A Simple and Robust Authenticated Multiple Key Agreement Scheme

Chin-Chen Chang[1, 2], Hao-Chuan Tsai[2], and Yen-Chang Chiu [2]

[1] Department of Computer Science and Information Engineering,
Feng Chia University, Taichung, Taiwan, 40724, R.O.C.
[2] Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan, 621, R.O.C.
e-mail: {ccc, tsaihc, cyc94}@cs.ccu.edu.tw

Abstract

Providing confidential communications and with integrity over an insecure network is an important issue. To achieve these goals, a common key, called session key, has to be established between communicating parties, and several key agreement schemes have been proposed. However, most of them were designed to establish one session key for each communication. To reduce the communication cost, establishing multiple session keys has been proposed in the past decade. Unfortunately, all of them are either insecure or have violated the original requirement, without using one-way hash functions. Herein, we propose a multiple key agreement scheme with a novel architecture, which is different from the previous variants, to enable communicating parties establish multiple session keys

1. Introduction

The primary goal of cryptography is to provide users with confidential communications and with integrity over an insecure public network. In other words, confidentiality ensures that messages are transmitted to authorized parties in secret, and integrity provides that only authorized parties are able to modify information assets. To achieve these basic properties, arbitrary parties who intend to communicate with each other over an insecure network have to establish a common secret key also called a session key. The communicating parties first generate a session key by running an appropriate key agreement scheme to encrypt the transmitted messages for later communications. Hence, to preserve confidentiality and integrity of communications between the two parties, the main problem is how to securely establish a session for an appropriate key agreement scheme. In the past two decades, many key agreement schemes have been proposed [1, 2, 4, 8, 9], but most of them were designed to establish one session key for each communication. It is necessary to generate another session keys to protect later messages, and it wastes communication bandwidth to generate only one session key in each session if the communication parties still need to exchange messages later. Due to the communication cost should be minimized and the required security object should be preserved, establishing multiple session keys [3, 5] in one session is suitable for saving the communication bandwidth.

In 1998, Harn and Lin [6] proposed the first well-known authenticated multiple key agreement scheme which enabled two parties to authenticate each other without using one-way hash functions. However, Yen and Joye [11] pointed out that Harn-Lin's scheme is vulnerable to a forgery attack and then proposed an improved version. Unfortunately, Wu, He, and Hsu [10] demonstrated that Yen-Joye's scheme is still vulnerable to a forgery attack, so they also proposed a modified version. Although Wu-He-Hsu's scheme ensures higher security than the previous schemes, it violates the original requirement of not using one-way hash functions. And in 2001, Harn and Lin [7] proposed another enhanced version which preserved the original requirement. Unfortunately, Zhou, Fan, and Li [12] found that an adversary can impersonate a valid user to derive parts of the valid session keys in Harn-Lin's enhanced scheme. Later, they proposed a modified version to resolve such a security flaw.

According to our observations, the previous schemes suffer from a forgery attack due to a common reason: the transmitted messages are not well protected, so even if a valid user signs these messages, an adversary can still forge such message to derive the multiple session keys or parts of them without being detected. Hence, in this paper, we propose an improved version with a novel architecture which does not need to sign the transmitted messages. Also, the proposed scheme preserves the original requirement, as one-way hash functions are not used.

The rest of this paper is organized as follows. In In Section 2, we propose an improved scheme with a novel architecture and the analyses of the proposed scheme are detailed in Section 3. Finally, conclusions are proposed in Section 4.

978-0-7695-3486-2/08 $25.00 © 2008 IEEE
DOI 10.1109/SecTech.2008.11

2. The Proposed Scheme

According to our observation, such a security flaw is mainly derived from a problem; an adversary can forge arbitrary messages to replace the messages signed by the legal user to derive the multiple session keys, which implies that the integrity of the transmitted messages is not well protected. Herein, we propose an improved version with the novel architecture which provides explicit mutual authentication. The system setup is defined as follows. Let p and q be two large prime numbers, respectively, where $p = 2q+1$, and g be a generator for quadratic residue of the multiplicative group Z_p^*. The proposed scheme involves two phases, the setup phase and key generation phase with authentication. They are described as follows.

