International Journal of Hybrid Information Technology
Vol.9, No.2 (2016), pp. 123-136
http://dx.doi.org/10.14257/ijhit.2016.9.2.11

On the Security of a Group Key Agreement Protocol and Its Improvement with Pairings

Xiangjun Xin[1], Chaoyang Li[1], Dongsheng Chen[1] and Fagen Li[2]

[1]*School of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou 450002, China*
[2]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
xin_xiang_jun@126.com

Abstract

In the paper, we analyze the security vulnerability of the key agreement protocol proposed by Lee et al.'s. We present a forgery attack to their protocol. In this attack, the adversary can modify the signed message and forge a new signature, which can pass the verification. Then, we propose a new group key agreement protocol, which overcomes this security drawback. The new protocol can be proved to be secure under Elliptic Curve Discrete Logarithm Problem, Bilinear Computational Diffie–Hellman Problem and Square-Exponent Problem. On the other hand, in the new protocol, only three pairing operations are used, so it is more efficient. Our protocol is also a contributory group key agreement protocol.

Keywords: *group key agreement, security, bilinear pairing, mobile communication*

1. Introduction

Mobile communication greatly facilitates communication between mobile users. With the help of portable devices, such as cellular phones and personal digital assistants, the users can freely roam and enjoy mobile services.

Now, group communication is an important research issue for mobile communication. Secure mobile communication should guarantee the confidentiality and authentication for the mobile users and the communication messages. A group key agreement protocol can be used to realize the secure group communication in a mobile environment. In a group key agreement protocol, all participants cooperatively establish the group key. The communicating parties can use the group key together with standard cryptographic algorithms for message encryption and authentication in order to preserve privacy and authentication. A secure authenticated group key agreement in a mobile environment can guarantee the authentication for legitimate group members, and it also can guarantee the secure intergroup communication from nonmembers. On the other hand, one advantage of the contributory group key agreement protocol is that no participant can control the final value of the group key. Therefore, a contributory group key agreement is often used to prevent some parties from having any kind of advantage over the others. In this paper, we focus on the secure contributory group key agreement protocols.

Recently, many group key agreement protocols have been proposed [1-13], which can also be classified into two kinds: the static [1-5] and the dynamic [6-13]. The difference between the former and the later is whether the users can join or leave the group at any time. In 1996, the method of natural extensions of Diffie-Hellman key exchange to n-party case was proposed [6]. It was useful to construct the key agreement protocols for the dynamic group. Based on Diffie-Hellman key agreement, a key agreement for highly dynamic group was developed by Steiner, *et al.*, [7]. However, in [7], the security services

ISSN: 1738-9968 IJHIT

such as key integrity, entity authentication, non-repudiation and access control were ignored. In 1999, a conference scheme for providing dynamic participation was proposed [8]. But, this scheme was proved not to be secure against eavesdropping and impersonation [9]. Then, in 2004, based on ElGamal encryption scheme [14] and secret sharing techniques [15], a group key agreement protocol for imbalanced wireless networks was proposed by Bresson, *et al.,* [10]. However, their protocol didn't achieve perfect forward secrecy. At the same time, the protocol in [10] was not a contributory group key agreement protocol [5, 11]. By blending key trees with Diffie-Hellman key exchange, Kim, Perrig, and Tsudik proposed a novel key agreement approach for dynamic group [12]. The main feature of the protocols in [12] was the use of key trees to efficiently compute and update group keys. But, in [12], to make the protocols to be authenticated ones, all the transmitted messages had to be signed and verified by using some public key signatures such as DSA or RSA. Later, in 2008, Dutta and Barua proposed a new contributory group key agreement protocol in dynamic setting [13]. Unfortunately, in 2010, their protocol was proved to be insecure [16].

All the group key agreement protocols mentioned above were based on public key certificate. On the other hand, some of ID-based group key agreements were also proposed [17-19]. However, compared with the certificate-based protocols, the ID-based ones have to introduce the key generate centers(KGC) to generate private keys for all the users, and all the users must trust the KGC, which makes the ID-based protocols are only suitable for application in a closed environment. Then, in this paper, we mainly discuss the construction of the secure contributory certificate-based group key agreement in the static case. In fact, according to the recent research, it is found that most of the static certificate-based key agreement protocols have one shortcoming or another. For example, Asokan et al. proposed two kinds of group key agreement protocols [1], both of which were only suited for a small group of powerful devices. In their protocols, the asymmetric public key encryption and decryption algorithms had to be used by every participant. On the other hand, in the first kind of protocol, four rounds were used, and in the second one the rounds of protocol were linear with the number of the participants. In 2000, the protocol with single round was proposed by Boyd, *et al.,* [2]. However, in [2], the computational complexity was linear with the number of the participants, too. What is more, Boyd et al.'s protocol had not forward secrecy. In 2005, Nam et al presented a three-round group key agreement for a mobile environment [3]. But Tseng demonstrated that Nam's three-round protocol was not a contributory group key agreement protocol. Then, Tseng proposed a new group key agreement protocol for an imbalanced network [4]. Unfortunately, Tseng's protocol was a nonauthenticated protocol [5]. Then, in 2009, based on the bilinear pairing [20, 21], Lee presented a new group key agreement protocol for an imbalanced mobile environment to overcome the security drawback of Tseng's protocol [5]. In the imbalanced mobile environment, the systems shift the computational burden from the mobile users to the powerful node. However, in this paper, we prove that Lee's protocol is still not an authenticated protocol, since the transmitted messages can be modified and forged. Then, in this paper, a new authenticated group key agreement is proposed, which can be proved to be secure under the hardness assumption of Bilinear Discrete Logarithm Problem, Bilinear Computational Diffie–Hellman Problem and Square-Exponent Problem. Different from most of the static key agreement protocols above, our protocol is an authenticated one, and it has perfect forward and backward secrecy, since the session common group keys are independent each other from different session. Our protocol is also proven to be a contributory group key agreement protocol. On the other hand, in our paper, the authenticators are used instead of pairing operations. The authenticators can reduce the pairing operations and make our protocol more efficient than the one proposed by Lee. Here, the authenticator is not an entity but a triple (U_i, A_i, c_i), where U_i denotes the identity of a mobile client, and A_i is an element generated at random in an additive cyclic group G_1 by using Diffie-Hellman key exchange technique.

c_i, which is generated by using keyed cryptographic hash function H_1, can be seemed as the message authentication code for U_i and A_i. More security and performance analysis will be discussed in Section 4.

