CHAPTER 31

Conference Key Agreement Protocol using Oblivious Transfer

Ari Moesriami Barmawi, Shingo Takada, Norihisa Doi

Department of Computer Science, The Graduate School of Science and Technology, Keio University, JAPAN

Key words: conference key, individual key, oblivious transfer

Abstract: The basic idea of our protocol is establishing a conference key based on oblivious transfer which can be used in either asymmetric or symmetric cryptography, such that we can reduce the number of decryptions for the key confirmation without sacrificing the level of security. In our proposed method, we break the conference key into several individual secret keys in accordance with the amount of members within the group. This individual key will be used by each member to sign (encrypt (asymmetrically)) the established conference key in the key confirmation procedure. Then, each member multiplies all signed conference keys and decrypting (asymmetrically) the multiplied signed conference key using the multiplicative inverse of his locally calculated conference key.

Thus, each member only needs to perform one decryption for the key confirmation. Furthermore, by using the individual secret key, each member can directly communicate with each other by a support of the leader, while the leader does not gain any knowledge of messages which is exchanged between the communicating members. The last features can not be found in the previous method except in Li-Pieprzyk's. However, for the key generation we need only a less modular exponentiations than the former.

1 INTRODUCTION

There are many key agreement protocols which have been proposed for establishing session key between more than two users, such as those proposed by Burmester-Desmedt [1], Mike Just et. al. [2], Colin Boyd [3,4], Li-Pieprzyk [5], etc. Suppose for the key confirmation, each member signed

the conference key which is calculated by himself. To verify whether all members are holding the same conference key, each member of the group needs to verify all members signed keys. In this case, the previous protocols need several decryptions which increase computational burden to the group. Besides, the previous proposed protocols except Li-Pieprzyk's, did not provide any facilities for the two members in the group to communicate securely with each other (i.e. other members of the group outside the communicating members learn nothing about messages that have been exchanged between the two communicating members). Our protocol provides this facility, but it is more efficient than the Li-Pieprzyk's, since for the key generation, our protocol needs less modular exponentiations than Li-Pieprzyk's..

We propose a protocol for generating the conference and the individual secret keys at once, using oblivious transfer. Using our method, we can establish a conference key which can be used in either symmetric or asymmetric cryptography.

At the end of our proposed protocol each member of the group can calculate the conference and his individual keys (i.e., a key which has to be kept secret between the leader and a certain member of the group). The conference key is used to encrypt (symmetrically and asymmetrically) messages which can be decrypted by all members of the group. The individual key of member U_i is used to encrypt (symmetrically) messages which can be decrypted (symmetrically) only by member U_i itself and the leader of the group. In the key confirmation, this individual key is used for signing the established conference key. Then, each member can verify that all members have the same conference key by multiplying all signed keys and decrypt (asymmetrically) the result using the multiplicative inverse of his locally calculated conference key. Hence, for the key confirmation, each member has to perform only one decryption. Thus, instead of being used for encrypting/decrypting message symmetrically, the conference key can be used for encrypting message asymmetrically as well.

This individual key can be used for signing a contract. Suppose there is a contract which has to be signed by all members. The leader broadcasts a message, then each member signs this message using his individual key and sends it back to the leader. Furthermore, the leader can verify whether the message he sent has been signed by all members only by multiplying all signed messages and decrypting (asymmetrically) it with the multiplicative inverse of the conference key, instead of decrypting each message (one by one). Thus, in this case the computational burden of the leader will be decreased.

This paper first describes the features of the supporting protocols. Section 2 describes the detailed proposed protocol. Section 3 describes the security

analysis. And the comparison with the previous protocols. Section 4 makes concluding remarks.

2 The Proposed Protocol

In order to realize our proposed protocol, we invoke oblivious transfer [6] and multiplicative-additive share converter proposed by Gilboa [7]. However, due to space we will not describe the detail of these protocols. In our proposed protocol we assumed that the group is predetermined and one of the users is the leader whose commands must be authenticated and has public and secret keys. Thus, the leader's public key should be publicly known.

