Enhanced Security for the Modified Authenticated Key Agreement Scheme

***Minho Kim*[†] *and Çetin Kaya Koç*[††]**

[†]*Information Security Laboratory, School of EECS, Oregon State University, Corvallis, Oregon 97331, USA*
[††]*Information Security Research Center, Istanbul Commerce University, Eminönü, Istanbul 34112, TURKEY*

Summary
Hsu et al. [3] showed that the Ku-Wang [4] modified authentication key agreement scheme is vulnerable to the modification attack and further proposed an improvement of the Ku-Wang scheme. Recently, Lee-Lee [5] showed that the Hsu et al.'s scheme is vulnerable to the modification attack and proposed another improvement on the modified authenticated key agreement scheme. However, we will show that the Hsu et al.'s scheme suffers from the off-line guessing attack and the Lee-Lee's scheme is still vulnerable to off-line guessing, man-in-the-middle, and reflection attacks. We then propose an enhanced secure scheme to eliminate these security flaws.

Key words:
Key agreement, modification attack, off-line guessing attack, man-in-the-middle attack, reflection attack

1. Introduction

The Diffie-Hellman key agreement protocol was developed by Diffie and Hellman in 1976 and published in the ground-breaking paper ``New Directions in Cryptography [2]." The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets. However, it is vulnerable to the man-in-the-middle attack, because it does not authenticate the participants. In 1994, Anderson-Lomas showed how collision-rich hash functions can be used to detect those attacks while they are in progress [1]. Later, Seo-Sweeney proposed an efficient simple key agreement protocol that is based on a pre-shared password method, and modifies the Diffie-Hellman scheme to provide user authentication [6]. In the Seo-Sweeney protocol, two parties that have shared a common password can establish a session key by exchanging two messages. This protocol is more efficient than the Anderson-Lomas scheme in terms of computational time and exchanged messages. In 2000, Tseng [7] pointed out that the key validation of the Seo-Sweeney scheme cannot resist the replay attack. The adversary can successfully convince an honest party of a wrong session key. Tseng proposed an improved scheme to overcome this weakness. However, Ku-Wang showed that Tseng's modified authenticated key agreement protocol is vulnerable to the modification attack and the backward replay attack without modification [4]. They then proposed an improved scheme to strengthen the protocol. In 2003, Hsu et al. showed that the Ku-Wang modified authentication key agreement scheme is vulnerable to the modification attack, and further proposed an improvement of the Ku-Wang scheme in [3]. Recently, Lee-Lee showed that the Hsu et al.'s scheme is also vulnerable to the modification attack. An attacker can alter the transmitted messages to deceive the communicating parties into believing a wrong session key [5]. They then proposed another improvement on the modified authenticated key agreement scheme. The purpose of this paper is to first show that the Hsu et al.'s scheme suffers from the off-line guessing attack, and the Lee-Lee's scheme is still vulnerable to the off-line guessing attack, the man-in-the-middle attack, and the reflection attack. We then propose an improved security scheme to overcome these security defects.

The remainder of this paper is organized as follows. We first review the Hsu et al.'s and the Lee-Lee's Schemes in Section 2. We then describe our attack in Section 3. In Section 4, we propose an enhanced secure scheme. In Section 5, we make an analysis of the security of our scheme. Finally, we shall give a brief conclusion in Section 6.

2. Review of the Hsu et al.'s and the Lee-Lee's Schemes

We introduce the notation to describe both phases and explain the detailed steps in each phase.

2.1 Notations

- ***A***, ***B***, and ***E*** denote the two communicating users and the adversary.
- $\boldsymbol{ID_A}$ and $\boldsymbol{ID_B}$ denote the identities of user *A* and *B*.

Manuscript received March 5, 2006.
Manuscript revised July 25, 2006.

- ***a*** and ***b*** denote the random number chosen by *A* and *B*.
- ***n*** denotes a large prime number.
- ***g*** denotes a generator with the order *n*-1 in GF(*n*).
- $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ denote the session key of *A* and *B*.
- ***P*** denotes the common password shared between *A* and *B*.
- ***Q*** and $\boldsymbol{Q^{-1}}$ denote an integer computed from *P* and the inverse of *Q*(*mod n*).
- ⊕ denotes the bitwise XOR operation.
- **h(·)** denotes a one-way and collision-resistant hash function and $\boldsymbol{h(m_1, m_2)}$ means the hash of the concatenation of the message $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$.
- The expression ***A* (→)⇒ *B*: *X*** means *A* sends the message *X* to *B* via an (in)secure channel.