The rest of this paper is organized as follows: In Section 2, some basic knowledge and Lee's group key agreement protocol are briefly reviewed, and the security weakness of Lee's protocol is proved, too. In Section 3, we present a new contributory group key agreement protocol, whose correctness, security, and performances are analyzed in Section 4. In Section 5, we conclude.

2. Preliminary

2.1. Bilinear Pairings

Let λ be a security parameter. The pairing is defined as e: $G_1 \times G_1 \rightarrow G_2$, where G_1 is an additive cyclic group of prime order q, and G_2 is a multiplicative cyclic group of the same order and P is an arbitrary generator of G_1. A cryptographic bilinear pairing has the following properties:

Bilinear: For any $R, S \in G_1$ and $a, b \in Z_q^*$, $e(aR, bS) = e(R, S)^{ab}$. This can be restated as, for any $R, S, T \in G_1$, $e(R+S, T) = e(R, T)e(S, T)$ and $e(R, S+T) = e(R, S)e(R, T)$.

Non-degenerate: There exists $R, S \in G_1$ such that $e(R, S) \neq I$, where I denotes the identity element of the group G_2.

Computable: Given $R, S \in G_1$, there exists an efficient algorithm to compute $e(R, S)$.

The bilinear parings can be derived from the Weil or Tate pairing [20, 21].

Definition 1 The Elliptic Curve Discrete Logarithm Problem (ECDLP) in G_1 is defined as: Given the generator P of G_1 and $Q \in G_1$, compute $a \in Z_q^*$ such that $Q = aP$.

The ECDLP in G_1 is assumed to be computationally hard and can be efficiently reduced to DLP in G_2 [22].

Definition 2 Given a generator P of a group G_1 and a random triple (P, aP, bP), the Bilinear Computational Diffie-Hellman Problem (BCDHP) is to compute abP.

Assumption: In our paper, we always assume that ECDLP and BCDHP are hard computational problems such that there is no polynomial time algorithm to solve either of them.

2.2. Review of Lee's Group Key Agreement Protocol

In this section, Lee's group key agreement protocol [5] is briefly reviewed. Without loss of generality, let $U=\{U_1, U_2, ..., U_n\}$ be the initial set of participants that want to generate a common group key. Let $X_i \in Z_q^*$ and $Y_i(=X_iP)$ be U_i's long-term private key and long-term public key, respectively. Here, some notations used in Lee's protocol are shown in Table 1. The steps of Lee's group key agreement protocol are as follows:

Step 1 (Round 1) First, each U_i ($1 \leq i \leq n-1$) selects a random number $a_i \in Z_q^*$ and then computes a_i^{-1} and $A_i=a_iP$. Then, each U_i can generate the signature $S_i=X_iA_i$ and send the triple (U_i, A_i, S_i) to the powerful node U_n.

Step 2 (Round 2) After receiving each (U_i, A_i, S_i) ($1 \leq i \leq n-1$), U_n verifies $e(S_i, P)=e(A_i, Y_i)$. If it holds, U_n can ensure that (U_i, A_i, S_i) is sent by U_i. Then U_n selects a random number $a_n \in Z_q^*$ and computes $x_i=a_nA_i$. Then, U_n computes $B=H(U_n, x_1, x_2, ..., x_{n-1})$ and $S_n=X_nB$. Next, U_n can compute the common group key $K=e(a_nP, \sum_{i=1}^{n-1} x_i)$. Finally, U_n broadcasts (U_n, x_1, x_2, ..., x_{n-1}, S_n) to other nodes.

Step 3 (Common group key) After receiving the broadcast, each U_j ($1 \leq j \leq n-1$) computes $B=H(U_n, x_1, x_2, ..., x_{n-1})$ and verifies whether $e(S_n, P)=e(B, Y_n)$. If it is correct, each U_j can ensure that (U_n, x_1, x_2, ..., x_{n-1}) are sent by U_n. Then, each U_j ($1 \le j \le n-1$) can compute the common group key $K=e(x_j a_j^{-1}, \sum_{i=1}^{n-1} x_i)$.

Table 1. Notations in Lee's Protocol

Notations	Description
q, G_1, G_2, P, e	Public parameters as that described in section 2.1
a_i	A random number $\in Z_q^*$, ($1 \le i \le n$)
n	The number of participants involved in generating a common conference key
U_i	The participants, ($1 \le i \le n$)
U_n	The powerful node with less restriction
$U_1, U_2, ..., U_{n-1}$	The mobile devices with limited computing capability
H	A map-to-point hash function from $\{0, 1\}^*$ to G_1
X_i	Long-term private key $\in Z_q^*$, ($1 \le i \le n$)
Y_i	Long-term public key $Y_i(=X_iP)$

2.3. Security Weakness of Lee's Protocol

We prove that Lee's protocol is still not an authenticated protocol due to the forgery of the messages in round 1. In fact, in round 1, the transmitted and signed messages (U_i, A_i, S_i) ($1 \le i \le n-1$) can be modified and forged, where $S_i=X_iA_i$, and X_i is long-term private key of U_i with the corresponding public key Y_i. Given a triple (U_i, A_i, S_i), an adversary can modify it and forge a valid signed message by using the steps as follows:

Step 1 First, the adversary selects a random number $b_i \in Z_q^*$ and computes $A_i^*=b_iA_i$, and then forges the signature $S_i^*=b_iS_i$ on A_i^*.

