Efficient group Diffie–Hellman key agreement protocols

Lein Harn\textsuperscript{a}, Changlu Lin\textsuperscript{b,*}

\textsuperscript{a}Department of Computer Science Electrical Engineering, University of Missouri–Kansas City, MO 64110, USA
\textsuperscript{b}Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fujian 350007, PR China

\begin{abstract}
In a group Diffie–Hellman (GDH) key agreement protocol, all group members collaboratively establish a group key. Most GDH key agreement protocols took natural generalization of the original Diffie–Hellman (DH) key agreement protocol to arrange all group members in a logic ring or a binary tree and to exchange DH public keys. The computational cost and the communication rounds are the two most important factors that affect the efficiency of a GDH protocol when there are a large number of group members. In this paper, we propose GDH key agreement protocols based on the secret sharing scheme. In addition, we use a one-way key confirmation and digital certificates of DH public keys to provide authentication of group keys. In the proposed authenticated GDH key agreement protocol, each group member requires to broadcast three-round messages, $n$ modular exponentiations, $n$ polynomial interpolations and $n$ one-way functions. Our proposed solution is efficient, robust and secure.
\end{abstract}

\section{Introduction}

\subsection{Background}

Network applications are no longer just one-to-one communication; but involve multiple users (>2). Group communication implies a many-to-many communication and it goes beyond both one-to-one communication (i.e., unicast) and one-to-many communication (i.e., multicast). In a secure group communication, after all users being authenticated, a one-time session key needs to be shared among all group members. Most well-known group key establishment protocols can be classified into two categories:

- Centralized group key establishment protocols: a group key generation center (KGC) is engaged in managing the entire group.
- Distributed group key establishment protocols: there is no explicit group KGC, and each group member can contribute to the group key generation and distribution.

The class of centralized group key establishment protocols is the most widely used protocols due to its efficiency in implementation.

In a secure communication involving $n$ members ($n \geq 2$), a group key needs to be shared among all group members and uses it to encrypt and authenticate messages. According to [1], there are two types of key establishment protocols: key transfer protocols and key agreement protocols. Key transfer protocols rely on a mutually trusted key generation center (KGC) to
select session keys and then transports session keys to all communication entities secretly. In key agreement protocols, all communication entities collaboratively determine session keys. The most commonly used key agreement protocol is the Diffie–Hellman (DH) key agreement protocol [2]. In DH protocol, the session key is determined by exchanging DH public keys of two communication entities. Since the public key itself does not provide any authentication, a digital signature of the public key can be used to provide authentication. However, DH protocol can provide session key only for two entities; not for a group more than two members.

1.2. Related works

Computing a group DH key among a set of \( n \) group members is a special case of secure multiparty computation in which a group of \( n \) members who each possesses a private input \( k_i \), computes a function \( f(k_1, k_2, \ldots, k_n) \) securely [3]. For example, Tzeng and Tzeng [4,5] proposed a round-efficient conference key with \( f(k_1, k_2, \ldots, k_n) = g^{k_1 + k_2 + \ldots + k_n} \mod p \). Burmester and Desmedt [6] proposed a round-efficient (two-round) protocol (Protocol 3) with \( f(k_1, k_2, \ldots, k_n) = g^{k_1 + k_2 + \ldots + k_n} \mod p \). Most group DH protocols took natural generalization of the original DH key agreement protocol. For example, Ingemarsson et al. [7], Steer et al. [8], Burmester and Desmedt [6], and Steiner et al. [9] followed this approach to arrange group members in a logic ring and to exchange DH public keys; Lee et al. [10] and Kim et al. [11,12] arranged group members in a binary tree. In 1996, Steiner et al. [9] proposed a natural extension of DH protocol, named the group DH (GDH) key exchange. Later, in 2001, protocol proposed by Steiner et al. has been enhanced with authentication services and is proven to be secure [13]. In 2006, Bohli [14] developed a framework for robust group key agreement protocols that provides security against malicious insiders and active adversaries in an unauthenticated point-to-point network. Then, in 2007, Bresson et al. [15] constructed a generic authenticated GDH Key exchange protocol and the protocol is provably secure. Also, in 2007, Katz and Yung [16] proposed the first constant-round and fully scalable GDH protocol which is provably secure in the standard model (i.e., without assuming the existence of “random oracles”). In 2009, Brecher et al. [17] extended the tree-DH technique of GDH protocol with robustness, i.e., with resistance to faults resulting from possible system crashes, network failures, and misbehavior of the members. In 2011, Jarecki et al. [18] proposed a robust group key agreement protocol which can tolerate up to \( t \) nodes failure. One common feature in these protocols is that secure digital signatures are generated to provide authentication of DH public keys. Since generation and verification of digital signatures take times, the computational cost of each group member is the main concern in implementing these protocols especially when there are a large number of group members.