Setup phase

The setup phase is involved only once whenever users want to exchange the secret information or messages for later communications. For each legal user, denoted as U, he at first selects n private keys $sk_1, sk_2, ..., sk_n$ from Z_q. Next U computes the corresponding $y_i = g^{sk_i} \bmod p$, where $i = 1, 2, ..., n$. Given x_i, U can construct the n public keys by using n linear equations with respect to $a_1, a_2, ..., a_n$, i.e., $a_1 x_i + a_2 x_i^2 + ... + a_n x_i^n = y_i \bmod p$, for $i = 1, 2, ..., n$. We can use matrix form to represent these equations as follows:

$$\begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix} \begin{pmatrix} x_1 & \dots & x_n \\ \vdots & \ddots & \vdots \\ x_1^n & \dots & x_n^n \end{pmatrix} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix} \bmod p$$

Since each x_i is different from others, i.e., no element $(x_i - x_j) = 0$ exists, for $i, j = 1, 2, ..., n$ and $i \neq j$, which implies that the determinant of M, $\det M = \prod_{1 \le i, j \le n} (x_i - x_j) \neq 0$. The determinant of such a matrix is nonzero, therefore, these equations have a unique solution over the field Z_p.

Key Generation phase

Since the setup phase has been developed successfully, two arbitrary entities can authenticate each other and establish multiple common keys in a three pass interaction. Without loss of generality, we assume that two entities, Alice and Bob, want to establish multiple common keys. The details are described as follows:

First, Alice chooses n random numbers $u_1, u_2, ..., u_n \in_R Z_q$ and then computes $\alpha_i = g^{u_i} \bmod p$, for $i = 1, 2, ..., n$. Next, she generates exchanged values m_{A_i}'s, where $i = 1, 2, ..., n$. and computes the corresponding public values $\eta_i = m_{A_i} \cdot y_{B_i}^{u_i} \bmod p$. In addition, Alice calculates the sum of these n exchanged values, denoted as $M = \sum_{i=1}^{n} m_{A_i} \bmod q$. Subsequently, she computes $\mu_a = (g^M \bmod p \oplus M)$ and sends the computed result along with the previous messages, $\eta_i = m_{A_i} \cdot y_{B_i}^{u_i} \bmod p$, $\alpha_i = g^{u_i} \bmod p$, and the identity of Alice, denoted as ID_A, to Bob.

Upon receiving the transmitted messages, Bob verifies the validity of ID_A. If the format of ID_A is illegal, the key generation phase will be terminated. Otherwise, Bob recovers these n exchanged values by decrypting η_i, i.e., Bob has to use his private keys and the received messages to derive each m_{A_i} by

$$m_{A_i} = \frac{\eta_i}{\alpha_i^{sk_{B_i}}} \bmod p = \frac{m_{A_i} \cdot y_{B_i}^{u_i}}{(g^{u_i})^{sk_{B_i}}} \bmod p,$$

In addition, Bob also sums up all m_{A_i}'s, i.e., $M' = \sum_{i=1}^{n} m_{A_i} \bmod q$, and computes $\mu_a' = (g^{M'} \bmod p \oplus \mu_a)$. If the computed μ_a' equals to the received $M' = \sum_{i=1}^{n} m_{A_i} \bmod q$, then Alice is successfully authenticated by Bob. Next, Bob generates another n random numbers $v_1, v_2, ..., v_n \in Z_q$ and computes $\beta_j = g^{v_j} \bmod p$, for $j = 1, 2, ..., n$. Next, he embeds the other n exchanged values m_{B_j}'s, in the public values $\xi_j = m_{B_j} \cdot y_{A_j}^{v_j} \bmod p$. In addition, Bob calculates the sum of all m_{B_j}'s, denoted as $N = \sum_{j=1}^{n} m_{b_j} \bmod q$, and then computes $\mu_b = (g^N \bmod p \oplus N)$ and the first session key $\mu_{ab} = (\mu_a \oplus M')^N = g^{MN} \bmod p$. Subsequently, Bob sends the computed results $\beta_j = g^{v_j} \bmod p$, $\xi_j = m_{B_j} \cdot y_{A_j}^{v_j} \bmod p$, and $\mu_b = (g^N \bmod p \oplus N)$ along with the identity of Bob, ID_B, to Alice.

