Hierarchical Key Agreement Protocols in Group-user Systems

Shiuh-Jeng Wang¹,* Yuh-Ren Tsai², Chien-Chih Shen², Pin-You Chen²

¹ Department of Information Management Central Police University, TaoYuan, Taiwan 33304
² Institute of Communications Engineering National Tsing Hua University, Hsinchu, Taiwan 30013
*Correspondence: sjwang@mail.cpu.edu.tw

Abstract

In recent years, the ubiquity of communication networks speed up the development of wireless network and the Internet applications. In addition to the common service provided by the telecommunication enterprise, how to connect some distinct users to form a network which allow them to deal with their own works is a popular topic that many scholars engage in, [1] [2] [8]. When constructing the network, the key management is the most important problem that each system has to solve. There are many key agreement protocols which are non-hierarchical. However, the hierarchy is ubiquitous in our real life. In this paper, we have achieved a verifiable hierarchical key derivation scheme using the elliptic curve cryptography and the bilinear mapping function which can achieve the same security level but use less bit number when comparing to the RSA system.

Keywords: Hierarchical key agreement, elliptic curve cryptography, bilinear mapping function, verifiable.

1. Introduction

Since the number of wireless communication users is growing rapidly, centralized network control center have became much busier, and the risk of attack has also increased. Therefore, applications for decentralize systems such as ad hoc networks have been developed. The “self-organized” property of ad hoc networks allows such systems to forgo a central control center. The users in the ad hoc network can manage their own work without the help of the control center.

There are many real-world examples of hierarchy; for instance, in a corporation, employees in different job positions have different abilities and responsibilities in accordance with their rank; when using a network, there are different bandwidths for downloading and uploading in accordance with the collected fees. In our scheme, we separate users into distinct levels, and each level is assigned a level key. Only the user having higher rank can derive the keys belonging to the users of lower rank. When transmitting secret information between users, they use a common key constructed by adjacent users to encrypt and decrypt secret information. In the case that the chairman wants to check secret information transmitted between managers and the rank-and-file employees, the chairman can derive the common key through this protocol and then decrypt the cipher to obtain the message. Other related research, by contrast, uses a key derivation method that does not protect the secret information [8]. Nevertheless, in our scheme, the derived key is the common key constructed by adjacent users, and each user can maintain the secrecy of their own information.

We use the elliptic curve cryptosystem (ECC) and the bilinear mapping function in our protocol. Compared to a RSA or DLP (discrete logarithm problem) having the same bit length, the ECC has higher security and better performance [4].

The remainder of this paper is organized as follows. In Section 2, we present our scheme.
Section 3, analyses are discussed. Finally, conclusions are drawn in section 4.

2. The Proposed Scheme

In this section, we will introduce the proposed scheme, a hierarchical multi-party key agreement protocol, in detail.

2.1 Initializations

First, the system needs a trusted third party to establish parameters, such as \( n = p \cdot q \), where \( p \) and \( q \) are two distinct large prime numbers; then it chooses a number \( e \) which is relative prime to the Euler function \( \varphi(n) = (p-1)(q-1) \) and computes a number \( d \) such that \( e \cdot d \equiv 1 \mod \varphi(n) \). \( e \) will be public information and \( d \) will be secret information of the system.

Let \( E(F_w) \), where \( w \) is a prime number, be a supersingular elliptic curve. For a prime number \( \alpha \), there are points having order- \( \alpha \) in the group \( E(F_w) \). Further, the group \( E(F_{w}) \), where \( l \leq 6 \), has an order- \( \alpha \) additive group \( G_i \). The extended field \( F_{w} \), as a multiplicative group of non-zero elements, also has the same order \( \alpha \); we define it to be the multiplicative group \( G_i \). Let \( B: G_i \times G_i \rightarrow G_2 \) be a bilinear mapping function.

In the setting mentioned above, the trusted third party will set and send an identification key to each user according to the real-world situation. In order to establish the identification key, we define a parameter called identifier: \( id_i = f(ID_i, L_i, G) \); where \( f \) is a hash function, \( ID_i \) is the identity of user \( i \), \( L_i \) is the corresponding level of user \( i \), and \( G \) is the generating point of the group \( G_i \). Then, the trusted third party uses the secret number \( d \) that it computed and multiplies it by the identifier \( id_i \) as the identification key \( s_i (= d \cdot id_i) \). The identification keys are transferred to each user through a secure channel during construction of the system.