After Joux’s proposal [19] to use pairings to enable a one-round tripartite key exchange (KE) in 2000, several extensions of authenticated group key exchange protocols [20–24] were published. Unfortunately, most of pairing-based group KE protocols are not very efficient, i.e., the number of rounds grows with the group size. In 2004, Choi et al. [22] proposed a pairing-based group KE protocol which requires a constant number of rounds, broadcast of \( n \) messages, and every member needs to compute two pairings and \( 4n \) modular exponentiations. Barua et al. [21] proposed a pairing-based group KE using a tree. Du et al. [23] proposed an authenticated ID-based group key exchange scheme which attains a constant number of rounds. In 2008, Desmedt and Lange [25] proposed a constant round pairing-based authenticated group KE with lower computational complexity per member than other protocols. Wu et al. [26] and Zhang et al. [27] presented the definition of asymmetric group key agreement and this model is focused on implementing secure channels for group–oriented communications. Recently, Gu et al. [28] proposed an integrated group key agreement protocol to reduce the rekeying time in a hierarchical access control. Konstantinou [29] proposed an ID-based group key agreement protocol with efficient constant round in ad hoc networks.

There are group key transfer protocols using the secret sharing. In 1989, Laih et al. [30] proposed the first group key transfer protocol using a \( (t, n) \) secret sharing scheme. In their scheme, each member needs to register at a conference chairperson initially and shares a secret with the chairperson. The conference chairperson is responsible to select a random conference key as the secret and uses the secret sharing scheme to broadcast shares of the secret to members. Later, there are papers [31–33] following the same approach to distribute group messages to multiple members secretly. Cao et al. [34] proposed a constant-round group key exchange protocols using the secret sharing with the universally composable security. Harn and Lin [35] proposed an authenticated group key transfer protocol using the secret sharing scheme. In a recent paper, Olimid [36] discussed the security of this kind of group key transfer protocol.

1.3. Our contributions

Clearly, the computational cost and the communication rounds are two important factors that affect the efficiency of a GDH protocol, especially when the number of group members grows large. In this paper, we propose GDH protocols using the secret sharing scheme. So far as we know, there has no GDH protocol that incorporates both DH scheme and the secret sharing scheme. Our proposed GDH protocols are secure, robust and efficient. The proposed basic GDH protocol is “proven” to be secure provided the DH problem is intractable. In our proposed authenticated GDH protocol, each member employs a one-way hash function to generate the DH key confirmation and uses it to provide the authentication of group keys. The authenticated GDH protocol can provide key secrecy, perfect forward secrecy and key independence of group keys. In addition, the authenticated GDH protocol can resist unknown key-share attack and key compromise impersonation attack. We list the contributions of this paper below.
Our protocols are efficient in terms of computational cost and the communication rounds.

The security of the basic GDH protocol is proven to be secure provided the DH problem is intractable.

The authenticated GDH protocol is robust to accommodate dynamic change of group memberships.