2.2 Review of the Hsu et al.'s scheme

There are two phases in this scheme: the key establishment phase and the key validation phase.

2.2.1 Key Establishment Phase

E1. *A* → *B*: X_1.

A computes $X_1 = g^{aQ} \bmod n$ and sends X_1 to *B*.

E2. *B* → *A*: Y_1.

B computes $Y_1 = g^{bQ} \bmod n$ and sends Y_1 to *A*.

E3. *A* computes the session key

$$K_1 = Y^a \bmod n = g^{ab} \bmod n,$$

where $Y = Y_1^{Q^{-1}} \bmod n = g^b \bmod n$.

E4. *B* computes the session key

$$K_2 = X^b \bmod n = g^{ab} \bmod n,$$

where $X = X_1^{Q^{-1}} \bmod n = g^a \bmod n$.

After Step E4, *A* and *B* obtain the same session key $K_1 = K_2 = g^{ab} \bmod n$.

2.2.2 Key Validation Phase of the Hsu et al.'s scheme

HV1. *A* → *B*: X_2.

A computes $X_2 = h(ID_A, K_1)$ and sends X_2 to *B*.

HV2. *B* verifies the validity of X_2 with $h(ID_A, K_2)$.

HV3. *B* → *A*: Y_2.

If the value is correct,

B computes $Y_2 = h(ID_B, K_2)$ and sends Y_2 to *A*.

HV4. *A* verifies the validity of Y_2 with $h(ID_B, K_1)$.

After Step HV4, *A* and *B* share the common session key $K_1 = K_2 = g^{ab} \bmod n$.

2.3 Review of the Lee-Lee's scheme

The modified authenticated key agreement scheme described in [5] consists of two phases: the key establishment phase and the key validation phase. The key establishment phase is the same as the key establishment phase in the Hsu et al.'s scheme.

2.3.1 Key Validation Phase of the Lee-Lee's scheme

V1. *A* → *B*: X_2.

A computes $X_2 = h(ID_A, X_1, K_1)$ and sends X_2 to *B*.

V2. *B* verifies the validity of X_2 with $h(ID_A, X_1, K_2)$.

V3. *B* → *A*: Y_2.

If the value is correct,

B computes $Y_2 = h(ID_B, Y_1, K_2)$ and sends Y_2 to *A*.

V4. *A* verifies the validity of Y_2 with $h(ID_B, Y_1, K_1)$.

After Step V4, *A* and *B* shared the common session key $K_1 = K_2 = g^{ab} \bmod n$.

3. Our Attack

First, we show an attack against [3] with the off-line guessing attack. Later, we show three possible attacks against [5]; the off-line guessing attack, the man-in-the-middle attack, and the reflection attack.

To attack these schemes, we suppose that the adversary *E* would eavesdrop and interpose the communication between *A* and *B*.

3.1 Off-line Guessing Attack on the Hsu et al.'s scheme

GAH1. *E* → *A*: Y_E.

E monitors and intercepts the message $X_1 = g^{aQ} \bmod n$ in Step E1. He randomly selects integer *Z*, and then computes his own value, $Y_E = g^Z \bmod n$. Next, he sends Y_E to *A*.

GAH2. *A* computes the session key

$K_1^* = Y^{*a} \bmod n = g^{ZaQ^{-1}} \bmod n$,

where $Y^* = Y_E^{Q^{-1}} \bmod n$.

GAH3. A → *B*: X_2^*.

To verify the validity of the session key K_1^*,

A computes $X_2^* = h(ID_A, K_1^*)$ and sends X_2^* to *B* in Step HV1.

GAH4. *E* intercepts X_2^* and tries to find the suitable value.

First, he computes Q_E and Q_E^{-1} from the guessing password P_E. Second, he computes his own value $X_E^* = h(ID_A, X_1^{Z(Q_E^{-1})^2})$. Next, he compares $X_2^* = h(ID_A, g^{ZaQ^{-1}} \bmod n)$ with X_E^*.