2.2 The Protocols

All of the users in the system are arranged in linear order according to their own identity \( ID_i \) and time stamp. If there are users who join or leave the system, the trusted third party merely recomputes the related information between the new adjacent users and updates it. The details of the protocol are as follows:

**Step 1.** Each user \( P_i, i \in \{1, 2, \ldots, m\} \), selects one secret number \( r_i \in \mathbb{Z}_n \); then, each user computes and publishes the value of \( x_i : x_i = e \cdot r_i \cdot G \). The trusted third party also must check all \( x_i, i \in \{1, 2, \ldots, m\} \) to ensure they are all different, or else the trusted third party must ask users having the same secret number to pick another one, so that all of them are totally distinct.

**Step 2.** Each user \( P_i, i \in \{2, 3, \ldots, m-1\} \), and its adjacent users will construct common keys using the concept of Diffie-Hellman key agreement and elliptic curve cryptography, and then compute:

\[
v_i = r_{i-1} \cdot r_i \cdot G + h(r_i \cdot r_{i+1}, G - r_i \cdot G).
\]

Here, \( r_{i-1} \cdot r_i \cdot G \) is the common key which is constructed by \( P_{i-1} \) and \( P_i \), and defined as \( k_{i-1,i} \). \( r_i \cdot r_{i+1} \cdot G \) is the common key that is constructed by \( P_{i+1} \) and \( P_i \) and defined as \( k_{i,i+1} \). The notation \( h \) is a one-way hash function which can map one point of the group \( G_i \) to another one point of the same group. Since there is only one user next to users \( P_i \) and \( P_{i+1} \), the corresponding \( v_i = C_i, i \in \{1, m\} \), where \( C = f(x_1, x_2, \ldots, x_m) \cdot G \) and \( f \) is a secure one-way function.

Considering the verification, each user must compute another parameter:

\[
w_i = s_i \cdot f(x_i, v_i, C) \cdot r_i.
\]

After that, each user publishes the following information, \( (ID_i, L_i, v_i, w_i) \).

**Step 3.** (Authentication) Each user, \( P_i, i \in \{1, 2, \ldots, m\} \), can authenticate one another to ensure whether the other user is valid. The following details apply:
\[ B(A_i \cdot s_i, G) \cdot r_i \cdot \mathcal{I}^{(\mathcal{I})} = B(A_j \cdot s_j, G) \cdot r_j \cdot \mathcal{I}^{(\mathcal{I})}, \]

where \( s_i \) and \( s_j \) are the identification keys corresponding to users \( P_i \) and \( P_j \), and \( i \neq j \); \( A_i \in \mathbb{Z}_p \) and \( A_j \in \mathbb{Z}_q \) are the random numbers chosen by users \( P_i \) and \( P_j \).

**Step 4.** (Level key derivation) The level key, defined as \( K_{\text{level}} \), is established as the common key constructed by the users who are in the last two ranks in level \( l \). For example, we assume that there are three users, \( P_i, P_j, \) and \( P_k \), and they have the relation \( P_i < P_j < P_k \). In addition, \( P_i \in L_{k-1} \) and \( P_j, P_k \in L_k \). The level key \( K_{\text{level}} \) is the common key constructed by \( P_j \) and \( P_k \), such that \( K_{\text{level}} = r_j \cdot r_k \cdot G \).

The following are the details of the key derivation:
\[ k_{i,j+l} = v_i - h(k_{i,j+l} - r_i \cdot G) = v_i - h(r_i \cdot r_{i+l} \cdot G - r_i \cdot G). \]

We notice that \( v_i \) is the public information, and only user \( P_{i+l} \) who has higher rank can obtain \( k_{i,j+l} \) and \( r_i \cdot G \). In addition, due to the characteristics of hash functions, the derivation of the hierarchical key is unidirectional.

