In this paper, we propose a robust authenticated key agreement protocol in which two communication parties exchange a secret session key and authenticate each other. The protocol can be considered as an improvement of simple authenticated key agreement algorithm (SAKA). However, our protocol is more significant in that it addresses the problem of the off-line password guessing attack.

Keywords: key agreement, password guessing attack

1. INTRODUCTION

In 1976, Diffie and Hellman[1] proposed the well-known public key distribution scheme based on the discrete logarithm problem (DLP) to enable two authenticated parties to agree a common secret session key for use with conventional symmetric encryption algorithm. However, the original scheme is vulnerable to the man-in-the-middle attack since it does not provide authentication of the secret session key. To integrate authentication into the key agreement scheme, several methods have been proposed. One method is that two parties use certificates (e. g. digital signature) from a trusted authority. The main problem of this method is that if the trusted authority is compromised then the total system would be in danger. Another method is that two parties use not certificates from the trusted authority but pre-shared a common secret password each other [2-3]. In this method, an adversary cannot impersonate as two parties without knowing a secret password.

The researchers have studied many authenticated key agreement protocol based on password up to now [4-8]. In 1999, Seo and Sweeny [4] proposed the simple authenticated key agreement algorithm(SAKA), modified Diffie-Hellman key exchange protocol to provide user authentication, based on the DLP using a pre-shared password. It prevents the man-in-the-middle attack and the amount of traffic is the same as the Diffie-Hellman scheme with only two packets required to agree on the secret session key. Later, Lin et al. [6] showed a weakness of the Seo and Sweeny’s scheme and proposed the improved scheme. The improved scheme can be resistant to the man-in-the-middle attack and the dictionary attack. Recently, Hsieh et al. [8] showed that the security enhancement for the SAKA scheme proposed by Lin et al. is still vulnerable to the off-line password guessing attack. In this paper, we propose a robust simple authenticated key agreement protocol in which two communication parties exchange a secret session key and authenticate each other. The protocol can be considered as an improvement of the SAKA. However, our protocol is more significant in that it addresses the problem of the off-line password guessing attack.

Our paper is structured as follows: In the next Section, we will review the SAKA scheme and Lin et al.’s protocol which points out a weakness of the SAKA scheme and mention cryptanalysis of Lin et al.’s protocol. In Section 3, we will propose a robust authenticated key agreement protocol. Finally, the conclusion of this paper will be derived in Section 4.

This research was supported by University IT Research Center Project
2. PREVIOUS WORK

2.1 The simple authentication key agreement algorithm

We assume that Alice and Bob share a common password PWD before the protocol begins. The simple authentication key agreement algorithm (SAKA) uses the same public values \( p \) and \( g \) as the original Diffie-Hellman one, where \( p \) is a large prime and \( g \) is a primitive root with order \( p - 1 \) over \( \text{GF}(p) \). Fig. 1 shows the SAKA scheme.

![Fig. 1. The SAKA scheme](image)

**Key Establishment Phase:**

- e.1) Alice and Bob each may obtain two integers \( Q \) and \( Q^{-1} \) mod \((p-1)\) from a common password PWD, where \( Q \) could be computed in any predetermined way from PWD.
- e.2) Alice selects a random integer \( a \). Then Alice computes \( X_i = g^{aq} \mod p \) and sends it to Bob.
- e.3) Bob selects a random integer \( b \). Then Bob computes \( Y_i = X_i^{bQ^{-1}} \mod p \) and sends it to Alice.
- e.4) Alice computes \( Y = Y_i^{Q^{-1}} \mod p \) and \( \text{key}_1 = Y^a \mod p \).
- e.5) Bob computes \( X = X_i^{Q^{-1}} \mod p \) and \( \text{key}_2 = X^b \mod p \).

Consequently, Alice and Bob share \( \text{key}_1 = \text{key}_2 = g^{ab} \mod p \) as a common secret session key.

**Key Validation Phase:**

- v.1) Alice computes \((\text{key}_1)^0 \mod p\) and sends it to Bob.
- v.2) Bob computes \((\text{key}_2)^0 \mod p\) and sends it to Alice.
- v.3) Alice and Bob check whether \((\text{key}_2)^{0Q^{-1}} = \text{key}_1 \mod p\) and \((\text{key}_1)^{0Q^{-1}} = \text{key}_2 \mod p\) holds or not.

2.2 Lin et al.’s protocol

Lin et al. showed that the SAKA scheme has three weaknesses as follows [6]:

1) It cannot confirm identity of the user;
2) It cannot withstand the dictionary attack;
3) It does not provide perfect forward secrecy;

To overcome the above weaknesses, an enhanced scheme is proposed. Fig. 2 shows an enhanced SAKA scheme.

![Fig. 2. Lin et al.’s protocol](image)

**Key establishment phase:**

The system parameters and key establishment phase of Lin et al.’s scheme are the same as the original SAKA one.

**Key validation phase:**

- v.11) Alice computes \( X' = Y^{Q^{-1}} \mod p \) and sends it to Bob.
- v.22) Bob computes \( Y' = X^{Q^{-1}} \mod p \) and sends it to Alice.
- v.33) Alice and Bob check whether \( Y' = (g^a)^{Q^{-1}} \mod p \) and \( X' = (g^b)^{Q^{-1}} \mod p \) holds or not, respectively.
2.3 Cryptanalysis of Lin et al.’s protocol

Recently, Hsieh et al. showed that Lin et al.’s protocol, an enhancement of the SAKA scheme, is insecure against the off-line password guessing attack. The process of the off-line password guessing attack as follows: An adversary first guesses a password PWD and computes a corresponding \( Q \) and \( Q^{-1} \) mod \( p \) from PWD. He/she intercepts messages \( X_1 \) and \( Y' \) in e.2) and v.22).