2.1 Detailed Proposed Protocol

Suppose a group has r members (including the group leader) who are honest. Furthermore, the conference key is actually the product of r individual keys. The individual key of each member is determined by each member's randomly chosen number and the leader's one. For obtaining the conference and individual keys the group has to execute the following protocol:

1. All members and the leader have to agree on the size of their randomly chosen numbers n_i that will determine the conference and their individual keys (suppose the size of each n_i is l bit) and a large strong prime θ . Let $\beta = \Phi(\theta)$ (where $\Phi(\theta)$ is the Euler Totient Function of θ). Furthermore, they choose a number α which is the generator of $\mathbb{Z}_\beta^*$.
2. Let U_1 be the group leader, and U_i (for $i=2, \dots, r$) are members. Suppose each member as well as the leader has their own public and secret keys, where the public key of each member is known only by the leader.
3. Each member U_i chooses any integer $_{\text{dum}_i}$, encrypts it with the leader's public key, and sends it to the leader.
4. Each member privately chooses a number n_i .
5. Suppose U_1 chooses n_1 as his multiplicative share whose size is l (we set n_y as the y^{th} bit of n_i for $(y=1, \dots, l)$ and $s_{1,1}, \dots, s_{1,l}$). He then sets l pairs of $(t_{1,y}^0, t_{1,y}^1)$ where $(t_{1,y}^0 = 2^y n_1 + s_{1,y})$ and $(t_{1,y}^1 = s_{1,y})$. Each member U_i has his own multiplicative share n_i which consists of l bits $(n_{i,1}, \dots, n_{i,l})$. U_1 and all members invokes one out of two oblivious transfer, such that $(n_i n_1 = x_i + x_1)$ (where x_i and x_1 are the secret additive shares of user U_i (for $i=2, \dots, r$) and U_1 respectively) using Gilboa's method. In this case, $x_1 = -\sum_{y=1, \dots, l} s_{1,y}$ and $x_i = \sum_{y=1, \dots, l} t_{i,y} n_{i,y}$.
6. Each member U_i broadcasts $(y_i \equiv \alpha^{x_i + \text{dum}_i} \bmod \beta)$ encrypted with his secret key and y_i itself.

7. The leader verifies whether y_i comes from U_i by decrypting the encrypted message using the member's public key and comparing the value of y_i which is obtained from the signed message and the plaintext one. If they are equal (i.e. the leader verifies that this message was signed by the legitimate sender) then he executes the following procedure:

- Calculating $(\alpha^x_i \bmod \beta \equiv (y_i \alpha^{-dum_i}) \bmod \beta)$
- Calculating $(P \equiv \alpha^{((r-1)x_1 + \sum_{i=2,r} x_i) \bmod \beta})$. The leader chooses a number dum_1 such that P' (where $(P' = [P(\alpha^{dum_1})] \bmod \beta)$) is not congruent to $1 \bmod \beta$. The individual key of the leader is defined as follows: $K_1 \equiv B' \bmod \beta$, where $B \equiv \alpha^{dum_1} \bmod \beta$. Then the conference key K is

$$K = P \bmod \beta \quad (1)$$

where t should be agreed upon in advance by all members of the group and t is not congruent to $0 \bmod \lambda(\beta)$ (where λ is the Carmichael Function [8]).

- Broadcasting the size of $(\alpha^{((r-1)x_1 + dum_1 - \sum_{i=2,r} dum_i) \bmod \beta})$ denotes as u along with $(\alpha^{n_1} \bmod \beta)$ and the signed $H(\alpha^{((r-1)x_1 + dum_1 - \sum_{i=1,r} dum_i) \bmod \beta})$ (where H has to be agreed upon in advance by all members and the leader). Otherwise, he will interrupt the protocol execution, and then repeat the protocol from the beginning.
8. U_1 sends $[\alpha^{((r-1)x_1 + dum_1 - \sum_{i=2,r} dum_i) \bmod \beta}]$ by executing u one out of two obli-vous transfer.
9. Each member U_i verifies whether the leader is a legitimate one by comparing the values of the conference key K by using the following equation:
- $$K = [\alpha^{((r-1)x_1 + dum_1 - \sum_{i=2,r} dum_i) \prod_{i=2,r} y_i} \bmod \beta] \quad (2)$$
10. Each member U_i can calculate its individual key as follows:
- $$K_i = \alpha^{n_i} \bmod \beta \quad (3)$$
- and the leader can calculate U_i 's individual key by using the following equation:
- $$K_i = \alpha^{(x_i + x_1)^t} \bmod \beta \quad (5)$$
11. To verify whether all members (including the leader) hold the same conference key, each party broadcasts $\{K\}_{K_i}$ whose value is equal to $SK_i = K^{K_i} \bmod \beta$. Then each member calculates
- $$K \equiv (\prod_{i=1,r} SK_i)^{(K^{-1}) \bmod \beta} \bmod \beta \quad (6)$$
- If the value of equation (6) and the value of K which is locally calculated by a member are equal then the member can verify that all members hold an equal value of K .