The authenticated GDH protocol can provide key secrecy, perfect forward secrecy and key independence and can resist unknown key-share attack and key compromise impersonation attack.

The organization of this paper is as follows. In Section 2 we present the model for GDH systems including security, attacks and adversaries. In Section 3 we present a basic GDH protocol which is based on a secure broadcast encryption scheme using the secret sharing. In Section 4 we present an authenticated GDH protocol and in Section 5 we discuss the security and in Section 6 we compare the performance of proposed protocol with other protocols. We conclude in Section 7.

2. Model of group Diffie–Hellman key agreement protocol

In this section, we describe the model of our GDH protocols including the system requirements, the adversary, security goals and possible attacks of group key. Throughout this paper, we use the symbol, $n$, to denote the number of group members.

2.1. Protocol description

In our proposed GDH protocols, there has no key generation server. The group key is determined by all group members collaboratively. Each group member contributes a one-time DH secret $k_i$. For example, a group consisting of four members with individual secrets, $k_1, k_2, k_3, k_4$, respectively, our GDH protocols enable each group member to compute the group key $g^{2k_1k_2+k_1k_3+k_1k_4+k_2k_3+k_2k_4+k_3k_4} \mod p$ secretly. Each group key is used for only one communication session and our GDH protocol can accommodate dynamical change of group memberships. In other words, we do not need to consider the actions to add/remove members in secret communications. When a new group communication session is set up, a new group key will be generated.

In our authenticated GDH protocol, each member needs a pair of long-term DH private and public keys and the long-term DH public key has been digitally signed by a trusted Certificate Authority (CA). The digital certificate of public keys will be used to authenticate group keys. In our authenticated GDH protocol, each member uses a one-way function to generate a key confirmation of the group key. Through this key confirmation, it provides group key authentication. Since the computation of a one-way function is faster than generation and verification of a digital signature, our proposed GDH protocol is more efficient than authenticated GDH protocols using the digital signatures.

The communication rounds is another important factor affecting the efficiency of a GDH protocol. The basic GDH protocol has two rounds and the authenticated GDH protocol has three rounds.

2.2. Type of adversaries

We consider two types of adversaries: insider and outsider. The inside attacker is a legitimate member who knows the group key; but inside attacker may try to recover other members' secrets (long-term private keys). After knowing each long-term private key, the inside attacker is able to reveal other group keys that he is not authorized to know or is able to impersonate other members in a secure group communication. The outside attacker may try to recover the group key that he is unauthorized to know. This attack is related to the secrecy of group keys. In our authenticated GDH protocol, each member contributes a one-time DH public key in the first round; but only legitimate members can generate the one-way key confirmations. The one-time DH public and private keys are used for only one group key agreement. The outside attacker may also try to impersonate an legitimate member in the group communication. In security analysis, we will show that none of these attacks can work properly against our authenticated GDH protocol.

2.3. Security of group keys

We assume that a sequence of group keys is denoted as $K = \{K_1, K_2, \ldots, K_n\}$. We consider the following security goals of group keys.

(a) Key secrecy: It is computationally infeasible for the adversary to discover any group key $K_i$.

(b) Perfect forward secrecy: It ensures that any key will not be compromised if one of the long-term private keys is compromised in the future.

(c) Key independence: The adversary who knows a subset of group keys, $K' \subset K$, cannot discover any other group key, $K_i \in K - K'$.

2.4. Attacks to the proposed protocol

We consider the following attacks of our authenticated GDH protocol.
(a) **Unknown key-share attack:** An entity $A$ ends up believing that she/he shares a key with $B$, and although this is in fact the case, while $B$ mistakenly believes that the key is instead shared with an entity $E \neq B$.

(b) **Key compromise impersonation attack:** Suppose $A$'s long-term private key is compromised. The adversary who knows $A$'s long-term private key can impersonate $A$, since this value identifies $A$. However, this attack enables the adversary to impersonate other entities to $A$.

