IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON PRIVACY PRESERVATION FOR LARGE-SCALE USER DATA IN SOCIAL NETWORKS

Received January 22, 2018, accepted February 22, 2018, date of publication March 12, 2018, date of current version March 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2813432

Enhanced Instant Message Security and Privacy Protection Scheme for Mobile Social Network Systems

ZHEN WANG[1,2], ZHAOFENG MA[1,2], SHOUSHAN LUO[1,2], AND HONGMIN GAO[1,2]

[1]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Zhaofeng Ma (mzf@bupe.edu.cn)

This work was supported by the National Natural Science Fundamental of China under Grant 61272519, Grant 61170297, Grant 61572080, and Grant 61472258.

ABSTRACT Instant messaging (IM) systems can be considered the most frequently used applications in mobile social networks. Nowadays, people are becoming increasingly concerned about data security and privacy protection with IM applications. Therefore, a comprehensive enhanced secure IM scheme was proposed in this paper, which is based on the elliptic curve cryptosystem and the advanced encryption standard algorithm. An offline key agreement process between users was designed under the computational Diffie–Hellman (CDH) assumption by updating the ephemeral key periodically. The proposed scheme supports denial of replaying attack and denial of forgery attack by utilizing timestamps and the elliptic curve digital signature algorithm. It supports multiple types of messages (such as document and multimedia messages) and prevents privacy leakage by storing sent and received messages with ciphertext. We proved the security of the proposed scheme under the elliptic curve discrete logarithm assumption and the CDH assumption. The comparison results of the proposed scheme with other schemes and the results of an experiment show that it is a comprehensive secure scheme with high security and good practicability.

INDEX TERMS Instant messaging, signature, elliptic curve cryptosystem, privacy protection, key agreement.

I. INTRODUCTION

Instant Messaging (IM) systems have been used more frequently in recent years due to the development of mobile social networks and smartphones. By utilizing IM applications, people can communicate with friends or workmates conveniently and even exchange trade secrets with cooperation partners. Researchers and companies focused initially on functions and convenience when they designed IM systems in the early years [1]. The data that were transferred in IM systems were plaintext for almost all applications. Therefore, attackers on the Internet and illegal users of the applications could eavesdrop or intercept the plaintext easily through software such as Wireshark and even falsify the plaintext to deceive the message recipient. Nowadays, most IM applications have payment functions and users often exchange private and secret data with others, such as private photos, bank account information and trade secrets, through IM applications. Thus, people have become concerned about data security and privacy protection with IM applications. Hence, researchers have turned their attention to the security of IM systems, such as authentication between user and server, encryption of data that are transferred over mobile social networks, verification of when a message reaches the recipient, storage security of data that are received from a sender through IM systems for smartphones, and anonymity among users in IM systems [2]–[7].

II. RELATED WORKS

To solve the aforementioned issues for IM systems on social networks, some approaches and secure protocols have been proposed by researchers in related fields. Kungpisdan S *et al.* proposed a secure IM session key generation and distribution protocol that is based on data that are stored in advance on the client server, which can be applied in the scenario that users are sometimes offline [8]. Luo *et al.* [9] designed a secure IM scheme for protecting the confidentiality, integrity

and authentication of IMs that utilizes certificateless signcryption based on bilinear pairings, which reduces the communication and computational complexities compared with the sign-then-encrypt method. Yusof *et al.* [10] applied a hash algorithm in a secure module of an IM system to encrypt and convert the data into hash values, which can ensure that unauthorized persons cannot view the original data through the network. However, it is not sufficiently secure for IM system to use only a hash algorithm in a secure module . For instant text messaging in mobile devices, Pozo *et al.* [11] proposed a new symmetric encryption mechanism, which has lower complexity, higher performance and stronger robustness compared with the DES and AES encryption algorithms in text messaging. Wang *et al.* [12] designed a secure IM system by using identity-based cryptosystems, which consists of a group of private key generators. Although it incurs a huge computational cost, it avoids the key escrow problem by generating the master key according to a secure distributed key generation protocol.

Wanda P and Hantono proposed an efficient authentication method for mobile IM systems that is based on the Hyper Elliptic Curve Cryptosystem (HECC) algorithm [13], which is a novel approach in IM authentication for elevating the security level of the data. However, it is difficult to implement it because of the difficulty of converting the point (ux, uy) of a Hyper Elliptic Curve into a plaintext message. Loukas A *et al.* designed a secure and privacy-preserving IM system for mobile social networks by using Public Key Infrastructure (PKI) and the AES algorithm [14]. In this scheme, to protect the user's privacy, each time the user logs into the server, he or she needs set his or her pseudonym. One can obtain a user's location only after acquiring the user's permission. To improve the security level of IM applications, Wanda P *et al.* proposed a P2P secure mobile IM model that is based on a virtual network [15]. Messages must be delivered over the virtual network in this secure model. Tung *et al.* [16] designed an enhanced self-destructing-message IM architecture for mobile devices, in which the messages are encrypted by ephemeral keys and when the messages' constraints are satisfied, the ephemeral key that is used for encryption is deleted. Thus, the encrypted messages become unrecoverable. In addition, this scheme can be applied in the scenario in which the message recipient is sometimes offline. Eldefrawy *et al.* [17] proposed a secure IM protocol for a centralized communication group that is based on the ElGamal cryptosystem, RSA algorithm, and Chinese Remainder Theorem (CRT). It is suitable for enterprises, but cannot be applied to general IM applications such as WhatsApp and WeChat.

A. Ruiz-Martínez and Inmaculada Marín-López [18] proposed a lightweight mobile signature service that is based on PKI and the Session Initiation Protocol (SIP), which can generate an electronic signature with a mobile device for a user to sign an electronic document. IT can guarantee the integrity, authenticity and nonrepudiation of the document that is exchanged in the system. Karabey I *et al.* designed and applied a cryptographic IM application that is based on the PKI and AES algorithms [19]. However, it's an online chat-room application and is not suitable for mainstream IM systems. Chen H C *et al.* proposed a secure mobile IM scheme that utilizes Pseudo One Time Pad encryption and the Diffie-Hellman key exchange protocol [20]. This scheme takes the small memory and low processor speed of mobile devices into consideration. However, it's only suitable for encrypting text messages and cannot be applied to multimedia messages. Qin and Xu [21] designed a secure IM approach by adding a secure hardware module into the device. It has a high security level and is realizable by the government and military, but difficult for general persons. For sharing copyrighted or private multimedia content among users, Feng *et al.* [22] proposed an authorization delegation scheme that is based on proxy re-encryption and bilinear maps, which achieved fine-grained authorization delegation.

