ICS2002

Workshop on Cryptology and Information Security

Improved authenticated multiple-key agreement protocol without using conventional one-way function

Abstract

An authenticated multiple-key agreement protocol enables two entities to authenticate each other and construct multiple common keys in a two-pass interaction. Since Harn and Lin proposed the first multiple key-agreement without using a conventional hash function, there are several works in the literature. In 2001, Yen, Sun, and Hwang proposed an improved scheme that adopted the system timestamp to detect the replay message. Here, the authors show that an impersonator can easily forge message without detection, and can establish common session keys with the communicating party. To overcome the weakness, we propose an improved scheme. Compared with Harn-Lin's scheme and the previous variants, our scheme achieves better key utilization.

Index term: Cryptography, key agreement.

Dr. Hung-Yu Chien

Department of Information Management,

NanKai College,

NanTou, Taiwan, R.O.C.

E-mail: *redfish6@ms45.hinet.net*

Improved authenticated multiple-key agreement protocol without using conventional one-way function

Hung-Yu Chien[1], Jinn-Ke Jan[2]

1. *Department of Information Management, NanKei College, NanTou, Taiwan, R.O.C.*

2. *Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan, ROC.*

**Corresponding E-mail: redfish6@ms45.hinet.net*

Abstract

An authenticated multiple-key agreement protocol enables two entities to authenticate each other and construct multiple common keys in a two-pass interaction. Since Harn and Lin proposed the first multiple key-agreement without using a conventional hash function, there are several works in the literature. In 2001, Yen, Sun, and Hwang proposed an improved scheme that adopted the system timestamp to detect the replay message. Here, the authors show that an impersonator can easily forge message without detection, and can establish common session keys with the communicating party. To overcome the weakness, we propose an improved scheme. Compared to Harn-Lin's scheme and the previous variants, our scheme achieves better key utilization.

Keywords: Cryptograph, key agreement.

1. Introduction

In 1998, Harn and Lin [7] noticed that the conventional one-way function is widely employed in many digital signature schemes [2-5]. In these schemes, the system will become insecure because of the forgery attacks if the conventional one-way function is not used [7, 8, 12]. Furthermore, they also noticed that the security of these conventional one-way hash

functions, like MD5 [8], is based on the complexity of analysis of iterated functions but is not on a public hard problem [2, 3, 12] (the discrete logarithm problem is a public hard problem, and can be seen as a one-way function.) So, it may seem very difficult to break the security of these conventional one-way functions at the beginning, but it may become insecure to some special attacks later [8]. Therefore, Harn and Lin first proposed the authenticated key agreement protocol without using the conventional one-way function [7]. Moreover, their scheme greatly enhances the efficiency of key agreement by allowing two entities to establish multiple keys instead of one common key in a two-pass interaction [7].

Later, Yen and Joye [9] found that the attacker could easily forge, with high probability, the signature of the exchanged public keys in the Harn-Lin scheme. From this observation, Yen and Joye proposed their modified version. However, Wu et al. [10] found the same weakness in Yen-Joye's modified version. Wu et al. finally proposed their solution by exploiting the conventional one-way function. Unfortunately, this solution violates Harn-Lin's original requirement of no conventional one-way function. Therefore, Yen, Sun, and Hwang proposed an improved version without using the one-way function. The scheme adopts the timestamp to detect the replay message and to verify the authentic message.

In this article, we show that an impersonator can easily forge the message without detection, and can share common session keys with the communicating party. That is, the Yen-Sun-Hwang scheme is not secure. We also propose an improved scheme to overcome the weakness. The rest of this article is organized as follows. In Section 2, we briefly review Harn-Lin's scheme, Yen-Joye's modified version, Wu et al.'s scheme, and Yen-Sun-Hwang's scheme. In Section 3, we demonstrate that an impersonator can easily forge valid message and can establish common session keys with the communicating party. In Section 4, we describe our improved scheme, examine its security and discuss the key utilization. Finally, Section 5 concludes this article.

2. Review of previous works

In this section, we review the main ideas of those previous works [7, 9, 10, 14].