By checking whether \( (X_1)Q^i \mod X_1 = (Y')Q \mod p \) holds or not, he/she can verify the correctness of the guessed password. Because \( (X_1)Q^i \mod X_1 = g^a \mod p \) holds, he/she guesses the password successfully.

3. A ROBUST AUTHENTICATED KEY AGREEMENT PROTOCOL

3.1 The enhanced protocol

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{key establishment phase}&gt;)</td>
<td>(&lt;\text{key validation phase}&gt;)</td>
</tr>
<tr>
<td>(X_1 = g^{Q} \mod p)</td>
<td>(X_i = g^{Q} \mod p)</td>
</tr>
<tr>
<td>(Y = X_1^{-1} \mod p)</td>
<td>(Y_i = X_1^{-1} \mod p)</td>
</tr>
<tr>
<td>(\text{key}_1 = Y^a \mod p)</td>
<td>(\text{key}_2 = X^a \mod p)</td>
</tr>
<tr>
<td>(K_1 = Y^a \mod p)</td>
<td>(K_2 = X^a \mod p)</td>
</tr>
</tbody>
</table>

Fig. 3. The enhanced protocol

In the previous schemes [4-7], each scheme is vulnerable to the replay attack or the off-line password guessing attack. In the following, we will propose a robust authenticated key agreement protocol against these attacks. The system parameters are the same as the original Diffie-Hellman one. We assume that two authenticated parties, Alice and Bob, share a common secret password PWD before the protocol begins. Fig. 3 shows the enhanced protocol of the SAKA scheme.

**Key establishment phase:**
The system parameters and key establishment phase of the enhanced protocol are the same as the original SAKA except that Alice and Bob compute \( K_1 \) and \( K_2 \).

**Key validation phase:**
Step 1. Alice computes \( X' = Y^{K_1} \mod p \) and sends it to Bob.
Step 2. Bob computes \( Y' = g^{K_1} \mod p \) and sends it to Alice.
Step 3. Alice and Bob check whether \( Y' = g^{K_1} \mod p \) and \( X' = (Y')^{K_1} \mod p \) holds or not, respectively.

3.2 Security analysis

In the following, we present that the proposed protocol can withstand the replay attack and also the off-line password guessing attack.

**Replay attack:** Fig. 4 shows the process of the replay attack for our protocol.

![Fig. 4. The replay attack](image)

Upon seeing \( X_1 \) sent by Alice in e.2) of the original SAKA scheme, an adversary (Eve) can masquerade as Bob to re-send it back to Alice in e.3) as \( Y_i \). Then Alice will compute

\[
Y = Y_1^{-1} \mod p = X_1^{-1} \mod p = g^a \mod p \\
\text{key}_1 = Y^a \mod p = g^{a^2} \mod p \\
K_1 = g^e \mod p = p - 1
\]

In the key validation phase of our protocol, Alice compute \( X' = Y^{K_1} \mod p = (g^a)^{K_1} \mod p \) and...
sends it to Bob. Then Eve first intercepts $X'$. But, Eve cannot masquerade as Bob to re-send $X'$ back to Alice in Step 2 as $Y'$. Since the message $X' = Y^{K_1} \mod p$ resent by Eve is not equal to $g^{K_1} \mod p$, i.e. $X' = Y^{K_1} \mod p \neq g^{K_1} \mod p$. Accordingly, the enhanced protocol is secure against the replay attack. Conversely, even though Step 1 and Step 2 are exchanged, the enhanced protocol is also secure to the replay attack. Consequently, our enhanced protocol can withstand to the replay attack.

**Off-line password guessing attack:** In our protocol, we assume that Eve who first guesses the password PWD and computes a corresponding $Q$ and $Q^3 \mod (p-1)$ from PWD. And Eve intercepts four public messages, $X_1$, $Y_1$, $X'$ and $Y'$, on the public channel. To obtain the correct password, Eve must obtain the equation with the same value using the guessed password and four public messages. In other word, Eve must know $K_1$ and $K_2$ to validate whether the guessed password really corrects or not: However, (I) Even though Eve knows two messages $X_1$ and $Y_1$, he/she cannot compute $K_1$ and $K_2$ without knowing secret random number $a$ and $b$. (II) Even though Eve knows two messages $X'$ and $Y$, he/she cannot compute $K_1$ due to the discrete logarithm problem. (III) Even though Eve knows two messages $Y'$ and $g$, he/she also cannot compute $K_2$ by the reason as (II). Accordingly, Eve cannot obtain the equation with the same value using the guessed password and four public messages. Therefore, the enhanced protocol is secure against the off-line password guessing attack.

**4. CONCLUSION**

In this paper, we propose a robust authenticated key agreement protocol in which two communication parties exchange a secret session key and authenticate each other. The protocol can be considered as an improvement of simple authenticated key agreement algorithm. However, the proposed protocol is more significant in that it not only secure against the replay attack by using asymmetric message in the enhanced validation steps but also resistant to the off-line password guessing attack due to secret random numbers and the discrete logarithm problem.

**REFERENCE**