If they match, this guessing attack is succeeded. Otherwise, *E* tries to find the password again in this Step GAH4. Therefore, this off-line guessing attack can succeed when *E* impersonates *B*.

3.2 Off-line Guessing Attack on the Lee-Lee's scheme

GA1. *E* → *B*: X_E.

E monitors and intercepts the message $X_1 = g^{aQ} \bmod n$ in Step E1. He then computes his own value $X_E = g \bmod n$, and sends X_E to *B*.

GA2. *E* → *A*: Y_E.

E intercepts $Y_1 = g^{bQ} \bmod n$ in Step E2. He then replaces Y_1 with $Y_E = g \bmod n$, and sends Y_E to *A*.

GA3. *A* computes the session key $K_1^* = g^{aQ^{-1}} \bmod n$,

where $Y^* = Y_E^{Q^{-1}} \bmod n = g^{Q^{-1}} \bmod n$, and

$K_1^* = Y^{*a} \bmod n = g^{aQ^{-1}} \bmod n$.

GA4. *B* computes the session key $K_2^* = g^{bQ^{-1}} \bmod n$,

where $X^* = X_E^{Q^{-1}} \bmod n = g^{Q^{-1}} \bmod n$, and

$K_2^* = X^{*b} \bmod n = g^{bQ^{-1}} \bmod n$.

GA5. *A* → *B*: X_2^*.

To verify the validity of the session key K_1^*,

A computes $X_2^* = h(ID_A, X_1, K_1^*)$

$= h(ID_A, g^{aQ} \bmod n, g^{aQ^{-1}} \bmod n)$, and sends X_2^* to *B* in Step V1.

GA6. *E* intercepts X_2^* and tries to find the suitable value.

E computes Q_E and Q_E^{-1} from the guessing password P_E. He then computes his own value $X_E^* = h(ID_A, X_1, X_1^{(Q_E^{-1})^2})$. Next, he compares $X_2^* = h(ID_A, X_1, K_1^*)$

$= h(ID_A, g^{aQ} \bmod n, g^{aQ^{-1}} \bmod n)$ with X_E^*.

If they match, this guessing attack is successful. Otherwise, *E* tries to find the password again in Step GA6.

Thus, *E* can guess the password in this attack. Furthermore, if this attack is the on-line guessing attack, it might be successful. However, it will be detected by the system or the users, since *E* has to access several times more than the limited access time, if his attack fails. Therefore, this off-line guessing attack can be successful like the attack on the Hsu et al.'s scheme.

3.3 Man-In-The-Middle Attack

MA1. *A* → *B*: X_1 ⇒ *E* → *B*: X_{E1}.

A computes $X_1 = g^{aQ} \bmod n$ and sends X_1 to *B*. However, *E* monitors and intercepts the message X_1, and then replaces X_1 with his own value $X_{E1} = 1 \bmod n$. Next, he sends X_{E1} to *B*.

MA2. *B* → *A*: Y_1 ⇒ *E* → *A*: Y_{E1}.

B computes $Y_1 = g^{bQ} \bmod n$ and sends Y_1 to *A*. However, *E* also intercepts the message Y_1, and then replaces Y_1 with his own value $Y_{E1} = 1 \bmod n$. He then sends Y_{E1} to *A*. Next, *E* calculates $X_{E2} = h(ID_A, X_{E1}, K_1^*)$ and $Y_{E2} = h(ID_B, Y_{E1}, K_2^*)$.

MA3. For the session key,

A computes $Y^* = Y_{E1}^{Q^{-1}} \bmod n = 1 \bmod n$ and $K_1^* = Y^{*a} \bmod n = 1 \bmod n$.

MA4. By the same way,

B computes $X^* = X_E^{Q^{-1}} \bmod n = 1 \bmod n$, and $K_2^* = X^{*b} \bmod n = 1 \bmod n$.

Finally, *A* and *B* obtain the same session key $K_1^* = K_2^* = 1 \bmod n$. After Step MA4, user *A* and *B* start to verify their session key K_1^* and K_2^*.