Harn-Lin's Scheme:

The Harn-Lin scheme enables two entities to authenticate each other and to develop multiple common keys in a two-pass interaction. In the first pass, each entity generates and exchanges n public values in authenticated manner. After exchanging the authenticated messages, two entities verify the received messages and then generate n^2-1 keys [7], like the Diffie-Hellman [1] approach, in the second pass.

By taking a simple example of $n=2$, we introduce the idea of Harn-Lin's scheme as follows. The system initially publishes a large prime p and a primitive element α over $GF(p)$. Assume A and B be the two entities to authenticate each other and share multiple keys. The long-term secret key for A is x_a, and $cert(y_a)$ is the certificate of A's long-term public key $y_a = \alpha^{x_a} \bmod p$. The long-term secret key, long-term public key and certificate of the public key for B are $\{x_b, y_b, cert(y_b)\}$. Firstly, A randomly selects two secret numbers secrets k_{a_1} and k_{a_2}, and then computes their corresponding publics $r_{a_1} = \alpha^{k_{a_1}} \bmod p$ and $r_{a_2} = \alpha^{k_{a_2}} \bmod p$. Entity A then has his signature of these two publics by computing $s_a = x_a - (r_a)\cdot(k_{a_1} + k_{a_2}) \bmod p-1$, where $r_a = \alpha^{r_{a_1} r_{a_2}} \bmod p$. A finally sends $(r_{a_1}, r_{a_2}, s_a, cert(y_a))$ to B. Proceeding in a similar approach, B computes and sends $(r_{b_1}, r_{b_2}, s_b, cert(y_b))$ to entity A. After receiving messages from entity A, entity B verifies them by checking whether the following equation holds

$$y_a \equiv (r_{a_1} r_{a_2})^{r_a} \cdot \alpha^{s_a} \bmod p. \quad (1)$$

Entity A also verifies the messages received from B. If both A and B succeed in

their verifications, then they can derive four common keys: $K_1 = r_{a_1}^{k_{b1}} = r_{b_1}^{k_{a1}} = \alpha^{k_{a_1}k_{b_1}} \bmod p$, $K_2 = r_{a_1}^{k_{b2}} = r_{b_2}^{k_{a1}} = \alpha^{k_{a_1}k_{b_2}} \bmod p$, $K_3 = r_{a_2}^{k_{b1}} = r_{b_1}^{k_{a2}} = \alpha^{k_{a_2}k_{b_1}} \bmod p$, $K_4 = r_{a_2}^{k_{b2}} = r_{b_2}^{k_{a2}} = \alpha^{k_{a_2}k_{b2}} \bmod p$. But, A and B will only use three of these four keys to preserve perfect forward secrecy [7, 11, 13].

Yen-Joye's scheme:

Later, Yen and Joye [9] found that an attacker can forge A's message by finding some integer r'_{a_1} and r'_{a_2}, such that $r_{a_1} \cdot r_{a_2} = r'_{a_1} \cdot r'_{a_2} \bmod p$. We can easily see that such r'_{a_1} and r'_{a_2} still satisfy Equation (1). They showed an easy approach to deriving such r'_{a_1} and r'_{a_2} by finding a small factor q of r_{a_1} (or r_{a_2}). Then the attacker lets $r'_{a_1} = r_{a_1}/q$ and $r'_{a_2} = r_{a_2} \cdot q$ to have $r_{a_1} \cdot r_{a_2} = r'_{a_1} \cdot r'_{a_2} \bmod p$. Such a r'_{a_2} will be smaller than p with high probability when q is small. Therefore, from the eavesdropped r_{a_1} and r_{a_2}, an attacker can easily derive r'_{a_1} and r'_{a_2} to have a successful forgery attack on the Harn-Lin scheme. To conquer this insecurity, Yen and Joye proposed their modified scheme by limiting r_{a_1} and r_{a_2} in the range $[p/2, p-1]$ since 2 is the smallest factor of either r_{a_1} or r_{a_2}. They also replaced the signature equation as: $s_a = x_a - (r_{a_1} r_{a_2}) \cdot (k_{a_1} + k_{a_2}) \bmod p-1$. Accordingly, the new verification equation becomes $y_a \equiv (r_{a_1} r_{a_2})^{r_{a_1} r_{a_2}} \cdot \alpha^{s_a} \bmod p$.