Step 2 Then, the adversary sends the triple (U_i, A_i^*, S_i^*) to the powerful node U_n.

Based on the analysis above, we have the following theorem.

Theorem 2.1 Given a transmitted signature (U_i, A_i, S_i), an adversary can modify it and forge a new signature (U_i, A_i^*, S_i^*), which can pass the verification in round 2.

Proof According to the round 2, we know that a valid signature (U_i, A_i, S_i) satisfies the equation $e(S_i, P)=e(A_i,Y_i)$. Once the triple (U_i, A_i, S_i) is intercepted by an adversary, he can execute the steps above-mentioned so as to modify the transmitted message and forge a new signature. Note that $A_i^*=b_iA_i$ and $S_i^*=b_iS_i$. Therefore, using the bilinear property of the pairing, we have

$$e(S_i^*, P)= e(b_iS_i, P)=e(b_iA_i, Y_i)=e(A_i^*,Y_i).$$

Thereby, the modified and forged signature (U_i, A_i^*, S_i^*) by the adversary can pass the verification in round 2. This also makes that the victim U_i can not refuse the forged signature (U_i, A_i^*, S_i^*).

According to Theorem 2.1, it is known that the signature (U_i, A_i, S_i) can be modified and forged. So, in round 2, once the powerful node receives a signature, it cannot ensure whether the signature is generated by U_i or forged by the adversary. It makes that the transmitted and signed messages cannot be authenticated by U_n. What is more, once U_n receives the triples (U_1, A_1, S_1), (U_2, A_2, S_2), …, (U_{i-1}, A_{i-1}, S_{i-1}), (U_i, A_i^*, S_i^*), (U_{i+1}, A_{i+1}, S_{i+1}), …, (U_{n-1}, A_{n-1}, S_{n-1}), where (U_i, A_i^*, S_i^*) is a modified and forged signature, which can pass the verification, U_n will compute and broadcast (U_n, x_1, x_2, ..., x_{i-1}, x_i^*, x_{i+1}, ..., x_{n-1}, S_n) to other nodes, where $x_t=a_nA_t(1 \le t \le n-1,\ t \ne i)$, $x_i^*=a_nA_i^*$, $A_i^*=b_iA_i$, $S_i^*=b_iS_i$, $B=H(U_n, x_1, x_2, ..., x_{i-1}, x_i^*, x_{i+1}, ..., x_{n-1})$ and $S_n=X_nB$. Then, for the user U_i, the common group key should be

$$K=e(x_i^* a_i^{-1} b_i^{-1},\ x_i^* + \sum_{j=1, j \ne i}^{n-1} x_j).$$

Note that the forged signature (U_i, A_i^*, S_i^*) is a valid signature and $b_i \in Z_q^*$ is selected at random by the adversary. So the victim user U_i cannot refuse (U_i, A_i^*, S_i^*), and he also does not know b_i. Therefore, the victim user U_i cannot compute the common group key

$$K=e(x_i a_i^{-1} b_i^{-1},\ x_i^* + \sum\nolimits_{j=1, j\neq i}^{n-1} x_j\ ),$$

since U_i knows nothing about b_i. Without the common group key K, the victim node U_i cannot communicate securely with the other nodes.

According to the analysis above, it is found that Lee's protocol cannot authenticate the validity of transmitted data, and the victim U_i can neither compute the common group key nor communicate with the other nodes securely.

2.4. Efficiency Analysis of Lee's Protocol

To analyze the efficiency of Lee's protocol from pairings, we mainly analyze the pairing operations. In a pairing-based scheme, compared with the other operations, the pairing operation is the most time-consuming [21]. According to the best result [23], one pairing operation is about 11110 multiplications in $F_{3^{163}}$, while a point scalar multiplication of E/$F_{3^{163}}$ is a few hundred multiplications in $F_{3^{163}}$. Then, in a pairing-based scheme, the pairing operations should be less used. However, in Lee's authenticated protocol, to authenticate the signatures sent from the mobile users, U_n has to finish computing 2n-1 bilinear pairings. That is, in Lee's protocol, for the node U_n, the computing burden of the bilinear pairing has a linear relation with the numbers of mobile users. Therefore, to make the key agreement more efficient, in the next section, we improve the protocol such that the pairing operations of U_n are independent of the number of the mobile users. In fact, in our protocol, U_n only needs to compute two pairing operations.

3. New Construction of Authenticated Key Agreement Protocol in Imbalanced Mobile Environment

To overcome the security weakness and improve the efficiency of Lee's protocol, we present a new key agreement protocol in an imbalanced mobile environment. Here, for ease of making a performance comparison in Section 4.3, we extend Lee's notations in Table 2. The detailed steps of our protocol are described as follows.

Step 1 (Round 1) First, each U_i ($1 \leq i \leq n-1$) selects a random number $a_i \in Z_q^*$ and then computes $A_i = a_i X_i Y_n$, $k_i = H_1(a_i Y_i)$ and $c_i = Encrypt_{k_i}(a_i)$. Then, each U_i sends the authenticator (U_i, A_i, c_i) to the powerful node U_n. Note that X_i^{-1}, a_i^{-1}, $(a_i X_i)^{-1}$ and the pair (A_i, c_i) can be precomputed so as to reduce the computational cost. So, every node can precompute (X_i^{-1}, a_i^{-1}, $(a_i X_i)^{-1}$, a_i, A_i, c_i) off-line and store them on its own memory card.