### 3. Basic group Diffie–Hellman key agreement protocol

In this section, we propose a basic GDH protocol to allow $n$ group members collaboratively determine a group key secretly. This basic GDH protocol only provides secrecy of the group key to all group members. In the basic GDH protocol, each member selects a one-time DH secret and broadcasts the one-time DH public key in the first round. After receiving each DH public key from one of the other group members, a shared DH secret between two members is established. Thus, any member can establish $n - 1$ DH secrets with other members. Then, each member uses an unconditionally secure encryption scheme to distribute these $n - 1$ shared DH secrets to other members. The unconditionally secure encryption scheme is based on the secret sharing scheme.

#### 3.1. Secure broadcasting (SB) scheme with unconditional security

Assume that member $U_1$ wants to transmit the message, $m$, secretly to $n - 1$ members, $\{U_2, U_3, \ldots, U_n\}$, in a broadcast channel. The system has following parameters:

- $s_i$: a shared secret between member $U_1$ and member $U_i$, where $i \neq 1$. We assume that these shared secrets have been established initially.
- $p$: a large public prime number.
- $z_i$: a public identity of each member $U_i$ with $z_i \in [1, n - 1]$.

The detail descriptions of our proposed scheme is given in Fig. 1.

**Remark 1.** The SB scheme has one potential problem related to its efficiency. For any group member $U_i$, where $i \in [2, n]$, is able to recover the polynomial, $f_1(x)$, and obtain the shared secret, $s_j = f_1(z_j)$, between $U_1$ and any other group member, $U_j$, where $j \neq 1$. Thus, each shared secret can only be used for one communication session. If $U_1$ uses the same shared secret $s_j$ for sending multiple messages to member $U_j$, it may compromise the secrecy of messages since the shared secret, $s_j$, is no longer a secret for other group members after one communication session.

#### 3.2. Basic group Diffie–Hellman key agreement protocol

Assume that $n$ members, $\{U_1, U_2, \ldots, U_n\}$, want to set up a group key collaboratively in a broadcast channel. The goal of the basic GDH protocol is to provide the secrecy of the group key to all group members. The system has following parameters:

- $p$: a large prime number that is $2q + 1$, where $q$ is also a large prime.
- $g$: a generator for the subgroup $G_q$.

**Fig. 1.** Secure broadcasting scheme.
Each member \( U_i \) has two parameters:

- A one-time (short-term) DH private key \( k_i \): a number in \( \mathbb{Z}_q \) – \{1\}.
- A one-time (short-term) DH public key \( r_i = g^{k_i} \pmod{p} \). Since \( q \) is a prime number, \( r_i \) is a generator for \( G_q \).

The detail descriptions of our proposed basic protocol is given in Fig. 2.

**Note 1.** (a) In Round 1, each member selects a one-time DH secret and broadcasts its one-time DH public key. (b) In Round 2, each member constructs the combination of \( n-1 \) shared DH secrets with other members as the secret message and the shared DH secret with every other member as the shared secret in the SB scheme to distribute the secret message to other members secretly.

**Remark 2.** In basic GDH protocol, there is a shared secret between each pair of members. For example, between members \( U_i \) and \( U_j \), the shared DH secret is \( s_{ij} = g^{k_i k_j} \pmod{p} \). This shared secret is a one-time secret. Therefore, although in Round 2, each group member can recover the shared DH secret of other group members, this result will not cause any security problem if the shared DH secret is only used for one communication session.

### 4. Authenticated group Diffie–Hellman key agreement protocol

The goal of this authenticated GDH protocol is to provide the secrecy and authenticity of the group key to all group members.