In addition to those encryption approaches, researchers have focused their attentions on detecting and filtering sensitive or spam messages in IM systems, especially in enterprise IM systems [23]–[26]. In summary, the aforementioned approaches are partially secure IM schemes for mobile social networks. That is, there is no comprehensive secure approach that considers encryption, signature, message authentication and the low processor speeds of mobile devices. Thus, in this paper, we designed a secure IM scheme that not only considers these features, but also meet the requirements of protecting the received data on devices and denial of replay attacks on the system. Details and an analysis of the scheme will be presented in the following section.

III. PRELIMINARIES

Some definitions and difficult mathematical problems for elliptic curve described as follows:

A. DEFINITION OF ELLIPTIC CURVE

An elliptic curve E defined over F_p is a set of points $P = (x_n, y_n)$ where x_n and y_n are elements of F_p that satisfy a certain equation, if p is an odd prime and $p > 3$, then a and b shall satisfy $4a^3 + 27b^2 \neq 0 \pmod{p}$, and every point $P = (x_n, y_n)$ on E (other than the point 2) shall satisfy the equation in F_p: $y^2 = x^3 + ax + b$. All the points that satisfy the equation and the point 0∞ at infinity together constitute an elliptic curve.

All the points that satisfy the equation and the point 0∞ at infinity together constitute an elliptic curve, which is denoted as $E(F_p)$. For a secure cryptosystem, $E(F_p)$ should satisfy the following conditions [27]:

a) A field size p that defines the underlying finite field F_p, where $p > 3$ should be a prime.

b) If the elliptic curve was randomly generated, a bit string SEED with length at least 160 bits is needed (this is the Optional parameters).

c) Two field parameters a and b in F_p which is used to define the equation of the elliptic curve E: $y^2 = x^3 + ax + b$.

d) A point $G = (x_G, y_G)$ of prime order on E, *where* $G \neq 0$ is a must condition.

e) The order n of the point G, should be satisfied $n > 2^{160}$ and $n > 4p^{1/2}$;

f) The cofactor $h = \#E(F_p)/n$ *is* an optional parameter.

B. DEFINITION OF THE ADD OPERATION ON AN ELLIPTIC CURVE

Supposing that P and Q are two arbitrary points on $E(F_p)$ and $P \neq Q$, draw a straight line L through the two points and L will intersect with $E(F_p)$ at point M. Draw a straight line L' that is parallel to the Y-axis and L' will intersect with $E(F_p)$ at point N. Then, the add operation on the elliptic curve can be defined as $P + Q = N$, where point M is called the negative element of point N and is denoted as $-N$.

C. DEFINITION OF THE MULTIPLE POINT OPERATION ON AN ELLIPTIC CURVE

Refer to the definition of the add operation on an elliptic curve and if $P = Q$, draw a tangent line L through point P and find point N in the same way as with the add operation. Then, the multiple point operation on the elliptic curve can be defined as $P + P = 2P = N$.

D. ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (ECDL)

Given two points on $E(F_p)$ and $Q = n \cdot P$, where n is unknown and $n \in Z_q^*$, it is difficult to calculate n with polynomial-time algorithms.

E. COMPUTATIONAL DIFFIE-HELLMAN PROBLEM (CDH)

Given three points $P, m \cdot P$ and $n \cdot P$, where m and n are unknown and $m, n \in Z_q^*$, it is difficult to calculate $m \cdot n \cdot P$ with polynomial-time algorithms.

IV. PROPOSED ENHANCED SECURE IM SCHEME FOR MOBILE SOCIAL NETWORKS

A. REQUIREMENTS OF THE ENHANCED SECURE IM SCHEME

From the aforementioned research, we can confirm that the enhanced secure IM scheme should meet the following requirements:

1) Confidentiality. Messages that are transferred on the Internet must be encrypted so that only the specified recipient can decrypt the messages and obtain the correct plaintext.

2) Integrity. A client on the device must be able to determine whether the received message has been falsified by an attacker.

3) Authentication. A client on the device must be able to verify whether the received message was sent by the specified sender.

4) Denial of Replay Attack. A client on the device must be able to perform duplicate message elimination. Otherwise, attackers can replay the captured legal messages to disturb the normal functioning of the IM system.

5) The scheme supports various message types, including text, document, picture, audio and video.

6) The key for encrypting messages should change between sessions in case the key becomes corrupted and is applied to decrypt massive numbers of messages.

7) The scheme supports sending encrypted messages when the specified recipient is offline.

8) The messages that are stored in the device must be encrypted in case the device is lost to prevent leakage of the user's private data.

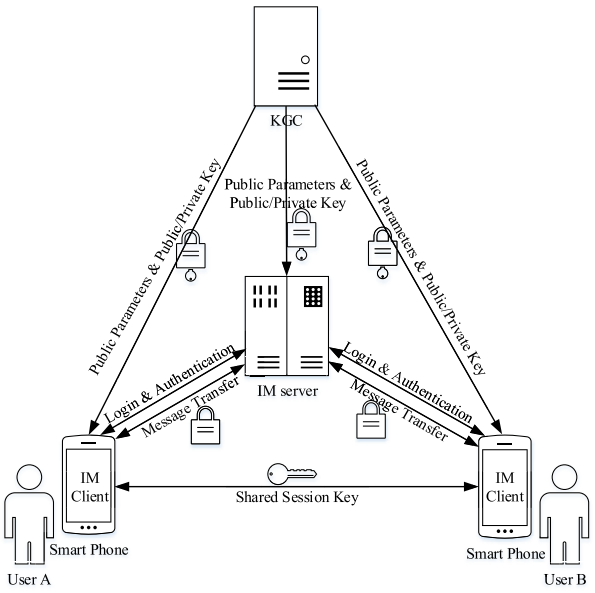


FIGURE 1. Instance of an application scenario for the proposed scheme.

B. ARCHITECTURE OF THE PROPOSED ENCHANCED SECURE IM SCHEME

To achieve mutual authentication between a user and the IM server, as well as between users, an asymmetric cryptographic algorithm is essential for the proposed scheme. Since PKI cryptography suffers from issues of certificate management and withdrawal, we use an identity-based cryptosystem in this paper. The identity-based cryptosystem utilizes a user's unique identifier as the user's public key, such as an ID card number or E-mail address. When a user signs up for an IM account, the account must be bound to the user's email address or phone number. The IM server authenticates the user by sending a verification code to the mailbox or phone number bound to the IM account and prevents the user from maliciously registering. An instance of an application scenario for the scheme that is proposed in this paper shown in Fig. 1, which has three modules: Key Generation Center (KGC), IM server and IM client.

The functions of these modules are described in detail as follows:

1) KGC

The KGC selects appropriate system parameters to ensure the security of the public-key cryptographic algorithm. Then,

the KGC makes some system parameters known to the public. In addition, the KGC generates a public/private key for each user based on the user's unique identifier and sends the private key to the user in a secure way.