2.2 Communication Among the Members

Our proposed protocol provides the facility for each member to be able to communicate with other member securely via the leader using the individual key. In this case, the leader helps the members to communicate with each other, but he does not any knowledge concerning the messages. The procedure is as follows:

1. First of all we assume that the leader is honest. During the communication, the two communicating members and the leader use asymmetric crypto-system.
2. Suppose member U_i will send a message M (where $M \in \mathbb{Z}_\theta$) to member U_j . Then, U_i sends U_j and the leader $(M^{K_i R_i \bmod \beta} \bmod \theta)$ where R_i and R_j are any integer chosen by U_i and U_j respectively, and $\gcd(R_i, \beta)=1$; $\gcd(R_j, \beta)=1$. Besides, $K_i R_i$ and $K_j R_j$ should not be congruent to $(0 \bmod \lambda(\beta))$.
3. The leader sends U_j message $(M^{R_i K_j \bmod \beta} \bmod \theta)$
4. Member U_j calculates $(M^{R_i K_j K_j^{-1} \bmod \beta} \bmod \theta)$ for obtaining $(M^{R_i \bmod \beta} \bmod \theta)$ and calculate $G = M^{(K_i R_i K_j R_j) \bmod \beta} \bmod \theta$. Furthermore, member U_j sends G to U_i .
5. U_j then calculates: $E = G^{(R_i K_i)^{-1} \bmod \beta} \bmod \theta$, and sends it to U_i along with $(M^{R_i \bmod \beta} \bmod \theta)$.
6. U_i obtains M by calculating: $D \equiv E^{(K_j R_j) \bmod \beta} \bmod \theta \equiv M \bmod \theta$.
7. To verify whether U_i is the legitimate member, U_j compares the values of $(M^{R_i \bmod \beta} \bmod \theta)$ sent by the leader and U_i . If they are equal then U_i is the legitimate member, otherwise he is an adversary.

Since the leader does not know the values of R_i and R_j , it is not possible for him to obtain M .

3 Security Analysis

This section discusses the security of our proposed scheme against passive and active intruder.

3.1 Passive Attacks

First, we assume that all members of the group (including the leader) are honest. Passive attacks whose aim is key recovery for a given session involves eavesdropping on message passed between participants for that session.

Then, the conference key recovery will be successful if the adversary can guess t and $a^{(r-1)x_1 + \sum_{i=2,r}^{dum} x_i} \bmod \beta$ after eavesdropping y_i and u .

If we assume that all members are honest and the size of t is $|t|$ bits, then there are $2^{|t|}$ possible values of t . Thus, the probability for obtaining the

value of K is equal to $(1/((\lambda(\beta)-1) \cdot 2^{l-1}))$ (where $(\lambda(\beta))$ is the Carmichael Function of β [8]). But since the largest possible order of an integer modulo β is $(\lambda(\beta)-1)$ while $((2^{l-1})(\lambda(\beta)-1)) > \lambda(\beta)-1$, the probability for obtaining K is equal to $1/(\lambda(\beta)-1)$. Thus, the protocol will be secure if we choose a large strong prime of β such that $\lambda(\beta)$ is large as well.