In addition to having the parameters used in the basic GDH protocol, each member in the authenticated GDH protocol needs two additional parameters:

- A long-term DH private key \( x_i \): a number in \( \mathbb{Z}_q \) – \{1\}.
- A long-term DH public key \( y_i = g^{x_i} \pmod{p} \). Since \( q \) is a prime number, \( y_i \) is a generator for \( G_q \).
Each member’s long-term DH public key, $y_i$, needs to be digitally signed by a trusted CA. The digital certificate can provide authentication of the public key. We assume that each member has obtained other group members’ digital certificates and has authenticated their public keys before staring the GDH protocol. The detail descriptions of our proposed protocol is given in Fig. 3.

**Note 2.** (a) In this authenticated GDH protocol, each shared DH secret between every pair of members involves one-time and long-term private/public keys of members. The purpose of using a one-time key is to provide key independence and perfect forward secrecy (we will explain this in next section). On the other hand, the purpose of using a lone-term key is to provide key authentication. (b) In Round 3, a key confirmation is generated by each member. Only after all key confirmations being verified successfully, the group key is established for a secure group communication. In case any failure of key confirmations, the authenticated GDH protocol needs to be restarted.

**Remark 3.** For this authenticated GDH protocol, each group member requires to broadcast three-round messages, $n$ modular exponentiations, $n$ polynomial interpolations and $n$ one-way functions.

### 5. Security analysis

In this section, we analyze the security of the SB scheme and the basic GDH protocol first. Then, we examine security properties of group keys. Finally, we analyze possible attacks of the authenticated GDH protocol.

#### Theorem 1. The SB scheme is unconditionally secure.

**Proof.** $U_i$ constructs the polynomial, $f_i(x)$, having $n - 1$ degree and broadcasts $f_i(i)$, for $i = 1, 2, \ldots, n - 1$. For each intended group member, $U_j$, where $i \in [2, n]$, knows one shared secret $s_i$. The point, $(z_i, s_i)$, is also on the polynomial $f_i(i)$. Thus, each

<table>
<thead>
<tr>
<th>Round</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1     | Each member $U_i$ broadcasts $r_i$.  
|       | - Step 1. After receiving all $r_j$, for $j = 1, 2, \ldots, n$ and $j \neq i$, $U_i$ computes the shared secrets, $s_{ij} = (y_j r_j)^{x_i + k_i} \pmod{p}$, for $j = 1, 2, \ldots, n$ and $j \neq i$.  
|       | - Step 2. Then, $U_i$ constructs the secret message, $K_i = \prod_{j=1, j \neq i}^{n} r_j^{k_i} \pmod{p}$, and constructs a polynomial $f_i(x)$ having $n - 1$ degree passing through $n$ points, $\{(0, K_i), (r_1, s_{1i}), \ldots, (r_j, s_{ji}), \ldots, (r_n, s_{ni}) \mid j \neq i\}$, following the SB scheme.  
|       | - Step 3. $U_i$ computes $f_i(j)$, for $j = 1, 2, \ldots, n - 1$.  
| 2     | Each member $U_i$ does follows.  
|       | - Step 1. Each member $U_i$ broadcasts $f_i(j)$, for $j = 1, 2, \ldots, n - 1$, publicly.  
|       | - Step 2. After receiving all $f_k(j)$, for $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$ and $k \neq i$, each member $U_j$ computes the shared secrets, $s_{ij} = (y_j r_j)^{x_i+k_i} \pmod{p}$, for $j = 1, 2, \ldots, n$, and uses them to recovers $K_j$, for $j = 1, 2, \ldots, n$ and $j \neq i$, following SB scheme.  
| 3     | Each member $U_i$ does follows.  
|       | - Step 1. Each member $U_i$ computes and broadcasts the key confirmation as $c_i = h(K_i', U_i, r_i)$, where $K_i' = \prod_{j=1}^{n} K_j \pmod{p}$.  
|       | - Step 2. After receiving all key confirmations from members, each member $U_i$ computes $h(K_i', U_j, r_j) = c_j$, for $j = 1, 2, \ldots, n$ and $j \neq i$. If $c_j' = c_j$, for $j = 1, 2, \ldots, n$ and $j \neq i$, the group key is $K = K'$; otherwise, restarts the protocol.  