2) IM SERVER

Anyone who uses the IM system must register an account for himself or herself in the system. The IM server stores all of the users' accounts, hash values of login passwords, individual information, and so on. The IM server acts as a transfer station during message transmission, which means that all of the messages that are transferred in the system pass through the IM server. If the recipient of the message is online, the server will transmit the message immediately; otherwise, will store the message in a database until the recipient is online, when it will retransmit the message to the recipient and delete it from database.

3) IM CLIENT

Users send and receive messages with an IM client in encryption mode. With an IM client, users can obtain session keys that are shared with others, which are used as encryption keys of the symmetric encryption algorithm to encrypt messages that are transferred in the system. With an electronic signature algorithm and an identity-based cryptosystem, the IM client can detect if the message was sent by the specified sender and falsified or replayed by attackers. In addition, the IM client can protect the sent and received messages in the database and store them with encryption, in case the device is lost.

C. SETUP OF AN IDENTITY-BASED PUBLIC KEY CRYPTOSYSTEM

The implementation of an identity-based cryptosystem has been generally based on Elliptic curve Cryptography (ECC) since it was proposed.

1) SETUP OF KGC

The setup of KGC for an identity-based cryptosystem is as follows:

a) KGC select a suitable elliptic curve $E(F_p)$ that satisfies the security requirements.

b) Select a point P on $E(F_p)$ as a generator, where the order of point P is a prime number, which is denoted as q. Then, KGC can obtain an additive cyclic group G that is generated by point P, such that the order of group G is also q.

c) Select a number $X_s \in Z_q^*$ randomly as the master private key of KGC and store it in secret. Then, calculate the public key of KGC as $P_{sys} = X_s \cdot P$.

d) Select three cryptography hash functions that satisfy the following:

$$\begin{aligned} &H_0 : \{0,1\}^* \to Z_q^* \\ &H_1 : \{0,1\}^* \times G \to Z_q^* \\ &H_2 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \times G \to \{0,1\}^k \end{aligned} \quad (1)$$

e) Finally, KGC makes parameters $\{E(F_p), G, q, P, P_{sys}, H_1, H_2\}$ known to the public.

2) EXTRACT PRIVATE KEY FOR USER

The unique identity of an arbitrary user A is denoted as ID_A. KGC chooses a number $r_A \in Z_q^*$ randomly and calculates $R_A = r_A \cdot P$ and $h_A = H_1(ID_A, R_A)$. The public key of A is $P_A = R_A + h_A \cdot P_{sys}$. Then, it makes R_A and ID_A known to the public. After that, KGC calculates the private key $Q_A = r_A + h_A \cdot X_s$ of user A and sends Q_A to A in a secure channel. User A can verify Q_A by $Q_A \cdot P = R_A + H_1(ID_A, R_A) \cdot P_{sys}$.

D. LOGIN AND KEY AGREEMENT BETWEEN USER AND SERVER

For the IM scheme that is proposed in this paper, a session between a user and a server is defined as the period of time between when the user logs in and goes offline (due to logout or network loss). In this period of time, the user may search for friends or change the user's individual information through the server, and the server may push some useful system information to the user. For a secure IM scheme, these messages should be encrypted during transmission and the server must verify the user's identity.

1) LOGIN AND KEY AGREEMENT PROCESS

In this scheme, user login and key agreement between user and server are combined into one process to save time. In addition, we use the ECDSA algorithm to sign and verify these messages. Details of the process are described as follows:

a) User A calculates the public key of the IM server as $P_{server} = R_{server} + H_1(ID_{server}, R_{server}) \cdot P_{sys}$. R_{server} was packaged in the software of the IM client when it was released.

b) User A selects a number $e_A \in Z_q^*$ randomly such that $0 < e_A < q$ and calculates $T_A = e_A \cdot P$.

c) Take T_A as the key seed and obtain the key of the AES algorithm by utilizing function $AES_{key} = f(T_A)$.

d) Encrypt the login password and obtain the encrypted data $d = \text{Encrypt}_{\text{AES}}(password, AES_{key})$.

e) Calculate $R = e_A \cdot P_{server}$ and obtain the current timestamp T_{time}. Then, calculate the number $m = H_0(T_{time}||ID_A||R_A||d||R)$.

f) Select a number $k \in Z_q^*$ randomly such that $0 < k < q$ and calculate $R_1(x_R, y_R) = k \cdot P$. Then, set $r = x_R$.

g) Calculate the number s by the user's private key Q_A with function $s = (m - Q_A \cdot r)k^{-1} \bmod q$.

h) Check r and s. If $r = 0$ or $s = 0$, repeat the process from step f.

i) Organize the message $<T_{time}, ID_A, R_A, d, R, r, s>$ into JSON format. Then, send the message to the server.

When the server has received the message, the server will process it as follows:

a) The server calculates the number $m = H_0(T_{time}||ID_A||R_A||d||R)$ and the public key P_A of user A such that $P_A = R_A + H_1(ID_A, R_A) \cdot P_{sys}$.

b) Calculate the point $U(U_x, U_y) = s^{-1}(m \cdot P - r \cdot P_A)$. If $U_x = r$, store P_A and proceed to the next step; otherwise, refuse the user's request to log into the server ($Ux \neq r$ means that the message has been falsified).

c) The server calculates key seed T_A of the AES algorithm such that $T_A = Q_{Server}^{-1} \cdot R$, where Q_{Server} is the private key of the server.

d) The server maintains a list $L<User, Timestamp>$ in the database. Search for the tuple $<ID_A, *>$ in L. If a tuple $<ID_A, T'_{time}>$ is found and $T_{time} > T'_{time}$, update the tuple $<ID_A, T'_{time}>$ by $<ID_A, T_{time}>$ in L and proceed to the next step (the user has logged in successfully); otherwise, refuse the request of the user to log into the server (it is a replay attack).

e) Obtain the key of the AES algorithm by utilizing function $AES_{key} = f(T_A)$ and obtain the password $pwd = \text{Decrypt}_{\text{AES}}(d, AES_{key})$. Obtain the user's password pwd' from the database of the IM server and verify $\text{SHA1}(pwd) = pwd'$ (the password of user was stored as a hash value in the database for security). If it satisfies the equation, proceed to the next step; otherwise, refuse the request of the user to log into the server.

f) The server chooses a number $e_S \in Z_q^*$ and $0 < e_S < q$ randomly and calculates the point $T_S = e_S \cdot P$.

g) Select a number $k \in Z_q^*$ and $0 < k < q$ randomly and calculate $R(x_R, y_R) = k \cdot P$. Then mark $r = x_R$.

h) obtain the current timestamp T_{time} and calculate the number $m = H_0(T_{time}||ID_{Server}||T_S)$. Then calculate the number s by server's private key Q_{Server} with function $s = (m - Q_{Server} \cdot r) \cdot k^{-1} \bmod q$.

i) Check r and s, if $r = 0$ or $s = 0$, then repeat the process from step f.

j) Organize the message $<T_{time}, ID_{Server}, T_S, r, s>$ in JSON format and send the message to the user.