Wu et al.'s improved version:

Later, Wu, He and Hsu examined the combinations of factors for the pair (r_{a_1}, r_{a_2}) instead of just considering one small factor q of r_{a_1} or r_{a_2}. They found that an attacker

can forge successfully with a probability greater than 1/18 [10] in Yen-Joye's version. To conquer the insecurity, Wu et al proposed their improvement by incorporating the conventional one-way function into their modified signature equation and verification equation as $s_a = x_a - h(r_{a_1}, r_{a_2}) \cdot (k_{a_1} + k_{a_2}) \bmod p-1$ and $y_a \equiv (r_{a_1} r_{a_2})^{h(r_{a_1}, r_{a_2})} \cdot \alpha^{s_a} \mod p$, respectively. Unfortunately, this modification violates Harn-Lin's original requirement of using no conventional one-way function [7].

Yen-Sun-Hwang's scheme:

To preserve Harn-Lin's requirement, Yen-Sun-Hwang's improved scheme has the following signature generation equation and the verification equation.

$$s_a = x_a - (r_{a_1} \oplus r_{a_2}) \cdot (k_{a_1} + k_{a_2}) \bmod p-1 \quad (2)$$

$$y_a \equiv (r_{a_1} r_{a_2})^{r_{a_1} \oplus r_{a_2}} \cdot \alpha^{s_a} \mod p. \quad (3)$$

Yen, Sun, and Hwang also noticed that if an attacker intercepts a valid message $\{r_{a_1}, r_{a_2}, s_a, Cert(y_a)\}$, then he can impersonate A and then replays the message to B such that B believe he is A, even the attacker does not know the secret session keys. To overcome this weakness, they adopted the timestamp to refine the signature generation equation and the verification equation as follows, where $Time_a$ is A's current timestamp.

$$s_a = x_a - (r_{a_1} \oplus r_{a_2} \oplus Time_a) \cdot (k_{a_1} + k_{a_2}) \bmod p-1 \quad (4)$$

$$y_a \equiv (r_{a_1} r_{a_2})^{r_{a_1} \oplus r_{a_2} \oplus Time_a} \cdot \alpha^{s_a} \mod p. \quad (5)$$

3. Impersonation attack and key compromise

In this section, we show that an attacker can easily impersonate A and can establish

common session keys with B. That is, Yen-Sun-Hwang's scheme is not secure.

Suppose that the attacker has eavesdropped a valid message $\{r_{a_1}, r_{a_2}, s_a, Time_a, Cert(y_a)\}$ from the network. The attacker chooses a random number k'_{a_1} and lets $r'_{a_1} = \alpha^{k'_{a_1}} \bmod p$. Then he lets $r'_{a_2} = (r_{a_1} \cdot r_{a_2})(r'_{a_1})^{-1} \bmod p$ and $Time_{a'} = r'_{a_1} \oplus r'_{a_2} \oplus r_{a_1} \oplus r_{a_2} \oplus Time_a \bmod p-1$. For each randomly chosen k'_{a_1}, there will be a corresponding pair $\{r'_{a_1}, r'_{a_2}, Time_{a'}\}$. The attacker can find as many such pairs as he wish, and choose the suitable pairs, according to the timestamp. The attacker can wait until $Time_{a'}$, and sends the message $\{r'_{a_1}, r'_{a_2}, s_a, Time_{a'}, Cert(y_a)\}$ to B. We can easily check that B will accept this message and responds the message $(r_{b_1}, r_{b_2}, s_b, Time_b, cert(y_b))$ to A. Finally, the attacker will share two common session keys $K_1 = \alpha^{k'_{a1} k_{b1}} \bmod p$ and $K_2 = \alpha^{k'_{a1} k_{b2}} \bmod p$ with B. The system is insecure.

4. Our improved scheme

We first introduce our modified scheme, and then examine the security and the key utilization.