MA5. *A* →*B*: X_2^* ⇒ *E* →*B*: X_{E2}.

A computes $X_2^* = h(ID_A, X_1, K_1^*)$ and sends it to *B*. *E* intercepts X_2^*, and then easily substitutes X_2^* with X_{E2}, since X_{E2} and Y_{E2} are already calculated after Step MA2. Next, he sends X_{E2} to *B*.

MA6. After received it, *B* verifies the validity of $X_{E2} = h(ID_A, X_{E1}, K_1^*)$ with $h(ID_A, X_{E1}, K_2^*)$.

MA7. *B* →*A*: Y_2^* ⇒ *E* →*A*: Y_{E2}.

Similarly, *B* computes $Y_2^* = h(ID_B, X_2, K_2^*)$ and sends Y_2^* to *A*. However, *E* substitutes Y_2^* with Y_{E2}, and sends Y_{E2} to *A*.

MA8. After *A* received it, he verifies the validity of $Y_{E2} = h(ID_B, Y_{E1}, K_2^*)$ with $h(ID_B, Y_{E1}, K_1^*)$.

Since the session key $K_1^* = K_2^* = 1 \bmod n$, *A* and *B* get the common session key and convince it without doubt. Thus, *E* can attack Lee-Lee's scheme, if he can eavesdrop certain users' communications.

3.4 Reflection Attack

RA1. *A* →*E*: X_1.

A computes $X_1 = g^{aQ} \bmod n$ and sends X_1 to *B*.

RA2. *E* →*A*: X_E.

E intercepts the message X_1 in Step E1. He then computes his own value $X_E = g \bmod n$, and sends *A* the X_E instead of Y_1.

RA3. *A* →*E*: X_2^*.

A computes the session key $K_1^* = g^{aQ^{-1}} \bmod n$, where $Y^* = X_E^{Q^{-1}} \bmod n = g^{Q^{-1}} \bmod n$ and

$$K_1^* = Y^{*a} \bmod n = g^{aQ^{-1}} \bmod n.$$

To verify the validity of the session key K_1^*, *A* computes $X_2^* = h(ID_A, X_1, K_1^*)$ $= h(ID_A, g^{aQ} \bmod n, g^{aQ^{-1}} \bmod n)$, and sends X_2^* to *B* in Step V1. However, *E* intercepts X_2^*, and ferrets out the proper value as we attacked in Step GA6. Finally, he can compute Q_E and Q_E^{-1} from the guessing password P_E. As we have shown in this attack, it is obvious that all users have communicated with malicious adversary *E*, not with the trusted parties. Thus, *E* is able to put up an illusion to deceive other users.

4 Our Enhanced Secure Scheme

To resist the above weaknesses, we propose an enhanced secure scheme taking into account off-line guessing, the man-in-the-middle, and the reflection attacks of [5].

4.1 Enhanced Key Establishment Phase

EE1. *A* →*B*: X_1.

A computes $X_1 = g^{aQ} \oplus Q$ and sends X_1 to *B*.

EE2. *B* →*A*: Y_1.

B computes $Y_1 = g^{bQ} \oplus Q$ and sends Y_1 to *A*.

EE3. *A* computes the session key $K_1 = g^{abQ} \bmod n$, where $Y = Y_1 \oplus Q = (g^{bQ} \oplus Q) \oplus Q = g^{bQ}$ and $K_1 = Y^a \bmod n$.

EE4. *B* computes the session key $K_2 = g^{abQ} \bmod n$, where $X = X_1 \oplus Q = (g^{aQ} \oplus Q) \oplus Q = g^{aQ}$ and $K_2 = X^b \bmod n$.

After Step EE4, *A* and *B* obtain the same session key $K_1 = K_2 = g^{abQ} \bmod n$.

4.2 Enhanced Key Validation Phase

In this phase, to fend off the man-in-the-middle attack, *A* and *B* check the value of $K_1 \neq 1$ and $K_2 \neq 1$, respectively. If the values are correct, then they can start this phase.

EV1. *A* →*B*: X_2.

A computes $X_3 = K_1^{-a} \bmod n = g^{bQ} \bmod n$ and $X_2 = h(ID_A, Y_1, K_1) \oplus h(X_3)$, and then sends X_2 to *B*.