When the user has received the message, the user will process it as follows:

a) The user calculates the number $m = H_0(T_{time}||ID_{Server}||T_S)$ and the point $U(U_x, U_y) = s^{-1}(m \cdot P - r \cdot P_{Server})$. If $U_x = r$, proceed to the next step; otherwise, refuse the key agreement with the server ($Ux \neq r$ means that the message has been falsified).

b) The user maintains a list $L<User, Timestamp>$ in the client. Search for the tuple $<ID_{Server}, *>$ in L. If a tuple $<ID_{Server}, T'_{time}>$ is found and $T_{time} > T'_{time}$, update the tuple $<ID_{Server}, T'_{time}>$ by $<ID_{Server}, T_{time}>$ in L and proceed to the next step (the user has accepted it); otherwise, refuse the key agreement with the server (it is a replay attack).

Finally, user A will calculate two shared secrets K_{AS}^1 and K_{AS}^2 and the final session key SK_{AS}.

$$\begin{aligned} K_{AS}^1 &= Q_A \cdot T_S + (Q_A + e_A) \cdot P_{Server} \\ K_{AS}^2 &= e_A \cdot T_S \\ SK_{AS} &= H_2(ID_A, ID_{Server}, T_A, T_S, K_{AS}^1, K_{AS}^2) \end{aligned} \tag{2}$$

The IM server will calculate two shared secrets K_{SA}^1 and K_{SA}^2 and the final session key SK_{SA}.

$$\begin{aligned} K_{SA}^1 &= Q_{Server} \cdot T_A + (Q_{Server} + e_S) \cdot P_A \\ K_{SA}^2 &= e_S \cdot T_A \\ SK_{SA} &= H_2(ID_{Server}, ID_A, T_S, T_A, K_{SA}^1, K_{SA}^2) \end{aligned} \tag{3}$$

As the group G is an additive cyclic group, we can obtain the following equations:

$$\begin{aligned} K_{AS}^1 &= Q_A \cdot T_S + (Q_A + e_A) \cdot P_{Server} \\ &= Q_A \cdot T_S + (Q_A + e_A) \cdot (R_{Server} \\ &\quad + H_1(ID_{Server}, R_{Server}) \cdot P_{sys}) \\ &= Q_A \cdot T_S + (Q_A + e_A) \cdot (r_{Server} + h_{Server} \cdot X_s) \cdot P \\ &= Q_A \cdot T_S + Q_A \cdot Q_{Server} \cdot P + e_A \cdot Q_{Server} \cdot P \\ &= Q_A \cdot T_S + Q_A \cdot Q_{Server} \cdot P + Q_{Server} \cdot T_A \\ &= Q_A \cdot T_S + Q_A \cdot Q_{Server} \cdot P + Q_{Server} \cdot T_A \end{aligned} \tag{4}$$

In the same way:

$$\begin{aligned} K_{SA}^1 &= Q_{Server} \cdot T_A + (Q_{Server} + e_S) \cdot P_A \\ &= Q_{Server} \cdot T_A + Q_{Server} \cdot Q_A \cdot P + Q_A \cdot T_S \end{aligned} \tag{5}$$

Therefore, $K_{AB}^1 = K_{SA}^1$.

$$K_{AS}^2 = e_A \cdot T_S = e_A \cdot e_S \cdot P = T_A \cdot e_S = K_{SA}^2 \tag{6}$$

Hence, $SK_{AS} = SK_{SA}$, which means that user A and the server have the same session key.

2) SEND AND RECEIVE MESSAGES BETWEEN USER AND SERVER

After that, but before the user is offline, the session key SK_{AS} will be used to encrypt messages between the user and the server (not between users) with the AES algorithm. For example, suppose the user wants send plaintext M to the server. The process consists of the following steps:

a) Obtain ciphertext $C = \text{Encrypt}_{\text{AES}}(M, SK_{AS})$.

b) Obtain the current timestamp T_{time} and calculate number $m = H_0(T_{time}||IDA||C)$.

c) Select a number $k \in Z_q^*$ randomly such that $0 < k < q$ and calculate $R(x_R, y_R) = k \cdot P$. Then, mark $r = x_R$.

d) Calculate the number s by the user's private key Q_A with function $s = (m - Q_A \cdot r)k^{-1} \bmod q$.

e) Check r and s. If $r = 0$ or $s = 0$, repeat the process from step f.

f) Organize the message $<T_{time}, ID_A, C, r, s>$ in JSON format and send the message to the server.

When the server has received the message, the server will process it as follows:

a) The server calculates number $m = H_0(T_{time}||ID_A||C)$ and point $U(U_x, U_y) = s^{-1}(m \cdot P - r \cdot P_A)$. If $U_x = r$, proceed to the next step; otherwise, refuse to accept the message ($Ux \neq r$ means that the message has been falsified).

b) Search for the tuple $<ID_A, *>$ in L. If a tuple $<ID_A, T'_{time}>$ is found such that $T_{time} > T'_{time}$, update the tuple $<ID_A, T'_{time}>$ by $<ID_A, T_{time}>$ in L and proceed to the

next step (the server has accepted the message); otherwise, refuse to accept the message (it is a replay attack).

c) Calculate the plaintext M' by $M' = \text{Decrypt}_{\text{AES}}(C, SK_{SA})$. For $SK_{SA} = SK_{AS}$, the server will obtain the plaintext $M = M'$.

E. PROCESSES SUPPORT FOR OFFLINE KEY AGREEMENT AND MESSAGE ENCRYPTION BETWEEN USERS

For the IM scheme that is proposed in this paper, suppose that the message recipient is offline when the message sender is sending the message and only allows friends to send messages to each other. If user A wants to chat with user B, user A should request to be user B's friend.

1) REQUEST PROCESS FOR ADDING CONTACTS

The process for adding contacts in the proposed IM system is described in detail as follows:

a) User A obtains R_B from the IM server by searching ID_B of user B and calculates the public key P_B of user B by $P_B = R_B + H_1(ID_B, R_B) \cdot P_{sys}$.

b) User A chooses a number $e_A \in Z_q^*$ randomly such that $0 < e_A < q$ and calculates $T_A = e_A \cdot P$.

c) Take T_A as the key seed and obtain the key of the AES algorithm by utilizing function $AES_{key} = f(T_A)$.

d) Encrypt the plaintext request information M and obtain ciphertext $C = \text{Encrypt}_{\text{AES}}(M, AES_{key})$.

e) Calculate $R = e_A \cdot P_B$ and obtain the current timestamp T_{time}. Then, calculate number $m = H_0(T_{time}||ID_A||R_A||C||R)$.

f) Sign number m by carrying out steps c to e of the client in section ''Send and receive messages between user and server'' and obtain signature values r and s.

g) Organize the message $<T_{time}, ID_A, R_A, C, R, r, s>$ in JSON format and send the message to the server. The server will transfer the message to user B when user B is online.