Fig. 3. Authenticated group Diffie–Hellman key agreement protocol.
group member has enough information (i.e., $n$ points on the polynomial) to recover the polynomial and the message which is hidden in the constant term of the polynomial (i.e., $f_j(0)$). However, there are only $n - 1$ points available for any outside attacker. This information is insufficient to recover the polynomial. The security of this scheme does not depend on any additional assumption. □

The security of DH schemes has thus far been based some intractability assumptions. Schemes analyzed in the random-oracle model [37] generally rely on the Computational Diffie–Hellman assumption (CDH-assumption) which states that given two values, $g^a$ and $g^b$, a computationally bounded adversary cannot recover the DH secret $g^{ab}$ [38].

**Theorem 2.** The basic GDH protocol is secure provided the CDH-assumption is intractable.

**Proof.** In Round 2, the secret $K_i$ is sent to other group members using the SB scheme which is unconditionally secure. Therefore, we only need to examine whether the adversary can reveal the group key from broadcast information in Round 1. Assume to the contrary that there exists a probabilistic polynomial time algorithm available to the adversary that given $r_i$, where $r_i = g^{k_i} (\text{mod} p)$, for $i = 1, 2, \ldots, n$, outputs $g^{\sum_{i=1}^{n} k_1 k_2 \ldots k_n} (\text{mod} p)$, with a probability which is not negligible. We consider a special case (i.e., for $n = 2$) that given $r_1 = g^{k_1} (\text{mod} p)$ and $r_2 = g^{k_2} (\text{mod} p)$, outputs $g^{2k_1 k_2} (\text{mod} p)$. Then, we have $g^{2k_1 k_2} = g^{k_1 k_2} (\text{mod} p)$. Thus, the adversary found the DH secret $g^{k_1 k_2}$. This result contradicts the CDH-assumption. □

In the following theorem, we want to prove that the authenticated GDH protocol can satisfy the security requirements of group keys presented in Section 2.3.

**Theorem 3.** The authenticated GDH protocol can provide key secrecy, perfect forward secrecy and key independence.

**Proof.**

(a) *Key secrecy:* In Round 2, each member $U_i$ uses the SB scheme to distribute the secret message, $K_i$, to other members. $U_i$ computes the shared DH secrets, $s_{ij} = (y_i r_i)^{k_1 k_2} (\text{mod} p)$, for $j = 1, 2, \ldots, n$, where $y_i$ is a long-term public key of one of the other group members. This ensures that only legitimate group members can recover the secret, $K_i$, and the group key.

(b) *Perfect forward secrecy:* The group key is a function of a set of one-time keys collaboratively chosen by all group members. Therefore, a group key will not be compromised if any of the long-term private keys has been compromised in the future. For example, we consider a special case (i.e., for $n = 2$) that given $r_1 = g^{k_1} (\text{mod} p)$ and $r_2 = g^{k_2} (\text{mod} p)$, the group key is $K = g^{2k_1 k_2} (\text{mod} p)$. This group key, $K$, is a function of one-time keys, $k_1$ and $k_2$. It ensures that any group key will not be compromised if one of the long-term private keys is compromised in the future.

(c) *Key independence:* It is obvious since each group key is a function of random integers chosen by all group members. For example, we consider a special case (i.e., for $n = 3$) that given $r_1 = g^{k_1} (\text{mod} p), r_2 = g^{k_2} (\text{mod} p)$, and $r_3 = g^{k_3} (\text{mod} p)$, the group key is $K = g^{2k_1 k_2 + k_1 k_3 + k_2 k_3} (\text{mod} p)$. Since $k_1$ is a one-time (short-term) DH private key selected by each member $U_i$, the group key will be different for each communication session. □

In the following theorem, we want to show that the authenticated GDH protocol can satisfy the objective of key authentication.

