多重金鑰之認證交換協定
Authenticated Key Agreement Protocol for Exchanging n^2 Keys without Using One-way Hash Function

Chih-Jen Huang†(黃智任) Shih-Hsu Chang†(張世旭) Wen-Hsing Hsu†(許文星)
ren@snoopy.ee.nthu.edu.tw jang@snoopy.ee.nthu.edu.tw whhsu@mercury.ee.nthu.edu.tw
† Department of Electrical Engineering, National Tsing Hua University,
101, Section 2, Kuangfu Rd., Hsinchu 300, Taiwan, R.O.C.(清華大學電機系)

Abstract

An authenticated multiple-key agreement protocol without using a one-way function is proposed. The authors utilize an additional Diffie-Hellman mechanism to eliminate a known weakness. Benefited by the mechanism, the proposed protocol is capable of withstanding the known-key attack, even if all n^2 session keys are adopted. Conventionally, only n^2-1 keys should be adopted for the reason of the known-key attack. This scheme thus has a higher efficiency of exchanging session keys.
Keywords: Cryptography, Authenticated key agreement protocol, Known-key attack

1 Introduction

An authenticated key agreement protocol is a mechanism for two users to authenticate each others and exchange a session key simultaneously. There are many situations that need lots of session keys for two parties to securely communicate. In order to exchange n^2-1 keys, an authenticated key agreement protocol, based on Diffie-Hellman key distribution scheme [1], was proposed without using a one-way hash function [4]. Yen and Joye [8] however found a security weakness of this protocol and proposed an improved version. They add a constraint on the temporary random public keys. The improved one still suffers from the similar weakness [7]. The reason comes from their providing no authentication of integrity for temporary random public keys. Wu etc. suggested eliminating the weakness by employing an additional one-way hash function [7]. Computing the one-way hash function nevertheless is a heavy load. In general, the security of most one-way functions are based on the complexity of analyzing an iterated simple function. Their security may turn out to be vulnerable to some special attacks later [3]. Besides, if a one-way function is adopted in the protocol, the overall security will rely on both the one-way function and the Diffie-Hellman scheme. In this paper, the security of proposed scheme is based on the Diffie-Hellman mechanism and can be easily analyzed and understood.

In order to reduce the load and overcome the weakness, we propose an improved protocol without using a one-way hash function. We add an additional Diffie-Hellman mechanism to provide authentication of the temporary random public keys. Such an improvement can both eliminate the known weakness and prevent attackers from using the known-key attack. In addition, all n^2 session keys can be utilized without the security risks.

In the following sections we first preview the Yen-Joye scheme and then propose an improvement in section 3. The security analysis of the improved scheme about the weakness [8, 7] and the known-key attack [5] is detailed discussed in section 4. The conclusion is given in section 5.

2 Yen-Joye scheme [8]

There are two phases: the authentication phase and the key generation phase. In the first place, users exchange n random public keys in an authenticated way. Secondly, they generate the shared n^2-1 session keys. In the following, P is a large prime and α is a primitive element over $GF(P)$. User A has a long-term secret key x_A and a corresponding public key $y_A = \alpha^{x_A} \bmod P$. User B has a long-term secret key x_B and a corresponding public key $y_B = \alpha^{x_B} \bmod P$.

In the authentication phase, user A computes the n (here we use $n = 2$ for simplicity) temporary random public keys $r_{A1} = \alpha^{k_{A1}} \bmod P$ and $r_{A2} = \alpha^{k_{A2}} \bmod P$, where k_{A1} and k_{A2} are ran-

domly selected such that

$$\lceil P/2 \rceil \leq r_{A1}, r_{A2} \leq P - 1. \quad (1)$$

Note that employing the above constraint on r_{A1} and r_{A2} prevents another pair of r_{A1}, r_{A2} from satisfying the verification equation. However, it does not completely eliminate the security weakness [7].

For certifying the two numbers r_{A1} and r_{A2}, user A then utilizes one of the signature schemes in reference [3] as

$$s_A = x_A - r_A\, k_A \bmod (P-1), \quad (2)$$

where $k_A = k_{A1} + k_{A2} \bmod (P-1)$ and $r_A = r_{A1}\, r_{A2} \bmod P$. Once A completes above computation, she then sends $\{r_{A1}, r_{A2}, s_A, cert(y_A)\}$ to user B, where $cert(y_A)$ is the certificate of A's public key. User B similarly sends $\{r_{B1}, r_{B2}, s_B, cert(y_B)\}$ to user A. Upon the receipt of A's messages, user B verifies the authenticity and the integrity of r_{A1} and r_{A2} by checking the following equation:

$$y_A \stackrel{?}{\equiv} (r_{A1}\, r_{A2})^{r_{A1}\, r_{A2}} \alpha^{s_A} \bmod P. \quad (3)$$

If it holds, user B generates the shared session keys as follows:

$$\begin{aligned} K_1 &= r_{A1}^{k_{B1}} \bmod P, \\ K_2 &= r_{A2}^{k_{B1}} \bmod P, \\ K_3 &= r_{A1}^{k_{B2}} \bmod P, \text{ and} \\ K_4 &= r_{A2}^{k_{B2}} \bmod P. \end{aligned}$$

User A similarly verifies and generates the same shared session keys, but only three of them should be adopted for avoiding the known-key attack [2]. As described above, A and B share three authenticated session keys.