When user B is online and has received the message from the server, user B will process it as follows:

a) User B calculates number $m = H_0(T_{time}||ID_A||R_A||C||R)$ and the public key P_A of user A such that $P_A = R_A + H_1(ID_A, R_A) \cdot P_{sys}$.

b) Verify the message by carrying out steps a) and b) of the server in section ''Send and receive messages between user and server''. If verification is successful, proceed to the next step.

c) User B calculates key seed T_A of the AES algorithm such that $T_A = Q_B^{-1} \cdot R$. Then, obtain the key of the AES algorithm by utilizing function $AES_{key} = f(T_A)$ and obtain the plaintext request information M by $M = \text{Decrypt}_{\text{AES}}(C, AES_{key})$. If the request is approved, proceed to the next step.

d) User B chooses a number $e_B \in Z_q^*$ randomly such that $0 < e_B < q$ and calculates point $T_B = e_B \cdot P$.

e) Obtain the current timestamp T_{time} and calculate number $m = H_0(T_{time}||ID_B||T_B)$. Sign number m by carrying out steps c e of the client in section ''Send and receive messages between user and server'' and obtain signature values r and s.

f) Organize the message $<T_{time}, ID_B, T_B, r, s>$ in JSON format and send the message to the server. The server will transfer the message to user A when user A is online.

g) Inform the server that users A and B have become friends with each other.

When user A is online and has received the message from the server, user A will process it by carrying out steps a and b of the server in section ''Send and receive messages between user and server''.

2) KEY AGREEMENT BETWEEN USERS

If verification is successful, normally, the session key should be calculated in the same way as the key agreement between user and server. However, consider the worst-case scenario that the sender and the recipient are never online at the same time. Therefore, it is difficult to clearly identify the beginning and end of a session. In this case, a scheme in which the ephemeral key is updated periodically is proposed in this section.

In this scheme, the IM client maintains lists $L_1 <ID_*, T_{time}, e_A, T_A>$ and $L_2 <ID_*, T_{time\_*}, T_A, T_*, SK_{A*}>$ for user A. For every recipient, the IM client will keep two tuples at most, to cover the case in which user A has just updated the ephemeral key e_A and the new $T_A = e_A \cdot P$ has not been transferred to the recipient. If the recipient wants to send a message to user A at this moment, the recipient will use session key SK_{*A}, which was generated by the old T_A, to encrypt the plaintext. If user A keeps only one tuple $<ID_*, T_{time}, e_A, T_A>$ in L_1 for the recipient, the old tuple will be replaced by the new tuple and user A will not be able to calculate the correct SK_{*A} for decrypting the message and obtaining the correct plaintext.

For any contact in the IM client to user A, there is a tuple $<ID_*, T_{time}, e_A, T_A>$ in L_1 and a tuple $<ID_*, T_{time\_*}, T_A, T_*, SK_{A*}>$ in L_2; refer to the section ''Request process for adding contacts''. Suppose that the sender is user A, the recipient is user B and the time period for updating the ephemeral key is P_{time}. Then, the f process for key agreement between users is described in detail as follows:

a) Search a tuple in L_2 by $<ID_B, *, *, *, *>$, which has a larger value of T_{time} and is denoted as tuple $<ID_B, T_{time\_B}, T_A, T_B, SK_{AB}>$.

b) Input the timestamp T'_{time} (which is obtained from the following section, namely, ''Message encryption between users'') and search tuples in L_1 by $<ID_B, *>$. If two tuples are obtained, go to step e; otherwise, determine whether $T'_{time} - T_{time} > P_{time}$ is satisfied (the tuple of the search result is $<ID_B, T_{time}, e_A, T_A>$).

c) If $T'_{time} - T_{time} > P_{time}$, user A chooses a number $e_{A1} \in Z_q^*$ randomly such that $0 < e_{A1} < q$ and calculates $T_{A1} = e_{A1} \cdot P$. Then, calculate the new session key SK_{AB1} as follows:

$$\begin{aligned} K_{AB}^1 &= Q_A \cdot T_B + (Q_A + e_{A1}) \cdot P_B \\ K_{AB}^2 &= e_{A1} \cdot T_B \\ SK_{AB1} &= H_2(ID_A, ID_B, T_{A1}, T_B, K_{AB}^1, K_{AB}^2) \end{aligned} \tag{7}$$

Add the tuple $<ID_B, T'_{time}, T_{A1}, T_B, SK_{AB1}>$ into L_2 and add the tuple $<ID_B, T'_{time}, e_{A1}, T_{A1}>$ into L_1. Go to step h.

d) If $T'_{time} - T_{time} < P_{time}$, the session key is SK_{AB} in the tuple $<ID_B, T_{time\_B}, T_A, T_B, SK_{AB}>$, which was found from L_2 in step a. Go to step h.

e) User A finds two tuples from L_1 in step b and selects the tuple $<ID_B, T_{time}, e_A, T_A>$ that has a bigger T_{time}, which is denoted as $<ID_B, T_{time\_big}, e_{A2}, T_{A2}>$. The other tuple is denoted as $<ID_B, T_{time\_small}, e_{A3}, T_{A3}>$.

f) If $T'_{time} - T_{time\_big} < P_{time}$, the session key is SK_{AB} in the tuple $<ID_B, T_{time\_B}, T_A, T_B, SK_{AB}>$, which was found from L_2 in step a. Go to step h.

g) If $T'_{time} - T_{time\_big} > P_{time}$, delete the tuple $<ID_B, T_{time\_small}, e_{A3}, T_{A3}>$ from L_1 and delete tuples $<ID_B, ^*, T_{A3}, ^*, ^*>$ from L_2. Then, repeat step c.

h) At this time, user A obtains tuple $<ID_B, T_{time\_B}, T_A, T_B, SK_{AB}>$ or $<ID_B, T'_{time}, T_{A1}, T_B, SK_{AB1}>$, which is denoted as $<ID_B, T_{time}, T_A, T_B, SK_{AB}>$. SK_{AB} will be used as a key to encrypt messages in the next section.

3) MESSAGE ENCRYPTION BETWEEN USERS

The scheme supports various message types, including text, document, picture, audio and video. Before the message encryption, the IM client will obtain the current timestamp T'_{time} to obtain the tuple $<ID_B, T_{time}, T_A, T_B, SK_{AB}>$.

For text messages, the IM client will obtain the encrypted chipertext $M = \text{Encrypt}(text, SK_{AB})$.