The improved scheme

A selects two secret random numbers secrets k_{a_1} and k_{a_2}, and then computes their corresponding publics $r_{a_1} = \alpha^{k_{a_1}} \bmod p$ and $r_{a_2} = \alpha^{k_{a_2}} \bmod p$. Then A has his signature generation equation as $s_a \oplus K_{AB} = Time_a \cdot x_a - (r_{a_1} \oplus r_{a_2}) \cdot (k_{a_1} + k_{a_2}) \bmod p-1$, where $K_{AB} = \alpha^{x_a x_b} \bmod p$ is the long-term secret key between A and B. A sends

($r_{a_1}, r_{a_2}, s_a, Time_a, cert(y_a)$) to B. B will verify the message by checking whether $y_a^{Time_a} = (r_{a_1} r_{a_2})^{r_{a_1} \oplus r_{a_2}} \cdot \alpha^{s_a \oplus K_{AB}} \bmod p$.

The security Analysis

The security of our improved scheme is based on the discrete logarithm problem, and this improved scheme is resistant to the forgery attacks and the replay attack, which fail the previous versions [7, 9, 14]. Its resistance to the replay attack can be easily assured by checking the timestamp. Its resistance to the forgery attack can be analyzed as follows.

Given the values (r_{a_1}, r_{a_2}, s_a), it is impossible for an attacker to derive the corresponding $Time_a$ because it is a discrete logarithm problem and he does not know the K_{AB}. Given (r_{a_1}, r_{a_2}, $Time_a$), it is also impossible to derive the corresponding s_a because it is a discrete logarithm problem and the attacker does not know K_{AB}. The same argument still holds when the attacker try to derive r_{a_1} or r_{a_2}, given the rest of the parameters.

Now we examine the forgery attack in which the attacker make up the message from the eavesdropped ones. Given a valid message (r_{a_1}, r_{a_2}, s_a, $Time_a$), the attacker may find r'_{a_1} and r'_{a_2} such that $(r'_{a_1} r'_{a_2}) = (r_{a_1} r_{a_2}) \bmod p$, and then try to derive the corresponding $Time_{a'}$ and $s_{a'}$ to satisfy the verification equation. Then he has to solve the equations $Time_{a'} = Time_a \cdot (r'_{a_1} \oplus r'_{a_2}) \bmod p-1$ and $s_{a'} \oplus K_{AB} = -(s_a \oplus K_{AB}) \cdot (r'_{a_1} \oplus r'_{a_2}) \bmod p-1$. Since the attacker does not know K_{AB}, he has no way to derive the $s_{a'}$. The same result holds when the attacker tries other approaches to make up new message from the eavesdropped ones. So, our scheme is secure against the forgery attack and the replay attack.

The perfect forward secrecy and key utilization

Now we discuss the perfect forward secrecy and the key utilization of our improved scheme. Harn-Lin's scheme only uses three of the four common keys to preserve the perfect forward secrecy. We can easily check this by examining the signature equations $s_a = x_a - (r_a) \cdot (k_{a_1} + k_{a_2}) \bmod p-1$ and $s_b = x_b - (r_b) \cdot (k_{b_1} + k_{b_2}) \bmod p-1$, where $r_a = \alpha^{r_{a_1} r_{a_2}} \bmod p$ and $r_b = \alpha^{r_{b_1} r_{b_2}} \bmod p$. Then, we can compute $x_a x_b = r_a r_b (k_{a_1} k_{b_1} + k_{a_1} k_{b_2} + k_{a_2} k_{b_1} + k_{a_2} k_{b_2}) + s_a r_b (k_{b_1} + k_{b_2}) + r_a s_b (k_{a_1} + k_{a_2}) + s_a s_b \bmod p-1$ and $K_{AB} = \alpha^{x_a x_b} = (K_1 K_2 K_3 K_4)^{r_a r_b} (r_{b_1} r_{b_2})^{s_a r_b} (r_{a_1} r_{a_2})^{r_a s_b} \alpha^{s_a s_b} \bmod p$. From the equation, we can see that an adversary can derive the log-term secret key K_{AB} between A and B if he knows four consecutive session keys $K_1, K_2, K_3,$ and K_4. Therefore, Harn-Lin's scheme only uses three of the four common session keys.