3.2 Active Attack

Active attack is usually defined as an impersonation attack. Recall that a key agreement protocol is successful if each of the members accept the identity of the other and terminate with the same key. In our proposed protocol, the adversary who will impersonate a member of the group (for example member U_i) may choose his/her random number n_i' , performing Gilboa's method with the leader for obtaining his/her additive share x_i' and calculating y_i' . Thus, the probability for impersonating the legitimate member depends on the probability of finding t and the secret key of U_i . Suppose the size of t is $|t|$ and the probability for finding U_i 's secret key is δ , then the probability for breaking the protocol is $[(\delta)/(2^{l-1})]$.

4 Comparison with Previous Proposed Protocols

Suppose that for the key confirmation each member of the group and the leader has to broadcast the conference key signed (asymmetrically) with each member's secret key. Thus, for verifying that all members of the group and the leader hold the same conference key, each member has to decrypt all encrypted messages which are sent by the other members. Using the previous protocols, this will contribute a higher computational burden for all members of the group in the key confirmation if they use signature such as RSA signature to sign the conference key, because each member has to decrypt several signed messages. In our proposed protocol each member signs/encrypts (asymmetrically) the established conference key using his individual key and broadcasts it. A member can verify whether all members have an equal value of conference key by comparing the conference key which is signed by all members and the one he has calculated. This will reduce the computational burden of each member for the key confirmation, since using our proposed protocol each member needs to perform only one (asymmetric) decryption.

Suppose a member of a group U_i is going to propose a project, and he asks other members for approval of the proposal. Using the previous schemes, U_i has to do $(r-1)$ verifications to check whether all members of the group agree or not. It means that U_i has to do $(r-1)$ (asymmetric) decryptions. This

will increase the computational burden for U_i . In our proposed scheme, each member of the group has his/her own individual secret key which can be used to sign his/her agreement. So, all members will sign the agreement and U_i will verify all members agreement by doing *one* verification which means that he/she needs to perform only one (asymmetric) decryption for verifying all members' agreements.

Compared with Li-Pieprzyk's protocol our proposed protocol is more efficient, since using Li-Pieprzyk protocol, for establishing the conference key, each member has to perform about $(2r^2 + 3r + 1)$ exponentiations, but using our proposed method each member needs to perform not more than 10 exponentiations (roughly) and $l+u$ oblivious transfers.

5 Conclusion

We proposed a new key agreement protocol which is based on oblivious transfer. Our proposed scheme introduced individual keys which can be used by each user to sign a common message, which is not included in the previously proposed protocols. By using individual keys, we can reduce the number of verifications and also reduce the computational burden for the verifier. Besides, using this individual key a member can communicate securely with the other members via the leader, while the leader can not gain any knowledge about the message sent among his members.

Our proposed scheme will be secure as long as the value of t and the secret number chosen by each member are kept secret.

References

- [1] Burmester, M. and Desmedt, Y. G., *Efficient and Secure Conference Key Distribution*, Proceeding of Security Protocols International Workshop Cambridge, United Kingdom, Springer-Verlag, LNCS 1189, April 1996, pp. 119-129.
- [2] Just, M. and Vaudenay, S., *Authenticated Multy-Party Key Agreement*, Advances in Cryptology ASIACRYPT '96, Springer-Verlag, LNCS 1163, 1996.
- [3] Boyd, C., *On key agreement and Conference Key Agreement*, Proceeding of Information Security and Privacy Australasian Conference (ACISP), Springer-Verlag, LNCS 1270, 1997, pp 294-302.
- [4] Boyd, C., *Towards a Classification of Key Agreement Protocols*, Proceeding of Computer Security Foundation Workshop, 1995, pp. 38-43.
- [5] Li, C. and Pieprzyk, J., *Conference Key Agreement from Secret Sharing*, Proceeding of ACISP 1999, Springer-Verlag, pp 64-76.
- [6] Goldreich, O., *Secure Multy-Party Computation*, Working Draft, Download from the Internet, June 1998.
- [7] Gilboa, N., *Two RSA Key Generation*, Crypto 99, LNCS1666, Springer Verlag, 1999, pp. 116-129.
- [8] Adler, A. and Coury, John E., *The Theory of Numbers*, Jones and Barlett, 1995.