For messages of other types, the IM client will obtain the encrypted file $En_{file} = \text{Encrypt}(file, SK_{AB})$. Then, upload En_{file} to the IM server and obtain the download URL in the server. Organize the plaintext $text = <filename, \text{SHA1}(En_{file}), URL>$ in JSON format and obtain the encrypted chipertext $M = \text{Encrypt}(text, SK_{AB})$.

After that, the IM client calculates the number $m = H_0(T_{time}{}'||ID_A||T_A||T_B||message\_type||M)$. Then, sign the number m by carrying out steps c to e of the client in section "Send and receive messages between user and server" and obtain signature values r and s.

Finally, organize message $<T'_{time}, ID_A, T_A, T_B, message\_type, r, s>$ in JSON format and send the message to the server. The server will transfer the message to user B when user B is online.

4) MESSAGE DECRYPTION BETWEEN USERS

When user B is online and has received the message from the server, user B will process it as follows:

a) User B calculates number $m = H_0(T_{time}{}'||ID_A||T_A||T_B||message\_type||M)$.

b) Verify the message by carrying out steps a) and b) of the server in section "Send and receive messages between user and server". If verification is successful, proceed to the next step.

c) Search for tuple $<ID_A, T_{time}, T_A, T_B, SK_{BA}>$ in L_2 by $<ID_A, ^*, T_A, T_B, ^*>$. If it is found, obtain the key SK_{BA} for decrypting M and go to step e; otherwise, go to step d.

d) Search for tuple $<ID_A, T_{time}, e_B, T_B>$ in L_1 by $<ID_A, ^*, ^*, T_B>$. Then, calculate the new session key SK_{BA} as follows:

$$\begin{aligned} K^1_{BA} &= Q_B \cdot T_A + (Q_B + e_B) \cdot P_A \\ K^2_{BA} &= e_B \cdot T_A \\ SK_{BA} &= H_2(ID_B, ID_A, T_A, T_B, K^1_{BA}, K^2_{BA}) \end{aligned} \tag{8}$$

Add tuple $<ID_A, T'_{time}, T_B, T_A, SK_{BA}>$ into L_2.

e) Obtain the message type from *message_type*. If it is a text message, obtain the plaintext $text = \text{Decrypt}_{\text{AES}}(M, SK_{BA})$ and show the text to user B.

f) Otherwise, obtain plaintext text$<filename, hash_value, URL> = \text{Decrypt}_{\text{AES}}(M, SK_{BA})$ and show the filename to user B. After that, download file En_{file} through URL and obtain $hash\_value' = \text{SHA1}(En_{file})$.

g) If $hash\_value = hash\_value'$, obtain the original file by $file = \text{Decrypt}_{\text{AES}}(En_{file}, SK_{BA})$ and open the original file for user B. Otherwise, notify user B that En_{file} is not a correct file.

5) MESSAGE PROTECTION IN DEVICE

To satisfy the requirements of historic message reviewing and privacy protection, the chat messages and the system messages will be encrypted and stored in the device. In this paper, we use the AES algorithm to encrypt these messages and the encryption key is generated by $AES_{key} = f(ID_U, Q_U, Device_{ID})$. ID_U is the unique identifier of the user, Q_U is the private key of the user, which is generated in section "Setup of the identity-based public key cryptosystem", and $Device_{ID}$ is the unique identifier of the device, such as its IMEI.

For a message that is sent by the user, the IM client will encrypt the plaintext message and store it directly. For the message to be accepted by the user, the IM client will first obtain the plaintext message with session key SK and show the plaintext to the user. Then, the IM client will encrypt the plaintext message by $AES_{key} = f(ID_U, Q_U, Device_{ID})$ and store it.

Text messages can be organized as structured data and will be stored in the database in the device. However, other types of messages, such as photos, videos and documents, cannot be stored in the database directly; these encrypted files will be stored in a fixed folder in the storage of device and the original filename and path of each encrypted file will be organized as structured data and stored in the database in the device.

V. SECURITY AND ANALYSIS OF THE PROPOSED ENCHANCED SECURE IM SCHEME

A. LOGIN SECURITY BETWEEN USER AND SERVER

Theorem 1: The login process between user and server is secure under the ECDL assumption.

Proof: Assume that KGC has generated private keys for user U and the IM server, and the adversary is denoted as E. According to the login process in the previous section, we can prove the security as follows:

a) User U chooses a number e_A randomly, calculates $T_A = e_A \cdot P$, and utilizes T_A as a key seed to encrypt the login password.

b) Calculate $R = e_A \cdot P_{Server}$ and obtain the current timestamp T_{time}. Finally, sign the message with the user's private key Q_U by ECDSA and send the organized message to the server.

c) When the server has received the message, it will verify the signature and T_{time}. Then, the server will obtain a key seed with the server's private key Q_{Server} by $T'_A = Q^{-1}_{Server} \cdot R$. Since $R = e_A \cdot P_{Server}$, $T'_A = Q^{-1}_{Server} \cdot R = Q^{-1}_{Server} \cdot$ and $e_A \cdot P_{Server} = Q^{-1}_{Server} \cdot e_A \cdot Q_{Server} \cdot P = e_A \cdot P = T_A$. Finally, the server will obtain the login password by the key seed.

Adversary E can only obtain R and P_{server}. In this case, he cannot obtain key seed T_A to obtain the login password because calculating T_A without e_A or Q_{Server} is an ECDL problem. If adversary E resends the message or resends the message after falsifying T_{time}, the server will detect the forgery attack and replay attack according to the login process in the previous section.

Thus, the login process between user and server is secure under the ECDL assumption.

B. KEY AGREEMENT SECURITY BETWEEN USERS OR BETWEEN USER AND SERVER

For the key agreement processes between users and between user and server, the same equations are utilized to generate the session key with the ephemeral and private keys. Hence, although the ephemeral key is generated at the beginning of a session between a user and the server, whereas it is generated and updated periodically for processes between users, we can prove the security of key agreement between users or between a user and the server in the same way.

Theorem 2: The key agreement process is secure under the ECDL assumption and the CDH assumption.

Proof: Assume that KGC has generated private keys for user A and user B, and the adversary is denoted as E. According to the key agreement process in the previous section, we can prove the security as follows:

a) User A chooses a number e_A randomly and calculates $T_A = e_A \cdot P$. Then, obtain the current timestamp T_{time} and sign the message with the user's private key Q_A by ECDSA and send the organized message to user B.

b) When user B has received the message, he will verify the signature and T_{time}. Then, user B chooses a number e_B randomly, calculates $T_B = e_B \cdot P$ and obtains the current timestamp T_{time}. Finally, sign the message with the user's private key Q_B by ECDSA and send the organized message to user A.

c) When user A has received the message and verified the signature and T_{time}, he will obtain the shared session key SK_{AB} with user B as follows:

$$\begin{aligned} K^1_{AB} &= Q_A \cdot T_B + (Q_A + e_A) \cdot P_B \\ K^2_{AB} &= e_A \cdot T_B \\ SK_{AB} &= H_2(ID_A, ID_B, T_A, T_B, K^1_{AB}, K^2_{AB}) \end{aligned} \quad (9)$$

Adversary E can only obtain T_A, T_B, P_A and P_B. In this case, obtaining K^1_{AB} without Q_A and e_A should solve the ECDL problem and obtaining K^2_{AB} with T_A and T_B should solve the CDH problem. In addition, users A and B can detect forgery and replay attacks by ECDSA and T_{time}.