The improved scheme is still vulnerable to the similar attack in [8], as claimed by Wu et al. [7]. They state that the constraint (1) can not prevent attackers from finding another pair $\{r'_{A1}, r'_{A2}\}$ which satisfies both the equations (2) and (3), since the probability for any attacker to find another valid pair is greater than 1/18. They suggested eliminating this weakness by simply employing an additional one-way hash function h in the signature generation and verification equations. That is to replace equations (2) and (3) with

$$s_A = x_A - h(r_{A1}, r_{A2})\, k_A \bmod (P-1), \quad (4)$$

$$y_A \stackrel{?}{\equiv} (r_{A1}\, r_{A2})^{h(r_{A1}\, r_{A2})} \alpha^{s_A} \bmod P. \quad (5)$$

Such a modification violates the original requirement without using a one-way hash function in the scheme, although the new scheme apparently provides higher security strength for mutual authentication.

3 The proposed scheme

Since modular exponentiation function can be considered as one-way function, the one-way hash function h could be replaced by the following functions for A and B separately.

$$h'_A = (y_B)^{k_{A1}}\, r_{A2} \bmod P \quad (6)$$

$$h'_B = (y_A)^{k_{B1}}\, r_{B2} \bmod P \quad (7)$$

The main difference between this scheme and the other schemes [4, 8] is the addition of Diffie-Hellman mechanism into the r_A in the equation (2). Since the public keys $\{y_A, y_B\}$ are necessary for computing the above equations, they must be known in advance of the computation.

For user A and user B, s_A and s_B are changed to:

$$s'_A = x_A - h'_A\, k_A \bmod (P-1). \quad (8)$$

$$s'_B = x_B - h'_B\, k_B \bmod (P-1). \quad (9)$$

Then A sends $\{r_{A1}, r_{A2}, s'_A, cert(y_A)\}$ to B, and B similarly sends $\{r_{B1}, r_{B2}, s'_B, cert(y_B)\}$ to A. Because user B knows x_B, he also has the ability to calculate h'_A, which is selected by the user A, with the following equation:

$$h'_A = (r_{A1})^{x_B}\, r_{A2} \bmod P. \quad (10)$$

The h'_A thus is a common authentication key between A and B. It can be utilized for verifying the authenticity of shared session keys as follows:

$$y_A \stackrel{?}{\equiv} (r_{A1}\, r_{A2})^{h'_A}\, \alpha^{s'_A} \bmod P. \quad (11)$$

If it holds, then B accepts that K_1, K_2, K_3, and K_4 are the four session keys authenticated between A and B. Benefited by the common keys h'_A and h'_B, this protocol provides the authentication of integrity for temporary random public keys.

Similarly, user A can calculate h'_B with her private x_A and she can verify the authenticity of the shared session keys by the following equations.

$$h'_B = (r_{B1})^{x_A}\, r_{B2} \bmod P, \quad (12)$$

$$y_B \stackrel{?}{\equiv} (r_{B1}\, r_{B2})^{h'_B}\, \alpha^{s'_B} \bmod P. \quad (13)$$

Consequently, A and B agree with each other on the authenticated four session keys.

4 Security analysis and discussion

For an attacker to impersonate user A, he can firstly choose a k_{A1} and a k_{A2} to find out an h'_A from equation (10), and then he must find out the s'_A satisfying the equation (11). It is apparently a discrete logarithm problem. Or he may choose an s_A firstly, and then he have to find out the r_{A1} and the r_{A2} satisfying the equation (11). It seems however to be a more hard problem than the discrete logarithm problem. In addition, if the Galois Field $GF(P)$ does not satisfy two conditions, then solving the discrete logarithm over $GF(P)$ is computationally feasible. Therefore, the two conditions have to be considered. First, $P-1$ should have a large prime factor q for the security of Pohlig-Hellman attack [6]. Secondly, the generator α of a group G_q of order q should have order q.