Step 2 (Round 2) After receiving each authenticator (U_i, A_i, c_i) ($1 \leq i \leq n-1$), U_n computes $B_i = A_i X_n^{-1}$, $k_i = H_1(B_i)$ and $a_i = Decrypt_{k_i}(c_i)$. Then, U_n verifies whether $B_i = a_i Y_i$. If it holds, U_n can ensure that (U_i, A_i, c_i) ($1 \leq i \leq n-1$) are sent by U_i. Then, U_n selects a random number $a_n \in Z_q^*$ and computes $D_i = a_n B_i$, $h = H_2(U_n, D_1, D_2, ..., D_{n-1})$ and $S = P/(X_n + h)$. Next, U_n can compute the common group key $K = e(a_n P, \sum\nolimits_{i=1}^{n-1} D_i)$. Finally, U_n broadcasts (U_n, D_1, D_2, ..., D_{n-1}, S) to other nodes.

Step 3 (Common group key) After receiving the broadcast, each U_j computes $h = H_2(U_n, D_1, D_2, ..., D_{n-1})$ and verifies whether $e(S, Y_n + hP) = g$. If it is correct, each U_j can ensure that (U_n, D_1, D_2, ..., D_{n-1}) are sent by U_n. Then, each U_j can compute the common group key

$$K = e((a_j X_j)^{-1} D_j, \sum\nolimits_{i=1}^{n-1} D_i).$$

Table 2. Notations in Our Protocol

Notations	Description
λ, q, G_1, G_2, P, e	Public parameters as that described in section 2.1
g	Public parameter in G_2 such that $g=e(P, P)$, which can be precomputed
a_i	A random number $\in Z_q^*$, $(1\le i\le n)$
k_i	A symmetric key shared by U_i and U_n $(1\le i\le n\text{-}1)$
c_i	The ciphertext of a_i
$Encrypt_{k_i}(\,)$	Secure standard symmetric encryption algorithm, where k_i is the symmetric key used in this algorithm
$Decrypt_{k_i}(\,)$	Decryption algorithm corresponding to $Encrypt_{k_i}(\,)$
n	The number of participants involved in generating a common conference key
U_n	The powerful node with less restriction
$U_1, U_2, ..., U_{n-1}$	The mobile devices with limited computing capability
H_1	A secure cryptographic hash function from G_1 to $\{0, 1\}^\lambda$
H_2	A secure cryptographic hash function from $\{0, 1\}^*$ to Z_q^*
X_i	Long-term private key in Z_q^* $(1\le i\le n)$
Y_i	Long-term public key $Y_i(=X_iP)$

Our authenticated protocol can be proved to be a contributory group key agreement protocol. We show the proof as follows.

Theorem 3.1 By running the proposed protocol, an identical group key can be established by all mobile clients. Each client can confirm that its contribution was included in the group key.

Proof. In our protocol, U_n broadcasts $(U_n, D_1, D_2, ..., D_{n-1})$ to all mobile clients, and each client U_i $(1\le i\le n-1)$ can use its long-term private key X_i and the secret number a_i to compute an identical group key K. That is, the following equations hold:

$$\begin{aligned} K &= e((a_1X_1)^{-1}D_1, \textstyle\sum_{i=1}^{n-1}D_i) \\ &= e((a_2X_2)^{-1}D_2, \textstyle\sum_{i=1}^{n-1}D_i) \\ &\vdots \\ &= e((a_{n-1}X_{n-1})^{-1}D_{n-1}, \textstyle\sum_{i=1}^{n-1}D_i). \end{aligned}$$

Due to the bilinear properties of the bilinear pairing we have $V=(a_jX_j)^{-1}D_j$ for $1\le j\le n-1$ such that $K=e(V, \sum_{i=1}^{n-1}D_i)$. Therefore, we have:

$$\begin{gathered} D_1= Va_1X_1, \\ D_2= Va_2X_2, \\ \vdots \\ D_{n-1}=Va_{n-1}\,X_{n\text{-}1}. \end{gathered}$$

Observing the above equations, each D_i includes the long-term private key X_i and secret number a_i of U_i . Therefore, the group key K contains all clients' long-term private key X_i and secret number a_i. That is, each client can confirm that its contribution was included in the group key.

4. Discussions

In this section, first, we prove the correctness of our protocol. Then, the security of our protocol is analyzed. At last, the performance comparison of our protocol with the old ones is presented.

4.1. Correctness

The correctness of our protocol is proved in the following theorems.

Theorem 4.1 In our protocol, all participants can establish and share an identical common group key K. That is, they can compute the common group key K by using the equations as follows:

$$K = e(a_n P, \sum_{i=1}^{n-1} D_i) = e((a_j X_j)^{-1} D_j, \sum_{i=1}^{n-1} D_i),$$

where 1≤j≤n−1.

Proof. According to our protocol, it is known that $Y_n=X_nP$, $A_j=a_jX_jY_n$, $B_j=A_jX_n^{-1}$ and $D_j=a_nB_j$. Using the properties of the bilinear pairing, we have

$$\begin{aligned} K &= e((a_j X_j)^{-1} D_j, \sum_{i=1}^{n-1} D_i) \\ &= e(a_j^{-1} X_j^{-1} a_n B_j, \sum_{i=1}^{n-1} D_i) \\ &= e(a_j^{-1} X_j^{-1} a_n A_j X_n^{-1}, \sum_{i=1}^{n-1} D_i) \\ &= e(a_j^{-1} X_j^{-1} a_n a_j X_j Y_n X_n^{-1}, \sum_{i=1}^{n-1} D_i) \\ &= e(a_n P, \sum_{i=1}^{n-1} D_i). \end{aligned}$$

This implies that the powerful node U_n and other mobile clients U_j (1≤j≤n−1) can share the common group key K.