Thus, the key agreement process between users or between user and server is secure under the ECDL assumption and the CDH assumption.

C. COMPARISON OF THE PROPOSED SCHEME WITH OTHERS

In this section, we compare the proposed enhanced IM scheme with others in terms of applicability and security, such as denial of replaying attack, denial of forgery attack, support for offline key agreement and privacy protection on the device. We present comparison results in Table 1.

TABLE 1. Results of comparison of the proposed scheme with others.

	Ref. [9]	Ref. [11]	Ref. [16]	Our scheme
Denial of replaying attack	×	√	×	√
Denial of forgery attack	√	×	√	√
Offline key agreement	×	×	√	√
Privacy protection on device	×	×	√	√
Multimedia & document message	Not mentioned	×	Not mentioned	√

The comparison results in Table 1 show that the enhanced secure IM scheme that is proposed in this paper is a comprehensive secure scheme with high security and practicability.

D. PERFORMANCE EFFICIENCY AND EXPERIMENTAL ANALYSIS

By analyzing the newly proposed IM scheme, we can obtain that the computationally heavy operations in the scheme are session key updating, message encryption, signature, message decryption, and signature verification. To explain the performance efficiency of the new scheme more intuitively, we conduct some experiments on a laptop and an android smart phone based on Java Developer's Kit (JDK) and Java Pairing Based Cryptography Library (JPBC) [28]. The laptop has an Intel CPU at 2.4 GHz, 12 GB memory and Windows 7. The android smart phone is Huawei Honor 8 with an 8-core processor, 3 GB memory, and Android 7.0.

Fig. 2 shows the computational overhead of the AES algorithm for encryption on an android smart phone, as we chose the AES symmetric key encryption algorithm to encrypt messages in the new scheme. The average encryption speed for AES on the test phone is approximately 7.32 Kb/ms. The IM scheme is a short message exchange scheme. Therefore, for a text message in the IM scheme, the size of the message content rarely exceeds 1 Kb. Hence, the encryption time

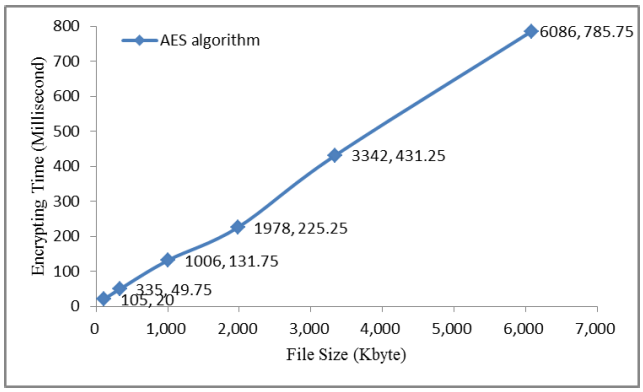


FIGURE 2. AES algorithm computational overhead for encryption on a mobile device.

for the text message rarely exceeds 0.14 ms. However, for multimedia and document messages, the time that it takes to encrypt a message depends on the file size.

TABLE 2. Computational overhead for different operations of elliptic curves on a laptop and a mobile device.

	d(159)	d(201)	d(224)
Pairing computation on mobile device	1095.9ms	1245.7ms	1517.2ms
Pairing computation on laptop	25.25ms	32.43ms	35.15ms
Exponentiation computation on mobile device	37.3ms	53.05ms	62.43ms
Exponentiation computation on laptop	2.43ms	3.78ms	4.68ms
Multiplication computation on mobile device	42.9ms	52.15ms	57.38ms
Multiplication computation on laptop	2.35ms	3.7ms	4.73ms

Table 2 shows the computational overheads for different elliptic curve operations on the laptop and the android smart phone, as the session key agreement protocol and signature algorithm are designed based on elliptic curves. We use the d(n)-type elliptic curve group, which is good for cryptosystems when group elements must be as short as possible. In each case, the curve group has a 160-bit group order and d(n) denotes that the base field size is n bits [29]. According to Table 2, the computational overhead of the exponentiation operation is equivalent to that of the multiplication operation, but the overhead of the pairing operation is much larger than those of the exponentiation and multiplication operations, especially on the mobile device. Although n=159 for the elliptic curve group, a pairing operation will take 1095.9 ms. Thus, the pairing operation is not friendly for mobile devices.

Assume that we use an elliptic curve group such that d(159) and the IM user has calculated and stored the public keys of his or her friend. The session key agreement process in the new scheme has 4 multiplication operations, which will take approximately 171.6 ms in total. For the signature algorithm, the signature process has 1 multiplication operation and will take approximately 42.9 ms, while the signature verification process has 2 multiplication operations and will take approximately 85.8 ms.

In summary, the experimental results show that we should not use pairing operations in mobile social networks due to the large computational overhead. For a text message in the new scheme, the process overhead depends mainly on the signature, as the encryption overhead can be ignored compared with the signature overhead. However, for multimedia and document messages, the process overhead depends mainly on the file size and signature.

VI. CONCLUSIONS

To overcome the data security and privacy protection problems of mobile social networks, an enhanced secure Instant Messaging system that is based on the Elliptic Curve Cryptosystem was proposed in this paper. The proposed scheme supports offline key agreement between users and data privacy protection on device. In addition, the scheme utilizes timestamps to deny replaying attacks and utilizes ECDSA to sign and verify the messages that are transferred in the IM system. The comparison results of the proposed scheme with others and the results of an experiment show that it is a comprehensive secure scheme with high security and good practicability.