After receiving the messages transmitted from Bob, Alice also uses her private keys and the received messages to derive m_{B_j} by

$$m_{B_j} = \frac{\xi_j}{\beta_j^{sk_{A_j}}} \bmod p = \frac{m_{B_j} \cdot y_{A_j}^{v_j}}{(g^{vj})^{sk_{A_j}}} \bmod p .$$

Similarly, Alice sums up all m_{B_j}'s, where $j = 1, 2, ..., n$, i.e., $N' = \sum_{j=1}^{n} m_{B_j} \bmod q$ and computes $\mu_b^{'} = (g^{N'} \bmod p \oplus \mu_b)$. If the computed $\mu_b^{'}$ equals to the received $N' = \sum_{j=1}^{n} m_{B_j} \bmod q$, then Bob will also successfully authenticated and the first session key $\mu_{ab} = (\mu_b \oplus N')^M = g^{MN} \bmod p$ can be established. Subsequently, Alice encrypts the items μ_{ab}, ID_A, and ID_B by using the first session key μ_{ab} and then sends the encrypted result to Bob. Finally, Bob can also use the session key μ_{ab} to decrypt the message to verify the results previously computed.

3. Analysis of Security

We show that the proposed scheme can resist against either passive adversaries or active ones. Before demonstrating the security of the proposed scheme, the queries models of the adversary are described as follows:

• Send(Π_U^i, M) : this query model enables adversary *A* to control all the communications in the proposed scheme. Adversary *A* sends a message *M* to an oracle Π_U^i, where Π_U^i denotes the instance *i* of a user *U*, and returns the response message.

• Reveal(Π_U^i) : this query model denotes known key attacks. Adversary *A* is allowed to send this query to a user oracle Π_U^i. If a session key *SK* has been accepted by Π_U^i, then *SK* is returned to *A*; otherwise, a symbol $\perp$ which denotes *NULL* is returned to *A*.

• Corrupt(*U*) : this query model denotes the property of perfect forward secrecy, i.e., the loss of a long-term key should not reveal the session keys established previously. Adversary *A* sends this query to a user *U*, and then obtains a long-term key of a user.

• Modify(Π_U^i, M) : this query model enables the adversary *A* to interrupt message *M* from the user *U* and then either modify with a new one or forward it to the intended user.

• Test(Π_U^i) : this query model denotes the semantic security of a session key *SK*. During the execution of the proposed scheme, adversary *A* can make a query to the oracle Π_U^i, and at the same time, he can make a single Test query. Upon receiving the query, the oracle Π_U^i generates a hidden bit *b*. If *b* = 1, it returns the true session key *SK* to *A*. Otherwise, a random string, which has the same length with a session key, is returned.

• $\Gamma(\{E, D\}, k, \{m, c\})$: this query allows the adversary A to access the encryption oracle Γ. If A sends an encryption query $\Gamma(E, k, m)$ to Γ, then Γ returns the corresponding ciphertext c to A. If A sends a decryption query $\Gamma(D, k, c)$ to Γ, then Γ returns the corresponding plaintext m to A.

We show that if two entities follow the proposed scheme, they can establish at least $n^2 + 1$ common keys in correctness.

Property 3.1. (Correctness). If two entities follow the proposed scheme, they establish at least $n^2 + 1$ common keys.