Next we examine the key utilization of our scheme. According to the signature generation equation, we have $Time_a \cdot Time_b \cdot x_a \cdot x_b = (s_a \oplus K_{AB}) \cdot (s_b \oplus K_{AB}) + (s_a \oplus K_{AB}) \cdot (r_{b_1} \oplus r_{b_2}) \cdot (k_{b_1} + k_{b_2}) + (s_b \oplus K_{AB}) \cdot (r_{a_1} \oplus r_{a_2}) \cdot (k_{a_1} + k_{a_2}) + (r_{a_1} \oplus r_{a_2})(r_{b_1} \oplus r_{b_2})(k_{a_1} + k_{a_2})(k_{b_1} + k_{b_2}) \bmod p-1$, and derive $K_{AB}^{Time_a Time_b} = \alpha^{(s_a \oplus K_{AB})(s_b \oplus K_{AB})} \cdot (r_{b_1} r_{b_2})^{(s_a \oplus K_{AB})(r_{b_1} \oplus r_{b_2})} \cdot (r_{a_1} r_{a_2})^{(s_b \oplus K_{AB})(r_{a_1} \oplus r_{a_2})} \cdot (K_1 K_2 K_3 K_4)^{(r_{a_1} \oplus r_{a_2})(r_{b_1} \oplus r_{b_2})} \bmod p$. From the equation, we see that the adversary cannot derive the long-term secret key even he gets the session key K_1, $K_2, K_3,$ and K_4. Therefore, our scheme can use all the four session keys and achieve better key utilization.

5. Conclusions

In this article, we have proposed a secure multiple-keys agreement protocol. This protocol does not employ the conventional one-way functions, and allows two entities to share multiple keys in a two-pass interaction. Compared with Harn-Lin's scheme and the

previous modified versions, our scheme not only preserves the original requirement but also withstand the forgery attack and the replay attack. Further, the improved scheme achieve better key utilization.

References

[1] W. Diffe, M.E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory* 22 (6), pp. 644-654, 1976.

[2] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inform. Theory* 33 (2), pp. 469-472, 1985.

[3] K. Nyberg, R.A. Rueppel, "Message recovery for signature scheme based on the discrete logarithm problem," in: *Advances in Cryptology-Eurocrypt' 94*, pp. 175-190, 1994.

[4] ' The digital signature standard by NIST' , *Comm. ACM 35 (7)*, pp. 36-40, 1992.

[5] A. Arazi, "Integrating a key cryptosystem into the digital signature standard," *Electronics Letters 29 (11)*, pp. 966-967, 1993.

[6] K. Nyberg, R.A. Rueppel, "Weakness in some recent key agreement protocols," *Electronics Letters 30 (1)*, pp. 26-27, 1994.

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

[8] H. Dobbertin, "The status of MD5 after a recent attack," *CryptoBytes 2 (2)*, pp. 1-6, 1996.

[9] S.M. Yen, and M. Joye, "Improved authenticated multiple-key agreement protocol," *Electron. Lett. 34 (18)*, pp. 1738-1739, 1998.

[10] T.S. Wu, W.H. He, and C.L. Hsu, "Security of authenticated multiple-key agreement protocols," *Electron. Lett.* 35 (5), pp. 391-392, 1999.

[11] C.H. Lim, and P.J. Lee, "Security of interactive DSA batch verification," *Electron. Lett.* 30 (19), pp. 1592-1593, 1994.

[12] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[13] Ateniese, G., Steiner, M., and Tsudik, G., "Authenticated group key agreement and friends," *Proceedings of the 5th ACM conference on Computer and communications security*, pp. 17 – 26, 1998.

[14] H.T. Yen, H.M. Sun, and T. Hwang, "Improved authenticated multiple-key agreement protocol," in: *Proc. 11th National Conf. Information Security*, TaiNan, Taiwan, pp. 229-231, May 2001.