**Step 5.** This step introduces the usage of the level key. The level key is an emblem of the hierarchy. The user who is in higher level always has the higher level key. The way to identify the level depends on highest level key the user is able to obtain. For example, if user, \( P_i \), cannot derive the level key of another user \( P_j \), who is in different level, then the level of user \( P_i \) is lower than the level of user \( P_j \).

Next, in situations where communication should be level-to-level, the system should assign a specific user to represent the level and be responsible for communicating with other users in different levels. In our scheme, the system chooses the user who has the highest rank in the corresponding level to be the ambassador. When doing level-to-level communication, there are two steps:
1) The chosen ambassadors corresponding to different levels verify the identity each other, just as they do in step 4.
2) After the identification, they use the corresponding level key to construct the common key:
\[ B(k_{i,j+l}, k_{j,j+l}) = B(G, G)^{(r_{i+j+l} \cdot \mathcal{I})}, \]

where \( k_{i,j+l} \) is the level key of the corresponding level of user \( P_i \) and \( k_{j,j+l} \) is the level key of the corresponding level of user \( P_j \), as the Figure 1 shows.

**Figure 1.** The common key between levels

We note that, \( B(G, G)^{(r_{i+j+l} \cdot \mathcal{I})} \) is the common key used in level-to-level communication. Furthermore, \( k_{i,j+l} \) and \( k_{j,j+l} \) are the level keys corresponding to level 1 and level 2.

### 3. Analyses and Discussions

Our proposed scheme has the following properties:

**Claim 1:** When authenticating the validity of the user by checking the validity of the identification key, the two parties who are running the authentication process merely must obtain random numbers \( A_i \) and \( A_j \) independently, and use the public information \( (ID_i, L_i, ID_j, L_j) \) offered by both of these users. With the bilinear mapping function, they can achieve authentication.

**Proof:** First, the two users exchange the selected random numbers \( A_i \) and \( A_j \), and then they compute the following and determine whether the results are identical. The details:
\[ B(A_i \cdot s_i, G) \cdot r_i \cdot \mathcal{I}^{(\mathcal{I})} = B(A_j \cdot s_i, G) \cdot r_j \cdot \mathcal{I}^{(\mathcal{I})}, \]

There is some information regarding identity and
corresponding level included both in the corresponding identification key \( s_i \) and \( s_j \), so only the valid users who have their own identification key which has the secret information \( d \) can pass the authentication, and both of them can also verify which the identity of the user. In other words, if a valid user claims another identity, the authentication will not be successfully implemented.

**Claim 2:** The process of deriving class keys is a unidirectional derivation in which only a high level user can derive the common keys of the low level users, and get the level key further.

**Proof:** We assume that user \( P_{j-1} \) who has lower rank wants to derive common key, \( k_{j-1,j} \), constructed by \( P_j \) and \( P_{j+1} \), who have higher rank than user \( P_{j-1} \). User \( P_{j-1} \) uses the published information \( v_j \) and the common key \( k_{j-1,j} \) which is known by \( P_{j+1} \), and then \( P_{j-1} \) can execute the following process:

\[
v_j - k_{j-1,j} = h(k_{j-1+1,j} - r_j \cdot G).
\]

As we have shown, \( P_{j-1} \) must next compute the inverse function of the hash function \( h \), but the hash function is unidirectional. As a result, it is difficult to obtain the inverse function of \( h \). So, the scheme fits the property of unidirectional derivation and it is secure.

4. Conclusions

We have proposed a hierarchical key agreement protocol that achieves unidirectional derivation and fits the demand for the hierarchical network structure. It can ensure that only users having higher rank can obtain common keys constructed by the users having lower rank, and then users having higher rank can get the corresponding level key further. Before obtaining the level key, higher level users can obtain a common key of lower level users which is used to encrypt the transmitted data. Thus, users having a higher level can know the transmitted data by decrypting the cipher using the common key they have derived. Obtaining the level key can help the user prove that the he is in a specific level. We also have designed authentication and verification schemes that allow a user to authenticate the identity of other users and to verify information published by other users. This is a useful property for wireless ad hoc networks. Our scheme uses elliptic curve cryptography and the bilinear mapping function to increase efficiency and security.

**Acknowledgements**

This work was supported in part by the National Science Council of the Republic of China under the Grant NSC 97-2221-E-015-001-

**References**