**Theorem 4.** The authenticated GDH protocol can provide the authentication of group keys.

**Proof.** The objective of authentication is that at the end of the protocol, each member is convinced that all group members have shared the same group key. In our proposed three-round protocol, only when all key confirmations have been verified successfully, the group key is agreed by all group members. Since a key confirmation is generated by each group member using his own version of “group key” as one of the inputs of a one-way function, this feature ensures that all group members should share the same group key in order to successfully verify all key confirmations. Let us consider two types of attackers: outsider and insider. For any outside attacker who does not know any long-term private keys of members, he cannot recover the group key (property of Key Secrecy). Therefore, outside attackers cannot generate a valid key confirmation. Let us consider the following insider attack. Suppose that entity $B$ intends to make $A$ to believe that $(A, B, C)$ forms a group; but $B$ impersonate $C$ to participate in the GDH protocol. In Rounds 1 and 2, entity $B$ can impersonate $C$ to send broadcast messages. However, entity $B$ does not know $C$’s long-term private key. $B$ does not know the shared DH secret between $A$ and $C$. Therefore, the forged secret message, $K_c$, selected by $B$ will be deciphered by $A$ into a different value which is unpredictable by $B$. The group key recovered by $A$ is unknown to $B$. This attack can be detected in Round 3 from the key confirmations. □

In the following theorem, we want to prove that the authenticated GDH protocol can resist unknown key-share attack and key compromise impersonation attack presented in Section 2.4.

**Theorem 5.** The authenticated GDH protocol can resist unknown key-share attack and key compromise impersonation attack.
Proof.

- (a) **Unknown key-share attack**: Each shared secret, \( s_{ij} = (y_i f_j)^{x_i-k_j} \pmod{p} \), between \( U_i \) and \( U_j \), involves both members’ long-term private/public keys. This ensures that only legitimate group members who own the corresponding long-term private keys can obtain the same group key and can compute valid key confirmations. Since attackers cannot generate a valid key confirmation, the unknown key share attack is detectable. For example, we consider a special case (i.e., for \( n = 2 \)) that given \( r_1 = g^{x_1} \pmod{p} \) and \( r_2 = g^{x_2} \pmod{p} \), the shared secret, \( s_{12} = (y_1 r_2)^{x_1-k_1} = (y_1 f_1)^{x_1-k_1} \pmod{p} \), between \( U_1 \) and \( U_2 \), involves both members’ long-term private/public keys. This ensures that only \( U_1 \) and \( U_2 \) who own the corresponding long-term private keys can obtain the same group key and can compute valid key confirmations. Any other entity cannot obtain the same group key and can compute valid key confirmations. It is impossible that \( U_1 \) ends up believing that she/he shares a key with \( U_2 \), and although this is in fact the case, while \( U_2 \) mistakenly believes that the key is instead shared with another entity \( E \).

- (b) **Key compromise impersonation attack**: Suppose that \( A \)'s long-term private key is compromised. The adversary who knows \( A \)'s long-term private key can impersonate other entities to \( A \) if the authentication is based on the shared DH long-term secret between two entities. However, since in our proposed authenticated GDH protocol the shared DH secret between two entities involves both a one-time and a long-term secrets of each entity, this attack cannot work properly unless the adversary knows both \( A \)'s long-term and one-time secrets. For example, we consider a special case (i.e., for \( n = 2 \)) that the group communication is between \( U_1 \) and \( U_2 \). If the adversary who knows the long-term private key, \( x_1 \), of \( U_1 \), the adversary can impersonate \( U_1 \) to communicate with \( U_2 \). However, the adversary cannot impersonate \( U_2 \) to communicate with \( U_1 \) unless the adversary who also knows the long-term private key, \( x_2 \), of \( U_2 \).