EV2. *B* →*A*: Y_2 .

B computes $X_3' = g^{bQ} \bmod n$, and verifies the validity of X_2 with $h(ID_A, Y_1, K_2) \oplus h(X_3')$. If they are correct, *B* computes $Y_3 = K_2^{-b} \bmod n$ $= g^{aQ} \bmod n$ and $Y_2 = h(ID_B, X_1, K_2) \oplus h(Y_3)$, and then sends Y_2 to *A*.

EV3. *A* computes $Y_3' = g^{aQ} \bmod n$, and verifies the validity of Y_2 with $h(ID_B, X_1, K_1) \oplus h(Y_3')$. If they are correct, *A* and *B* are able to share the common session key $K_1 = K_2 = g^{abQ} \bmod n$. If they are not, then they discard the session key.

5 Security Analysis

5.1 Off-line Guessing Attack

If *E* eavesdrops and intercepts the information for the off-line guessing attack, he can obtain information such as $X_1 = g^{aQ} \oplus Q$ and $Y_1 = g^{bQ} \oplus Q$. After that, *E* replaces X_1 and Y_1 with $X_E = g^{a'Q'} \oplus Q'$ and $Y_E = g^{b'Q'} \oplus Q'$, respectively. He then sends them to *A* and *B* in Step GA3 and GA4, respectively. Next, *A* and *B* compute $K_1^*, K_2^*, X_2^*,$ and Y_2^*,

where $K_1^* = Y^{*a} \bmod n = (g^{b'Q'} \oplus Q' \oplus Q)^a \bmod n$

from $Y^* = Y_E \oplus Q = (g^{b'Q'} \oplus Q') \oplus Q$,

$K_2^* = X^{*b} \bmod n = (g^{a'Q'} \oplus Q' \oplus Q)^b \bmod n$

from $X^* = X_E \oplus Q = (g^{a'Q'} \oplus Q') \oplus Q,$

$X_2^* = h(ID_A, Y_E, K_1^*) \oplus h(X_3^*)$

from $X_3^* = g^{b''Q''} \bmod n,$ and

$Y_2^* = h(ID_B, X_E, K_2^*) \oplus h(Y_3^*)$

from $Y_3^* = g^{a''Q''} \bmod n$.

They then send X_2^* and Y_2^* to each other for verification. *E* intercepts this information, and tries to find the appropriate values by the guessing attack. However, *E* cannot obtain the result. It is very difficult to solve the equation $K_1^* = g^{abQ} \bmod n$ without knowing a, b, and Q, because it meets the Diffie-Hellman problem [2]. In addition, *A* and *B* can detect the modified value X_E and Y_E from the key validation phase. As an example, *A* sends X_2^* to *B*, then *B* computes $X_3' = g^{bQ} \bmod n$ and verifies the validity of X_2^* with $h(ID_A, Y_1, K_2^*) \oplus h(X_3')$. *B* finds out that they are not matched, because of $Y_E \neq Y_1$ and $h(X_3^*) \neq h(X_3')$, even though K_1^* and K_2^* are the same. Therefore, our proposed scheme is secure against the off-line guessing attack.

5.2 Man-In-The-Middle Attack

If *E* wants to modify the value by using the man-in-the-middle attack, he can obtain and replace the information, such as X_1 to X_E , and Y_1 to Y_E . However, *E's* attack still ends in failure. With the reason aforementioned, *B* makes out that they are not equal through the key validation phase, because Y_E is not equal to his own value Y_1 , and $h(X_3^*)$ is not equal to $h(X_3')$. Furthermore, they already checked the values of $K_1 \neq 1$ and $K_2 \neq 1$ as a measure against as our man-in-the-middle attack. Thus, our proposed scheme withstands the man-in-the-middle attack.

5.3 Reflection Attack

Whenever *E* tries to attack with the reflection attack, he can intercept and replace the information such as $X_1 = g^{aQ} \oplus Q$ with $X_E = g^{a'Q'} \oplus Q'$, in Step EE1. After that, *A* computes the session key

$K_1^* = Y^{*a} \bmod n = (g^{b'Q'} \oplus Q' \oplus Q)^a \bmod n$

from $Y^* = Y_E \oplus Q = (g^{b'Q'} \oplus Q') \oplus Q$, and

$X_2^* = h(ID_A, Y_E, K_1^*) \oplus h(X_3^*)$

from $X_3^* = g^{b''Q''} \bmod n$.