Theorem 4.2 Given (U_i, A_i, c_i), U_n computes $B_i=A_iX_n^{-1}$, $k_i=H_1(B_i)$ and $a_i=Decrypt_{k_i}(c_i)$. Then, U_n verifies whether $B_i=a_iY_i$. If it holds, the powerful node U_n can verify the authenticator (U_i, A_i, c_i) (1≤i≤n−1) sent from U_i (1≤i≤n−1).

Proof. Note that the long-term private key X_n is only mastered by the powerful node U_n. Therefore, in round 2, only the powerful node U_n can compute $B_i=A_iX_n^{-1}$, $k_i=H_1(B_i)$ and $a_i=Decrypt_{k_i}(c_i)$. On the other hand, according to round 1, k_i is a valid symmetric key only if $k_i=H_1(a_iY_i)$. At the same time, $k_i=H_1(B_i)$. Because H_1 is a secure cryptographic hash function, H_1 is collision-resistant. So, due to the collision resistance of H_1 there must be the relation $B_i=a_iY_i$.

Theorem 4.3 If the equation $e(S, Y_n+hP)=g$ holds, where $h=H_2(U_n, D_1, D_2, ..., D_{n-1})$, then the message ($U_n$, D_1, D_2, ..., D_{n-1}) sent from U_n can be verified by each U_j (1≤j≤n−1).

Proof. Since $S=P/(X_n+h)$, where $h=H_2(U_n, D_1, D_2, ..., D_{n-1})$, we have

$$e(S, Y_n+hP)= e(P/(X_n+h), (X_n+h)P) =e(P, P) =g.$$

In fact, S can be seemed as the signature proposed by Zhang et al [21]. In [21], this signature S had been proved to be secure against forgery. Then, each U_j (1≤j≤n−1) can verify the message (U_n, D_1, D_2, ..., D_{n-1}) sent from the powerful node U_n.

4.2. Security Analysis

In this section, we prove that our protocol is secure. An attacker cannot obtain the established group key by eavesdropping on the messages transmitted over the public channel. To prove the security of our protocol, we adopt the Square-Exponent Problem (SEP) [21, 24, 25].

Let θ be a generator which generates the group G, i.e., for any $y \in G$, there exists $x \in Z_{|G|}$ such that $y=\theta^x$. The Square-Exponent Problem (SEP) is defined as follow.

Definition 3 Square-Exponent Problem (SEP) [21, 24, 25]: Given Θ, $\Theta^b \in G$, the Square-Exponent Problem (SEP) is to compute θ^{b^2}, where $b \in Z_{|G|}$.

Assumption: We always assume that SEP is hard computational problem such that there is no polynomial time algorithm to solve it with non-negligible probability.

Theorem 4.4 Under the hardness assumption of SEP, the proposed group key agreement protocol is secure. An attacker cannot obtain the established group key by eavesdropping on messages transmitted over the public channel.

Proof. To prove the security of our protocol, we use the contradiction proof technique under the hardness assumption of SEP. That is, we prove that if there exists an efficient probabilistic polynomial algorithm F which can obtain the established group key, we can construct another efficient algorithm F' to compute θ^{b^2} from Θ and Θ^b, which is conflict to the hardness assumption of SEP.

Assume that there exists an efficient probabilistic polynomial algorithm F that can compute the common group key K with a probability ε from the messages transmitted over the public channel. Based on the algorithm F, we show that another polynomial algorithm F' can be constructed to solve an instance of SEP with a probability ε. Assume that F' knows the long-term private key X_n. That is, F' can be run by the powerful node U_n. Now, F' setups the algorithm as follows. F' randomly selects B_1 and D_1 in G_1, where $B_1=aP$, $D_1=abP$, and a and b are unknown numbers. F' computes $A_1=X_nB_1$ and sets $\Theta := e(B_1, P)$. Let $G=<\Theta>$ be the cyclic group generated by the generator Θ. F' computes Θ^b by computing

$$\Theta^b=e(D_1, P). \tag{1}$$

Without knowing b, the goal of F' is to solve an instance of SEP: computing θ^{b^2} from Θ and Θ^b. To do so, algorithm F' randomly selects $w_1, w_2, ..., w_{n-2} \in Z_q^*$ and computes

$$\begin{aligned}
&B_2=w_1B_1,\ D_2=w_1D_1,\ A_2=w_1A_1,\\
&B_3=w_2B_1,\ D_3=w_2D_1,\ A_3=w_2A_1,\\
&\vdots\\
&B_{n-1}=w_{n-2}B_1,\ D_{n-1}=w_{n-2}D_1,\ A_{n-1}=w_{n-2}A_1.
\end{aligned}$$

Then, the algorithm F' has constructed all (A_i, D_i), for $1 \le i \le n-1$. It should be noted that in step 2 of our protocol in Section 3, $D_i=a_nB_i$. Here, according to the construction of algorithm F', there exists the relation $D_i=bB_i$. Then, in algorithm F', the number b can be seemed as the number a_n in Section 3. Then, F' calls F with all (A_i, D_i) for $1 \le i \le n-1$ so as to the attacker F computes the common group key K. If the algorithm F can compute the common group key

$$K = e(a_nP, \sum_{i=1}^{n\text{-}1} D_i) = e(bP, \sum_{i=1}^{n\text{-}1} D_i), \tag{2}$$

