R_u\text{−}p \cdot kpub, p, kpri, u)) \quad (10)$$

where **P**'s public key kpub,p is obtained from kpub,p =H_1(ID_p). Then **U** initiates the first pay- ment by sending a InitialPayment message, where J_1tokens T_1, T_2, . . . T_J1 are carried in this message. The parameters of the InitialPayment message include k_u−p(SN, T_J1, J1) and R_u−· kpub,u. The second parameter Ru−p · kpub,u will be used by **P** to derive the symmetric key k_u−p (see (11), Step P2).

Note that in this step, **P** cannot directly extract kpub,u easily from the second parameter Ru−p · kpub,u, and the InitialPayment message does not contain any information that may leak out **U**'s identity. Therefore, "user anonymity" is well protected.

Step P2. Upon receipt of the InitialPayment message, **P** uses the second parameter R_u−p· kpub,u and **P**'s private key kpri,p to generate the symmetric key k_u−p[by]

$$k_u\text{−}p = H2(\hat{}\ (Ru\text{−}p \cdot kpub,u, kpri,p)). \quad (11)$$

From Proposition III-B1, k_u−p is the same as ku−p. Using the symmetric key k_u−p, **P** decrypts the first parameter ku−p (SN, TJ1, J1) and obtains SN , TJ1 and J1. **P** uses SN as the index to query its database for the token verifier T0, N, and OI. According to the mobile application ID contained in OI P, identifies the mobile application that **U** wants to purchase, and prepares N data units of the mobile application (e.g., streaming data for N seconds). To verify the token TJ_1, **P** checks whether the equation H3(H3 · · · (H3(TJ1)))=?T0……j1 holds. If it holds, **P** ascertains that the token T_J1 is legal. **P** stores T_J1 for verifying the token carried in the next message and discards T0 to release the memory space. Then **P** encrypts the first unit to the J1th unit of the mobile application using the symmetric key k_u−p and responds with the J1\ data units carried in the Delivery message, to **U**.

Step P3. Upon receipt of the Delivery message, **U** decrypts the message using the symmetric key ku−p and obtains the J_1data units. Then, **U** starts the 2nd Payment to purchase the next J_2data units by sending **P** the Payment(T_J1+J2, J1 + J_2) message.

StepP4.Upon receipt of the Payment message, **P** decrypts the message using the symmetric key k_u−p and obtain T_J1+J2[and]J_1+J_2.**P**gets $J_2$by subtracting (*J*1 + *J*2) − *J*1) Then **P** checks whether . the equation *H*3(*H*3 …(*H*3(*TJ*1+*J*2))) ?= *TJ*1……..j2 holds If the equality holds, **P** ascertains that the token T_J1+J2 is legal .stores T_J1+J2 for verifying the token carried in next message and discards the token T_J1. Then, **P** encrypts the next J_2data units of the mobile application using the symmetric key k_u−p and delivers the J2 data units to **U**. Repeating Steps P3 and P4 **U** sends the succeeding tokens to **P**, and **P** delivers the

succeeding data units to **U**. This phase may be terminated if **U** stops paying the token or **P** stops delivering the mobile application.

Deposit Phase: Assume that **P** receives J (J≤N) tokens after the Paymentphase. In the Deposit phase, **P** redeems the J tokens from\ **O**. This phase consists of the following two steps.

Step D1. **P** sends **O** the deposit message carrying the parameters SN and ko−p(TJ, J).

Step D2. Upon receipt of the deposit message,**O** uses the first parameter SN and the index to query its database for ID_u, ID_p,N ,TN ,and ko−p. Using the symmetric key ko−p, **O** decrypts the second parameter ko−p(TJ, J) and obtains TJ and J, and then checks whether the equation H3(H3…(H3(TN)))?=TJ……..N-J holds to verify the token TJ. If the equation holds, **O** deposits the credit for *J* tokens into **P**'s account and takes the cost for J tokens from **U**'s account. The payment transaction is completed. Otherwise (i.e., H3(H3 …(H3(TN))) not equals TJ), **O** treats the sender of the deposit message as an adversary, and the deposit phase will not carried through.

Conclusion

This paper proposed a Framework of M-Commerce using ID-Based Cryptography platform for MEP over wireless mobile networks. In this platform, we take advantage of the emerging the ID-Based Cryptography which eliminatesthe necessity of certificates commonly required by other public key cryptography. Moreover, since ID-based cryptography can establish the shared key between two parties without additional message exchanges, symmetric key cryptography can be still used effectively, leading to significant computational cost. Our study shows that our MEP platform satisfies the requirements of secure trading (such as avoidance of overspending and double spending, fairness, user anonymity, and privacy) and has low computational cost. We expect that our MEP will provide a viable trading model for the future mobile applications and play an important role in the emerging m-commerce industries.