6. Comparison

In this section, we discuss the performance of our proposed authenticated group Diffie–Hellman key agreement protocol and compare it with other protocols [9, 25, 17, 34]. The protocol proposed by Steiner et al. [9] is based on the discrete logarithm (DL) which uses Diffie–Hellman public-key exchange for members connecting in a logic ring. This protocol does not provide group key authentication. The protocol proposed by Desmedt and Lange [25] is based on the pairing which uses tripartite key exchanges for members. The group key authentication is provided using digital signatures. The protocol proposed by Brecher et al. [17] is based on the DL which uses Diffie–Hellman public-key exchange for members connecting in a tree. The protocol by Cao et al. [34] is based on the paring which uses secret sharing to design group key exchange with universally composability. The group key authentication is provided using aggregate signature. Our proposed protocol is based on the DL which uses Diffie–Hellman public-key exchange and the secret sharing. One unique feature of our proposed protocol is to use the one-way function to provide group key authentication, but other protocols use the digital signature. The computational complexity of generating/verifying a digital signature is far more larger than the complexity of generating a one-way function.

In our proposed protocol, there are modular exponentiation, modular multiplication, one-way function, polynomial interpolation and polynomial evaluation used. Among these operations, the modular exponentiation is the most time-consuming operation. Therefore, we only consider the number of modular exponentiations needed in our proposed protocol. On the other hand, the complexity of each pairing-based multiplication is equivalent to the complexity of a modular exponentiation. For other DL based protocols, we only consider the number of signature generations, signature verifications and modular exponentiations.

We list the comparison in Table 1. From this table, it shows the efficiency of our proposed protocol. In Table 1, methodology/topology is denoted by M/T; cryptographic technology is denoted by Cry.Tech., communication rounds is denoted by Com.Round, authentication is denoted by Auth., computational complexity is denoted by Com.Complexity. Discrete logarithm is denoted by DL, modular exponentiation is denoted by E, multiplication is denoted by M, pairings is denoted by P, signature is denoted by S, and signature verification is denoted by V.

7. Conclusions

We propose a basic GDH protocol and proved that this protocol is secure provided the GDH-assumption is intractable. Then, an authenticated GDH protocol based on the secret sharing scheme and the basic GDG protocol is proposed in the

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GDH3 [9]</td>
<td>Diffie–Hellman and logic ring connection</td>
<td>( n + 1 )</td>
<td>None</td>
<td>DL</td>
<td>((5n - 6)E)</td>
</tr>
<tr>
<td>BDH [25]</td>
<td>Tripartite key exchanges</td>
<td>3</td>
<td>Signature</td>
<td>Pairing</td>
<td>( 2S + 4\log_2 n + 6P + 2\log_2 n ) M</td>
</tr>
<tr>
<td>R-TDH [17]</td>
<td>Diffie–Hellman and tree connection</td>
<td>3</td>
<td>Signature</td>
<td>DL</td>
<td>( 2S + 2(n - 1) + V + (\frac{1}{2}(n - 2)(n - 3) + 2)E )</td>
</tr>
<tr>
<td>ID-SS [34]</td>
<td>ID-based and secret sharing</td>
<td>3</td>
<td>Aggregate signature</td>
<td>Pairing</td>
<td>S + V + 2P</td>
</tr>
<tr>
<td>Our GDH</td>
<td>Diffie–Hellman and secret sharing</td>
<td>3</td>
<td>One-way function</td>
<td>DL</td>
<td>2nE</td>
</tr>
</tbody>
</table>

Note: \( n \) is the number of members.

paper. We prove that the proposed protocol can resist unknown key-share attack and key compromise impersonation attack.

In addition, we show that the protocol can provide key secrecy, perfect forward secrecy and key independence of group keys. The unique feature of our protocol is its efficiency in terms of computation and communications. Our protocol can handle joining of new users and departing of old users easily.

Acknowledgments

The authors would like to thank the reviewers for their suggestions to improve the quality of this paper. This research is supported by the National Natural Science Foundations of China under Grant No. 61103247.

References