F' can compute $\theta^{b^2} = K^{\frac{1}{1+w_1+\cdots w_{n-2}}}$. In fact, according to algorithm F', the bilinear property of the bilinear pairing and Eq.(1-2), we have

$$\begin{aligned}
&K^{\frac{1}{1+w_1+\cdots w_{n-2}}}\\
&= e(bP, \sum_{i=1}^{n-1} D_i)^{\frac{1}{1+w_1+\cdots w_{n-2}}}\\
&= e(bP, D_1 + D_2 + \cdots + D_{n-1})^{\frac{1}{1+w_1+\cdots w_{n-2}}}\\
&= e(bP, (1+w_1+w_2+\cdots w_{n-2})D_1)^{\frac{1}{1+w_1+\cdots w_{n-2}}}\\
&= e(D_1, P)^b
\end{aligned}$$

$$= \theta^{b^2}$$

That is, if there exists an efficient probabilistic polynomial algorithm F which can compute the group key with a probability ε, there exists another polynomial algorithm F' which can compute θ^{b^2} from Θ and Θ^b with the same probability ε too. This is conflict to the hardness assumption of SEP. Therefore, under the hardness assumption of SEP, the proposed group key agreement protocol is secure.

Theorem 4.5 Suppose $Encrypt_{k_i}(\ )$ is a secure standard symmetric encryption algorithm. Then, under the hardness assumption of BCDHP and the security assumption of the symmetric encryption algorithm $Encrypt_{k_i}(\ )$, the proposed protocol can resist the impersonation attack. An attacker cannot forge the valid authenticator (U_i, A_i, c_i) or the signature S.

Proof. For a valid authenticator (U_i, A_i, c_i), the equations $A_i=a_iX_iY_n=a_iX_nY_i$, $k_i=H_1(a_iY_i)$ and $c_i=Encrypt_{k_i}(a_i)$ hold. The unforgery of the authenticator (U_i, A_i, c_i) can be proved by using three steps, in which the contradiction proof technique is used.

Assume the authenticator (U_i, A_i, c_i) can be forged. That is, we assume that there exists a polynomial attacker F who successfully forged a valid authenticator (U_i, A_i, c_i), where $A_i=a_iX_iY_n=a_iX_nY_i$, $k_i=H_1(a_iY_i)$ and $c_i=Encrypt_{k_i}(a_i)$.

In the first step as follow, under the hardness assumption of BCDHP, we prove that F cannot know the random number a_i. In the second step as follow, we prove that it is impossible for F to known the symmetric key k_i used in the symmetric algorithms $Encrypt_{k_i}(\ )$ and $Decrypt_{k_i}(\ )$. In the third step, we prove that F can compute a correct ciphertext c_i without knowing symmetric k_i or the corresponding plaintext, which is conflict to the security assumption of the symmetric encryption algorithm $Encrypt_{k_i}(\ )$.

Step 1. In this step, we prove hat F cannot know the random number a_i. Note that X_n and X_i are long-term private keys of U_n and U_i, respectively, while $Y_n=X_nP$ and $Y_i=X_iP$ are corresponding public keys. F knows neither the long-term private key X_n nor the long-term private key X_i, but he knows the public keys $Y_n=X_nP$ and $Y_i=X_iP$. Note the authenticator (U_i, A_i, c_i) is forged by F. In fact, if F knows the random number a_i, he can compute $a_i^{-1}A_i=X_nX_iP$ from Y_n and Y_i. That is, if F knows the a_i, F can compute X_nX_iP from $X_nP(=Y_n)$ and $X_iP(=Y_i)$, which is conflict to the hardness assumption of BCDHP. Then, under the hardness assumption of BCDHP, it is impossible for the attacker F to know a_i.

Step 2. Note the authenticator (U_i, A_i, c_i) is a valid forgery forged by F. Then, $c_i=Encrypt_{k_i}(a_i)$, where $k_i=H_1(a_iY_i)$. In the symmetric algorithms $Encrypt_{k_i}(\ )$ and $Decrypt_{k_i}(\ )$, the symmetric key k_i has to be used. Note $k_i=H_1(a_iY_i)=H_1(a_iX_iP)$. At the same time, in step 1, we have proved that F knows neither a_i nor the long-term private key X_i. To obtain the symmetric key k_i, one has to compute a_iX_iP from a_iP and X_iP, which is an instance of BCDHP. Therefore, under the hardness assumption of BCDHP, it is impossible for the attacker F to know k_i.

Step 3. In step 1 and step 2, we have proved that the attacker F knows neither a_i nor the symmetric key k_i, but the authenticator (U_i, A_i, c_i) is a valid forgery forged by F, where c_i satisfies $c_i=Encrypt_{k_i}(a_i)$. That is, without knowing the symmetric key k_i or the plaintext a_i, F can compute a correct ciphertext $c_i=Encrypt_{k_i}(a_i)$, which is conflict to the security assumption of symmetric encryption algorithm $Encrypt_{k_i}(\ )$.

Therefore, from the proof of steps above, we have: under the hardness assumption of BCDHP and the security assumption of the symmetric encryption algorithm $Encrypt_{k_i}(\ )$, F cannot forge the valid authenticator (U_i, A_i, c_i). On the other hand, the signature S was proved to be secure against forgery in [21]. Therefore, an attacker cannot forge the valid authenticator (U_i, A_i, c_i) or the signature S.

Theorem 4.6 A disclosed session common group key does not affect the security of the proposed protocol.