Proof. Without any loss of generality, we assume that user U_i and user U_j want to establish multiple keys. For users U_i and U_j, they both have unique solutions in the setup phase to derive the exchanged messages embedded in the transmitted messages $\eta_i = m_{A_i} \cdot y_{B_i}^{u_i} \bmod p$, and $\alpha_i = g^{u_i} \bmod p$. After retrieving the messages, U_i and U_j can derive the corresponding $M' = \sum_{i=1}^{n} m_{A_i} \bmod p$ and $N' = \sum_{j=1}^{n} m_{B_j} \bmod p$ respectively, and then to check whether the verification messages μ_a and μ_b are correct or not. If no faults are detected in this stage, both of users can compute the first common key $\mu_{ab} = (\mu_b \oplus N')^M = g^{MN} \bmod p$ at first. In addition, both of them have exchanged *n* values, hence they can compute $C_1^n \times C_1^n = n^2$ common keys for later communications in a session. Eventually, there are $n^2 + 1$ common keys can be established if two entities follow the proposed scheme.

For robustness (fault tolerance), we show the following property.

Property 3.2. (Robustness). The malicious information will be excluded from the honest users.

Proof. An adversary can deviate the exchanged messages from the scheme in this way. First, he sends "wrong" $\eta_i = m_{A_i} \cdot y_{B_i}^{u_i} \bmod p$, $\alpha_i = g^{u_i} \bmod p$ and $\mu_a = (g^M \bmod p \oplus M)$ so that the victim computes different μ_{ab} and other session key for later communications. In this case, the victim will retrieve these fault results and generate the corresponding messages $\beta_j = g^{v_j} \bmod p$, $\xi_j = m_{B_j} \cdot y_{A_j}^{v_j} \bmod p$, and $\mu_b = (g^N \bmod p \oplus N)$. However, it is computational

infeasible for the adversary to derive the result from the victim since the exchanged messages are well protected, which implies that he cannot establish the first session key. In this stage, the victim can detect the malicious messages.

The next property shows that if the Decision Diffie-Hellman Assumption is hard, then the adversary, with the ability to perform the oracle queries $(Q_{se}, Q_{re}, Q_{co}, Q_{mo}, Q_{te}, \Gamma)$, has no advantage to derive a fresh session key from the proposed scheme. The property is described as follows:

Property 3.3. If the Decision Diffie-Hellman Assumption is hard, then for any $j, 1 \le j \le n$, the real communication transcript $(v_1, v_2, ..., v_n, \beta_j \bmod p, \xi_j, \mu_j)$ of U_j and the simulated one $(v'_1, v'_2, ..., v'_n, \beta'_j \bmod p, \xi'_j, \mu'_j)$ are computational indistinguishable, where $\beta_j = g^{v_j} \bmod p \in Z_P$, $\xi_j = m_{B_j} \cdot y_{A_j}^{v_j} \bmod p$, and $\beta'_j = g^{v'_j} \bmod p \in Z_P$, $\mu'_j = (g^{N'} \bmod p \oplus \mu_j)$.

Proof. Without loss of generality, assume that we want to distinguish the exchanged messages m and m' from $(v_1, \beta_j \bmod p, \xi_j, \mu_j)$ and $(v'_1, \beta'_j \bmod p, \xi'_j, \mu'_j)$. For a simulator, it selects exponents $i, j, k \in_R Z_q^*$ uniformly, and computes $E(m) = [m(g^b)^{ci}, g^i]$ and $E(m') = [mg^{ck}(g^b)^{cj}, (g^a)^k g^j]$, which are based on the public key g^{bc} and a generator g, for some $c \in_R Z_q^*$. In order to produce $m' = m$, we can observe as in the following:

The exchanged message m' is equal to m when $y = g^{ab} \bmod p$ for the triplet $[g^a, g^b, y(= g^x)]$, but if $x \neq ab \bmod q$, the message becomes $m' = mg^{(x-ab)kc} \bmod p$ since the oracle sees it as

$$c = [(g^{bc})^{ak+j} m', g^{(ak+j)}].$$

If $m' = m \bmod p$, then $g^{jc(x-ab)} \equiv 1 \bmod p \Rightarrow jc(x-ab) \equiv 0 \bmod q$, which means that $x \equiv ab \bmod q$. If $x - ab \neq 0 \bmod q$ which implies that $m' \neq m \bmod p$ and $g^{kc(x-ab)} \neq 1 \bmod p$ is a generator of G_q. Hence, the probability ϖ to distinguish m and m' from $(v_1, \beta_j \bmod p, \xi_j, \mu_j)$ and $(v'_1, \beta'_j \bmod p, \xi'_j, \mu'_j)$ can be represented as follows:

$\varpi = |\Pr[E_{g(1^n)}(m) = 1] - \Pr[E_{g(1^n)}(m') = 1]| \le \frac{1}{|q|}$ which is negligible and it is computational indistinguishable for m and m'. Note that $|q|$ is the length of the large prime number q.

Hence for any $v_1, v_2, ..., v_n \in Z_q$, and $\beta'_j = g^{v'_i}, 1 \le j \le n$, the probability $|\Pr[v_1, v_2, ..., v_n, \beta_j \bmod p] - \Pr[v_1 = v'_1, v_2 = v'_2, ..., v_n = v'_n, \beta_j = \beta'_j \bmod p]|$

$$\le \frac{1}{|q|} \cdot \frac{1}{|q|} \cdots \frac{1}{|q|} = \frac{1}{(|q|)^n}.$$

4. Conclusions

In this paper, we have reviewed the previous works up to the present which are either insecure or have violated the original requirement, without using one-way hash functions. Further, we have proposed an improved authenticated multiple-key agreement scheme with a novel architecture. The proposed scheme not only satisfies the original requirement but also establishes $n^2 + 1$ session keys in a session.

5. References

[1]. M. Bellare and P. Rogaway, "Provably Secure Session Key Distribution: The Three Party Case," *Proceedings of 27th ACM Symposium on Theory on Computing*, ACM Press, pp. 57–66, 1995.

[2]. S. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks," *Proceedings of IEEE Symposium on Research in Security and Privacy*, Oakland, California, pp. 72–84, 1992.

[3]. H. Y. Chien and J. K. Jan, "Improved Authenticated Multiple-key Agreement Protocol without Using Conventional One-Way Function," *Applied Mathematics and Computation*, vol. 147, no. 2, pp. 491–497, 2004.

[4]. W. Diffie and M. E. Hellman. "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[5]. L. Harn, "Digital Signatures for Diffie-Hellman Public Keys without Using One-Way Function," *Electronics Letters*, vol. 33, no. 2, pp. 125–126, 1997.

[6]. L. Harn and H. Y. Lin, "An Authenticated Key Agreement Protocol without Using One-Way Functions," *Proceedings of the 8th National Conference on Information Security, Kaohsiung, Taiwan*, pp. 155–160, 1998.

[7]. L. Harn and H. Y. Lin, "Authenticated Key Agreement without Using One-Way Hash Functions," *Electronics*

Letters, vol. 37, no. 10, pp. 629–630, 2001.
[8]. A. Odlyzko, "Discrete Logarithms: the Past and the Future," *Design, Codes, and Cryptography*, vol. 19, no. 2/3, pp. 129–145, 2000.
[9]. Y. M. Tseng, "On the Security of an Efficient Two-Pass Key Agreement Protocol," *Computer Standards and Interfaces*, vol. 26, no. 4, pp. 371–374, Aug. 2004.
[10]. T. S. Wu, W. H. He, and C. L. Hsu, "Security of Authenticated Multiple-Key," *Electronics Letters*, vol. 35, no. 5, pp. 391–392, 1999.
[11]. S. M. Yen and M. Joye, "Improved Authenticated Multiple-Key Agreement Protocol," *Electronics Letters*, vol. 34, no. 18, pp. 1738–1739, 1998.
[12]. H. S. Zhou, L. Fan, and J. H. Li, "Remarks on Unknown Key-Share Attack on Authenticated Multiple-Key Agreement Protocol," *Electronics Letters*, vol. 39, no. 17, pp. 1248–1249, 2003.