REFERENCES

[1] A. P. Oghuma, C. F. Libaque-Saenz, S. F. Wong, and Y. Chang, "An expectation-confirmation model of continuance intention to use mobile instant messaging," *Telematics Informat.*, vol. 33, no. 1, pp. 34–47, 2016.
[2] R. Alkhulaiwi, A. Sabur, K. Aldughayem, and O. Almanna, "Survey of secure anonymous peer to peer Instant Messaging protocols," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust (PST)*, Dec. 2016, pp. 294–300.
[3] N. B. A. Barghuthi and H. Said, "Social networks IM forensics: Encryption analysis," *J. Commun.*, vol. 8, no. 11, pp. 708–715, 2013.
[4] S. Muftic, N. bin Abdullah, and I. Kounelis, "Business information exchange system with security, privacy, and anonymity," *J. Electr. Comput. Eng.*, vol. 2016, Feb. 2016. Art. no. 7093642.
[5] C. Anglano, M. Canonico, and M. Guazzone, "Forensic analysis of the ChatSecure instant messaging application on android smartphones," *Digit. Invest.*, vol. 19, pp. 44–59, Dec. 2016.
[6] S. Puuska, M. J. Kortelainen, V. V. Venekoski, and J. Vankka, "Instant message classification in Finnish cyber security themed free-form discussion," *Ijcsa*, vol. 1, no. 1, pp. 97–109, 2016.
[7] Z. Zhang and B. B. Gupta, "Social media security and trustworthiness: Overview and new direction," *Future Generat. Comput. Syst.*, Oct. 2016. [Online]. Available: https://doi.org/ 10.1016/j.future.2016.10.007
[8] S. Kungpisdan and N. Moonviriyakit, "A highly secure instant messaging protocol," in *Proc. 14th World Multiconf. Syst., Cybern. Informat. (WMSCI)*, Orlando, FL, USA, 2010, pp. 155–160.
[9] M. Luo, L. Cai, Y. Ji, and P. Huang, "A secure instant messaging scheme based on certificateless signcryption," *J. Comput. Inf. Syst.*, vol. 10, no. 14, pp. 6067–6074, 2014.
[10] M. K. Yusof and A. F. A. Abidin, "A secure private instant messenger," in *Proc. 17th Asia–Pacific Conf. Commun.*, Kota Kinabalu, Malaysia, 2011, pp. 821–825.
[11] I. Del Pozo and M. Iturralde, "CI: A new encryption mechanism for instant messaging in mobile devices," *Procedia Comput. Sci.*, vol. 63, pp. 533–538, Jan. 2015.
[12] C.-J. Wang, W.-L. Lin, and H.-T. Lin, "Design of an instant messaging system using identity based cryptosystems," in *Proc. 4th Int. Conf. Emerg. Intell. Data Web Technol.* Xi'an, China, 2013, pp. 277–281.

[13] P. P. Wanda and B. S. Hantono, "Efficient message security based Hyper Elliptic Curve Cryptosystem (HECC) for mobile instant messenger," in *Proc. 1st Int. Conf. Inf. Technol., Comput. Electr. Eng.*, Semarang, Indonesia, 2014, pp. 245–249.

[14] A. Loukas, D. Damopoulos, S. Menesidou, M. Skarkala, and G. Kambourakis, "MILC: A secure and privacy-preserving mobile instant locator with chatting," *Inf. Syst. Frontiers*, vol. 14, no. 3, pp. 481–497, 2012.

[15] P. Wanda and B. S. Hantono, "Model of secure P2P mobile instant messaging based on virtual network," in *Proc. Int. Conf. Inf. Technol. Syst. Innov.* Bandung, Indonesia, 2015, pp. 81–85.

[16] T.-Y. Tung, L. Lin, and D. T. Lee, "Pandora messaging: An enhanced self-message-destructing secure instant messaging architecture for mobile devices," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Fukuoka, Japan, Mar. 2012, pp. 720–725.

[17] M. H. Eldefrawy, K. Alghathbar, M. K. Khan, and H. Elkamchouchi, "Secure instant messaging protocol for centralized communication group," in *Proc. 4th IFIP Int. Conf. New Technol., Mobility Secur.*, Paris, France, 2011, pp. 1–4.

[18] A. Ruiz-Martínez and C. Inmaculada Marín-Lòpez, "SIPmsign: A lightweight mobile signature service based on the Session Initiation Protocol," *Softw. Pract. Exper.*, vol. 44, no. 5, pp. 511–535, 2014.

[19] I. Karabey and G. Akman, "A cryptographic approach for secure client-server chat application using public key infrastructure (PKI)," in *Proc. 11th Int. Technol. Secur. Trans. (ICITST)*, Barcelona, Spain, 2016, pp. 442–446.

[20] H. C. Chen, H. Wijayanto, C. H. Chang, F. Y. Leu, and K. Yim, "Secure mobile instant messaging key exchanging protocol with one-time-pad substitution transposition cryptosystem," in *Proc. Comput. Commun. Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, 2016, pp. 980–984.

[21] H. B. Qin and X. Xu, "Solution to secure Instant Messaging based on hardware encryption," in *Proc. IEEE 16th Int. Conf. Commun. Technol. (ICCT)*, Shenyang, China, Oct. 2015, pp. 844–847.

[22] W. Feng, Z. Zhang, J. Wang, and L. Han, "A novel authorization delegation scheme for multimedia social networks by using proxy re-encryption," *Multimedia Tools Appl.*, vol. 75, no. 21, pp. 13995–14014, 2016.

[23] M. Xie, Z. Wu, and H. Wang, "Secure instant messaging in enterprise-like networks," *Comput. Netw.*, vol. 56, no. 1, pp. 448–461, 2012.

[24] B. C. Li and S. Z. Yu, "Keyword mining for private protocols tunneled over websocket," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1337–1340, Jul. 2016.

[25] G. Fahrnberger, "SIMS: A comprehensive approach for a secure instant messaging sifter," in *Proc. IEEE, Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Xi'an, China, Jun. 2015, pp. 164–173.

[26] E. S. M. El-Alfy and A. A. AlHasan, "Spam filtering framework for multimodal mobile communication based on dendritic cell algorithm," *Future Generat. Comput. Syst.*, vol. 64, pp. 98–107, Nov. 2016.

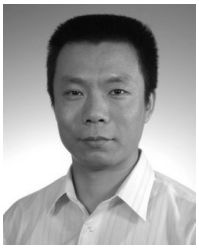
[27] *SM2/SM3/SM4, China Cryptography Specification, State Cryptography Administration*. Accessed: 2010. [Online]. Available: http://www.oscca.gov.cn

[28] A. de Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Comput. Commun.*, Jun. 2011, pp. 850–855.

[29] B. Lynn. *The Pairing-Based Cryptography Library*, Accessed: 2013. [Online]. Available: http://crypto.stanford.edu/pbc/

ZHEN WANG was born in 1989. He is currently pursuing the Ph.D. degree with the School of Cyber Security, Beijing University of Posts and Telecommunications. His research interests include mobile network security and digital rights management.

ZHAOFENG MA was born in 1974. He received the Ph.D. degree from Xi'an Jiaotong University in 2004. He was involved in post-doctoral research with Tsinghua University from 2005 to 2007. Since 2007, he has been engaged in education and research work with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include information security, digital rights management, and blockchain.



SHOUSHAN LUO was born in 1962. He is currently a Professor and a Ph.D. Supervisor with the School of Cyber Security, Beijing University of Posts and Telecommunications. His research interests include encode cryptography and network and information security.

HONGMIN GAO was born in 1987. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing University of Posts and Telecommunications. His research interests include mobile network security and digital rights management.