Proof. In general, in a group key agreement protocol, a compromised session common group key must not affect the security of other session common group key. That is, the requirement of forward or backward secrecy should be satisfied. In our protocol, the session common group key can be derived from

$$K = e(a_n P, \sum_{i=1}^{n-1} D_i) = e(a_n^2 P, \sum_{i=1}^{n-1} a_i Y_i),$$

where the random numbers $a_i (1 \leq i \leq n)$ are chosen independently by the participants U_i $(1 \leq i \leq n)$ respectively from each session. Then, all the session common group keys are independent each other from different sessions. Thereby, an adversary cannot derive another session common group key K' from a disclosed session common group key K.

4.3. Performance Comparison

In this Section, we show the performance comparison of Nam's protocol [3], Tseng's protocol [4], Lee's protocol [5] and our authenticated protocol. For ease of comparison, we use the notations defined as follows:

$|m|$: the bit length of a message m

T_{exp}: the time for modular exponentiation

T_{inv}: the time for modular inverse

T_{mul}: the time for modular multiplication

T_{bp}: the time for bilinear pairing

T_{smul}: the time for scalar multiplication

T_{sadd}: the time for scalar addition

T_{H}: the time for hashing operation

T_{dec}: the time for decrypting operation using the standard symmetric algorithm

In the Table 3 as follow, the contributory property, the nonauthenticated or the authenticated property, the number of rounds, the size of the transmitted messages, the computational complexity required for each client and the powerful node, and the underlying problems of different protocols are compared.

Table 3. Efficiency Comparisons

	Nam	Tseng	Lee	Our
CGKA	No	Yes	Yes	Yes
AP	No	No	No	Yes
NR	2	2	2	2
MSC	$\|U\|+\|p\|$	$\|U\|+\|p\|$	$\|U\|+2\|q\|$	$\|U\|+2\|q\|$
MSPN	$\|U\|+n\|p\|$	$\|U\|+(n-1)\|p\|$	$\|U\|+n\|q\|$	$\|U\|+n\|q\|$
CCC	$T_{\mathrm{exp}}+T_{\mathrm{mul}}$	$T_{\mathrm{exp}}+(n-1)T_{\mathrm{mul}}$	$3T_{\mathrm{bp}}+T_{\mathrm{smul}}+(n-2)T_{\mathrm{sadd}}+T_{\mathrm{H}}$	$2T_{\mathrm{bp}}+2T_{\mathrm{smul}}+n-1)T_{\mathrm{sadd}}+T_{\mathrm{H}}$
CCPN	$(n+1)T_{\mathrm{exp}}+nT_{\mathrm{inv}}+(2n-2)T_{\mathrm{mul}}$	$nT_{\mathrm{exp}}+(n-1)T_{\mathrm{mul}}$	$(2n-1)T_{\mathrm{bp}}+(n+1)T_{\mathrm{smul}}+(n-2)T_{\mathrm{sadd}}+T_{\mathrm{H}}$	$T_{\mathrm{bp}}+(3n-1)T_{\mathrm{smul}}+(n-2)T_{\mathrm{sadd}}+nT_{\mathrm{H}}+T_{\mathrm{inv}}+(n-1)T_{\mathrm{dec}}$
BCP	DLP	DLP	ECDLP/BCDHP	ECDLP/BCDHP/SEP

CGKA: contributory group key agreement; *NR*: number of rounds; *MSC*: message size sent by each client; *MSPN*: message size sent by the powerful node (via broadcast); *CCC*: computation costs required by each client (online); *CCPN*: computation costs required by the powerful node; *BCP*: based on cryptographic problem; *AP*: authenticated protocol

According to Table 3, it is found the protocols of Nam, Tseng and Lee have the same security weakness. That is, they are all noauthenticated protocols. In Theorem 4.5, the unforgery of the transmitted authenticator (U_i, A_i, c_i) and the signature S are proved. So, our protocol is an authenticated one. At the same time, our protocol has the forward and backward security property, which has been proved in Theorem 4.6.

On the other hand, it is found both Lee and our protocols are pairing-based protocol. However, in Lee's protocol, to authenticate the transmitted messages and construct the group key, U_n has to compute at least $2n$-1 pairing operations. That is, the more clients, the more pairing operations. In our protocol, U_n only needs to compute one pairing operations. It is found that the message size of every protocol listed in Table 3 grows linearly with the number of participants. This is because the resulting common group key should be composed of contributions by all participants in a contributory group key agreement protocol. So, it is necessary for the powerful node U_n to broadcast the contributions of all the participants to generate the common group key. This will cause that the message size broadcasted by the powerful node U_n grows linearly with the number of participants.

Let us consider the computational cost for each client. In our protocol, each client can precompute (X_i^{-1}, a_i^{-1}, $(a_iX_i)^{-1}$, a_i, A_i, c_i) off-line and store them on its memory card. Then, in our protocol, for each client, only two pairing operations online are required. But, in Lee's protocol, three pairing operations have to be computed online by each client.

Table 3 shows that our protocol is an authenticated protocol, while the others not. On the other hand, the security of our authenticated key agreement protocol is based on ECDLP, BCDHP and SEP.

5. Conclusions

The design of a secure group key agreement protocol for mobile wireless networks is an important issue to provide secure services among mobile devices. Although many group key agreement protocols have been proposed, most of them have one shortcoming or another. In this paper, we demonstrate that Lee et al.'s key agreement protocol in the mobile environment is a nonauthenticated protocol. Then, we propose a new one based on bilinear pairings. Our protocol overcomes the security drawback of Lee et al., and it is more efficient than the ones of the same kinds. The new protocol can be proved to be secure under the hardness assumptions of ECDLP, BCDHP and SEP. What is more, our protocol is a contributory group key agreement protocol. Our protocol can be used to guarantee the secure group communication for legitimate group members in an imbalanced mobile environment.