He then sends X_2^* to *B* for validation in Step EV1. *E* intercepts X_2^*, and tries to seek the suitable value, but it is hard to find a, b, and Q from the value X_2^* $= h(ID_A, Y_E, K_1^*) \oplus h(X_3^*)$ by using the information that *E* has, such as $X_1 = g^{aQ} \oplus Q$. Moreover, after Step EV1, *E* computes $Y_2^* = h(ID_B, X_E, K_2^*) \oplus h(Y_3^*)$ from

$Y_3^* = g^{a"Q"} \bmod n$, and sends Y_2^* to *A*. *A* can detect that this phase is modified. If *E* sends Y_2^*, then *A* computes $Y_3' = g^{aQ} \bmod n$, and verifies the validity of Y_2^* with $h(ID_B, X_1, K_1^*) \oplus h(Y_3')$.

Finally, *A* finds out that they are not matched, because of $X_E \neq X_1$ and $h(Y_3^*) \neq h(Y_3')$. In addition, if *E* does not respond to *A* after Step EV1, then *A* recognizes that this key agreement protocol is modified or realizes that something is wrong. He then discards this session key. Consequently, our proposed scheme can resist the reflection attack.

6 Conclusions

In this paper, we have shown that the Hsu et al.'s scheme suffers from the off-line guessing attack, and the Lee-Lee's scheme is still vulnerable to the off-line guessing attack, the man-in-the-middle attack, and the reflection attack. We then propose an enhanced secure scheme to overrule those security flaws.

References

[1] R. J. Anderson and T. M. A. Lomas, “Fortifying key negotiation schemes with poorly chosen passwords,” Electronics Letters, vol. 30, no. 13, pp. 1040-1041, 1994.

[2] W. Diffie and M. E. Hellman, “New directions in cryptography,” IEEE Transaction on Information Theory, vol. IT-22, no. 6, pp. 644-654, Nov 1976.

[3] C. L. Hsu, T. S. Wu, T. C. Wu, and C. Mitchell, “Improvement of modified authenticated key agreement protocol,” Applied Mathematics and Computation, vol. 142, no. 2-3, pp. 305-308, Nov 2003.

[4] W. C. Ku and S. D. Wang, “Cryptanalysis of modified authenticated key agreement protocol,” Electronics Letters, vol. 36, no. 21, pp. 1770-1771, 2000.

[5] N. Y. Lee and M. F. Lee, “Further improvement on the modified authenticated key agreement scheme,” Applied Mathematics and Computation, vol. 157, no. 3, pp. 729-733, Oct 2004.

[6] D. H. Seo and P. Sweeney, “Simple authenticated key agreement algorithm,” Electronics Letters, vol. 35, no. 13, pp. 1073-1074, 1999.

[7] Y. M. Tseng, “Weakness in simple authenticated key agreement protocol,” Electronics Letters, vol. 36, no. 1, pp. 48-49, 2000.

Minho Kim is s a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at Oregon State University. He has also worked as an assistant professor of Computer Science at Korea Air Force Academy. His research interests are in cryptographic algorithms and protocols for cryptography, computer arithmetic, computer and network security, and wireless communications.

Çetin Kaya Koç received his Ph.D. degree from University of California, Santa Barbara. Dr. Koç's research interests are in cryptographic engineering, algorithms and architectures for cryptography, computer arithmetic and finite fields, parallel algebraic computation, and network security. He has co-founded the Workshop on Cryptographic Hardware and Embedded Systems (CHES), and has been an Associate Editor of IEEE Transactions on Computers and IEEE Transactions on Mobile Computing. Dr. Koç has also been working as a consulting engineer with research and development interests in cryptographic engineering and embedded systems for several companies including Intel, RSA Security, and Samsung Electronics. Dr. Koç is currently on leave from Oregon State University, working at Information Security Research Center of Istanbul Commerce University in Istanbul, Turkey.