In order to clarify the security strength, the security weakness found by Yen and Joye [8] is examined. If the weakness exists, the attacker who knows r_{A1}, r_{A2}, and s_A can discover the other values r'_{A1} and r'_{A2} satisfying the following conditions.

$$(r'_{A1})^{x_B} r'_{A2} = (r_{A1})^{x_B} r_{A2} \bmod P \bmod (P-1), \quad (14)$$

$$r'_{A1} r'_{A2} = r_{A1} r_{A2} \bmod P. \quad (15)$$

The attackers however cannot calculate h'_A from equation (6) or (10), for they do not know k_{A1} or x_B. They cannot calculate h'_A from equation (8), for they do not know k_A and x_A. To find out h'_A in equation (11) is equivalent to solve the discrete logarithm problem. As a result, the attacker can not acquire the value h'_A for deriving the r'_{A1} and r'_{A2} from equation (14), thus he cannot impersonate user A by this way. Even if h'_A is revealed, to find out the (r'_{A1}, r'_{A2}) satisfying the above equations is apparently hard. Hence, our scheme is capable of withstanding the attacks in references [8, 7]. The constraint in equation (1) is no longer necessary.

For the known-key attacks in [5, 2], the security analysis of the modified scheme is given below. Users A and B have the four session keys:

$$K_1 = \alpha^{k_{A1}k_{B1}} \bmod P,$$
$$K_2 = \alpha^{k_{A2}k_{B1}} \bmod P,$$
$$K_3 = \alpha^{k_{A1}k_{B2}} \bmod P, \text{ and}$$
$$K_4 = \alpha^{k_{A2}k_{B2}} \bmod P,$$

where

$$(k_{A1} + k_{A2})\, h'_A = (x_A - s'_A) \bmod (P-1),$$

$$(k_{B1} + k_{B2})\, h'_B = (x_B - s'_B) \bmod (P-1).$$

Note that h'_A and h'_B, which are also common session keys shared between the two users, are computable for user A and user B only; therefore, even if all the four session keys were compromised, attackers could not easily derive the Diffie-Hellman public key ($\alpha^{x_A x_B} \bmod P$) from the following:

$$(K_1 K_2 K_3 K_4)^{h'_A h'_B} = (y_A)^{-s'_B} (y_B)^{-s'_A} \alpha^{s'_A s'_B} (\alpha^{x_A x_B}) \bmod P. \quad (16)$$

The modification consequently immunizes the scheme against the known-key attack, even if all the four session keys are compromised. Since only three session keys are used in references [4, 8, 2] for avoiding the known-key attack, they can use only three keys. Therefore, a higher efficiency of sharing session keys is successfully achieved in this paper. This result can also be further generalized to share n^2 session keys. It is noted that in references [8, 7, 2] they can use n^2-1 session keys for the security reason.

5 Conclusion

We have proposed an authenticated key agreement protocol for exchanging n^2 keys without using a one-way hash function. There are two Diffie-Hellman mechanisms in our scheme. One is responsible for authenticating the temporary public keys, the other is used for exchanging multiple session keys. Our improvement successfully eliminates the security weakness that the original scheme does not provide authentication of temporary random session keys. In addition, the proposed scheme can withstand the known-key attack, even if all shared n^2 session keys are adopted. In the previous implements, only n^2-1 keys should be used for avoiding the known-key attack. Consequently, a higher efficiency of exchanging keys is achieved, and an authenticated multiple-key agreement protocol without using a one-way function is described.

Acknowledgement

This work was supported partially by the National Science Council (NSC 88-2213-E-007-050) of the Republic of China.

References

[1] W. Diffie and M. Hellman, "New directions in cryptology," IEEE Transactions on Information Theory, Vol. IT-22, No. 6, pp. 644–654, 1976.

[2] L. Harn, "Modified key agreement protocol based on the digital signature standard," Electronics Letters, Vol. 31, No. 6, pp. 448–449, 1995.

[3] L. Harn, "Digital signature for Diffie-Hellman public keys without using a one-way function," Electronics Letters, Vol. 33, No. 2, pp. 125–126, 1997.

[4] L. Harn and H.Y. Lin, "An authenticated key agreement protocol without using one-way function," In Proc. 8th National Conf. Information Security, pp. 155–160, Kaohsiung, Taiwan, May 1998.

[5] K. Nyberg and R.A. Rueppel, "Weaknesses in some recent key agreement protocols," Electronics Letters, Vol. 30, No. 1, pp. 26–27, 1994.

[6] S.C. Pohlig and M.E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," IEEE Transactions on Information Theory, Vol. IT-24, pp. 106-110, 1978.

[7] T.S. Wu, W.H. He, and C.L. Hsu, "Security of authenticated multiple-key agreement protocols," Electronics Letters, Vol. 35, No. 5, pp. 391–392, 1999.

[8] Sun-Ming Yen and M. Joye, "Improved authenticated multiple-key agreement protocol," Electronics Letters, Vol. 34, No. 18, pp. 1738–1739, 1998.