Acknowledgements

This work is supported by the Natural Science Foundation of China (Grant No. 61272525), Science Research, the Foundation for Doctors of Zhengzhou University of Light Industry (NO. 20080014). and the Fundamental and Advanced Technology Research Project of Henan province (Principal Investigator: Xiangjun Xin).

References

[1] N. Asokan and P. Ginzboorg, "Key agreement in ad hoc networks", Computer Communications, vol. 23, no. 17, **(2000)**, pp.1627–1637.

[2] C. Boyd and J. M. G. Nieto, "Round-optimal contributory conference key agreement", Proceedings of public-key cryptography, **(2003)** January 6-8; Miami, FL, USA.

[3] J. Nam, J. Lee, S. Kim and D. Won, "DDH-based group key agreement in a mobile environment", Journal of Systems and Software, vol.78, no. 1, **(2005)**, pp. 73–83.

[4] Y. M. Tseng, "A resource-constrained group key agreement protocol for imbalanced wireless networks", Computer Security, vol. 26, no. 4, **(2007)**, pp. 331–337.

[5] C. C. Lee, T. H. Lin and C. S. Tsai, "A new authenticated group key agreement in a mobile

environment", Ann. Telecommun., vol. 64, **(2009)**, pp. 735–744.

[6] M. Steiner, G. Tsudik and M. Waidner, "Diffie–Hellman key distribution extended to group communication", Proceedings of the 13th ACM Conference on Computer and Communications Security, **(2006)** October 30- November 3; Alexandria, Virginia, USA.

[7] M. Steiner, G. Tsudik and M. Waidner, "Cliques: a new approach to group key agreement", Proceedings of the 18th International Conference on Distributed Computing Systems, **(1998)** May 26 – 29; Amsterdam, The Netherlands.

[8] M. S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications", IEEE Transaction on Vehicular Technology, vol. 48, no. 5, **(1999)**, pp. 1469–1474.

[9] S. L. Ng, "Comments on "Dynamic participation in a secure conference scheme for mobile communications", IEEE Transaction on Vehicular Technology, vol. 50, no. 1, **(2001)**, pp. 334 - 335.

[10] E. Bresson, O. Chevassut, A. Essiari, *et al.*, "Mutual authentication and group key agreement for low-power mobile devices", Computer Communications, vol. 27, no. 17, **(2004)**, pp. 1730–1737.

[11] J. Nam, S. Kim and D. Won, "Attacks on bresson-chevassutessiari-pointcheval's group key agreement scheme for low power mobile devices", Cryptology ePrint archive, report, vol. 251, **(2004)**

[12] Y. Kim, A. Perrig and G. Tsudik, "Tree based group key agreement", ACM Transactions on Information and System Security", vol. 7, no. 1, **(2004)**, pp. 60–96.

[13] R. Dutta and R. Barua, "Provably secure constant round contributory group key agreement in dynamic setting", IEEE Transactions on Information Theory, vol. 54, no. 5, **(2008)**, pp. 2007–2025.

[14] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, vol. 31, no. 4, **(1985)**, pp. 469–472.

[15] A. Shamir, "How to share a secret", Communications of the ACM, vol. 22, no. 11, **(1979)**, pp. 612–613.

[16] C.H Tan and G. Yang, "Comments on "Provably Secure Constant Round Contributory Group Key Agreement in Dynamic Setting", IEEE Transactions on Information Theory, vol. 56, no. 11, **(2010)**, pp. 5887 - 5888.

[17] K. W. Hu, J. F. Xue, C. Z. Hu, R. Ma and Z. Q Li, "An Improved ID-Based Group Key Agreement Protocol", Tsinghua Science and Technology, vol. 19, no. 5, **(2014)**, pp. 421-428.

[18] J. K. Teng, C. K. Wu and C. M.Tang, "An ID-based authenticated dynamic group key agreement with optimal round", Science China, Information Sciences, vol. 35, no. 11, **(2012)**, pp. 2542-2554.

[19] E. Konstantinou, "An efficient constant round id-based group key agreement protocol for ad hoc networks", The 7th International Conference on Network and System Security, **(2013)** June 3-4; Madrid, Spain.

[20] S. Mitsunari, R. Sakai and M. Kasahara, "A new traitor tracing, IEICE Transactions", vol. E85-A, no. 2, **(2002)**, pp. 481-484.

[21] F. Zhang, R. Safavi-Naini and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications", Public Key Cryptography, (**2004)** March1-4; Singapore.

[22] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", IEEE Transactions on Information Theory, vol. 39, no. 5, **(1993)**, pp. 1639–1646.

[23] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups", ACM Symposium on Applied Computing, **(2003)** March9 -12; Melbourne, Florida, USA.

[24] A. R. Sadeghi and M. Steiner, "Assumptions related to discrete logarithms: why subtleties make a real difference", Eurocrypt, **(2001)** May 6 – 10; Innsbruck (Tyrol), Austria.

[25] U. M. Maurer and S. Wolf, "Diffie-Hellman oracles", Advances in Cryptology-CRYPTO'96, (**1996)** August 18–22; Santa Barbara, California, USA.

Authors

Xiangjun Xin, He received his Ph.D. degree in Cryptography from Xidian University in 2007. He is now an associate professor in the School of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou, China. His recent research interests include cryptography and network security.

Chaoyang Li, He is now a postgraduate in the School of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou, China. His recent research interests include cryptography and network security.

Dongsheng Chen, He received his B.S. degree in mathematics from Henan University in 1982. He is now a professor in the School of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou, China. His recent research interests include linear algebra and its applications.

Fagen Li, He received his Ph.D. degree in Cryptography from Xidian University, Xi'an, P.R. China in 2007. He is now an associate professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His recent research interests include cryptography and network security.