Cryptanalysis of Simple Authenticated Key Agreement Protocols*

Ting-Yi Chang‡ Chou-Chen Yang† Min-Shiang Hwang†

Department of Management Information System†
National Chung Hsing University
250 Kuo Kuang Road,
402 Taichung, Taiwan, R.O.C.
Email: mshwang@nchu.edu.tw
Fax: 886-4-23742337

Department of Computer and Information Science‡
National Chiao Tung University
1001 Ta Hsueh Road,
Hsinchu, Taiwan, R.O.C.
Email: tychang@cis.nctu.edu.tw

April 14, 2004

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-004.

†Responsible for correspondence: Prof. Min-Shiang Hwang

Cryptanalysis of Simple Authenticated Key Agreement Protocols

Abstract

In this article, we will present a modification attack and a dictionary attack to subvert the security of the Tseng scheme and the Ku-Wang scheme. As we know, no existing schemes of simple authenticated key agreement protocols can successfully withstand our modification attack.

Keywords: Cryptography, information security, key agreement, key exchange.

1 Introduction

The Diffie-Hellman key agreement protocol [1] is a method for distributing a common session key to be shared between two users (Alice and Bob) to establish a secret communication over an insecure channel. The common session key can be determined by either user based on her/his own secret key and the partner's public key. The security here comes from the difficulty of computing discrete logarithms over a finite field. However, the scheme is vulnerable to the man-in-middle attack. Since the Diffie-Hellman key agreement protocol does not authenticate the user's identity, a man-in-the-middle (Eve) can easily reveal the conditional message transferred between the two parities.

In 1999, Seo and Sweeney [6] proposed a simple authenticated key agreement (SAKA) protocol based on the pre-shared password method. They modified the Diffie-Hellman scheme to provide user authentication and to verify the validity of common session key. Unfortunately, Sun [7], Tseng [8], and Lu et al. [5] separately showed the fact that the Seo-Sweeney SAKA scheme is vulnerable to the mounting replay attack and dictionary attack. At the same time, Tseng also proposed an improved version of the Seo-Sweeney SAKS scheme to help withstand the replay attack. Later, however, Ku and Wang [3] pointed out that the Tseng SAKA scheme is insecure in front of the backward replay attack and modification attack. To fight against these attacks, Ku and Wang enhanced the security of the Seo-Sweeney SAKA scheme to repair the flaws in the Tseng SAKA scheme.

On the other hand, Lin et al. [4] also offered an improved version of the Seo-Sweeney SAKA scheme. They claimed that their scheme could deal with the problems Sun had pointed out while providing perfect forward secrecy. Recently, Hsieh et al. [2] pointed out that Lin et al.'s SAKA scheme is vulnerable to the dictionary attack. In this article, we will show that the verification of the common session key cannot in fact be achieved by any of the schemes mentioned above [3, 4, 6, 8]. Any adversary can mount the modification attack to fool the two parties such that they believe that they have agreed upon the session key when the fact is the session keys are different for the two parties. The dictionary attack also threatens the security of the Tseng SAKA scheme and the Ku-Wang SAKA scheme.

The organization of this article is as follows. In the next section, we will briefly review the Tseng SAKA scheme and the Ku-Wang SAKA scheme, respectively. In Section 3, we will show that both schemes are vulnerable to the modification attack and dictionary attack. Furthermore, we will also show that the modification attack can easily damage the security of all the other schemes mentioned above. Finally, we will conclude this article in Section 4.

2 Literature Reviews

In this section, we will separately review the Tseng SAKA scheme [8] and the Ku-Wang SAKA scheme [3].

2.1 The Tseng SAKA Scheme

As in the Diffie-Hellman scheme [1], the system publishes two values p and g, where p is a large prime and g is a primitive element in $GF(p)$. Assume that Alice and Bob share a secret password S and a predetermined way to generate the two integers $Q \bmod p-1$ and $Q^{-1} \bmod p-1$ from the password S. The protocol is composed of two phases, the key establishment phase and the key verification phase, as follows:

Key establishment phase

(e. 1) Alice randomly chooses an integer a and computes $X_1 = g^{aQ} \bmod p$. Then, Alice sends X_1 to Bob.

(e. 2) Bob randomly chooses an integer b and computes $Y_1 = g^{bQ} \bmod p$. Then, Bob sends Y_1 to Alice.

(e. 3) When Y_1 is received, Alice computes $Y = Y_1^{Q^{-1}} = g^b \bmod p$ and $Key_1 = Y^a = g^{ab} \bmod p$.

(e. 4) When X_1 is received, Bob computes $X = X_1^{Q^{-1}} = g^a \bmod p$ and $Key_2 = X^b = g^{ab} \bmod p$.

Key verification phase

(v. 1) Alice sends Y to Bob.

(v. 2) Bob sends X to Alice.

(v. 3) Alice and Bob check whether $X = g^a \bmod p$ and $Y = g^b \bmod p$ hold or not, respectively. If they hold, Alice and Bob are sure that they have the common session key.

2.2 The Ku-Wang SAKA Scheme

To solve the security problems [3] in the Tseng SAKA scheme, Ku and Wang proposed an improved version of the Seo-Sweeney SAKA scheme. The key establishment phase in the Ku-Wang SAKA Scheme is the same as that in the Tseng SAKA scheme. We only briefly review the key verification phase of the Ku-Wang SAKA scheme that makes a difference.

Key verification phase

(v. 1) Alice computes $K_1 = (Key_1)^Q = g^{abQ} \bmod p$. Then, Alice sends K_1 to Bob.

(v. 2) When K_1 is received, Bob checks whether $Key_2 = K_1^{Q^{-1}} \bmod p$. If it holds, Bob believes that he has obtained the correct X_1 and Alice has obtained the correct Y_1. Then, he sends X to Alice.

(v. 3) When X is received, Alice checks whether $X = g^a \bmod p$. If it holds, Alice believes that she has obtained the correct Y_1 and Bob has obtained the correct X_1.

Though Ku and Wang claim that their scheme can withstand the modification attack, we will show that their scheme not only is vulnerable to the modification attack but also fall for the dictionary attack in the next section.

3 The Modification Attack and Dictionary Attack

Here in this section, we will have a look at the security problems of the schemes just mentioned [3, 4, 6, 8] in Table 1.

Ku and Wang have shown that the Tseng SAKA scheme is vulnerable to the backward replay attack and modification attack. They proposed an improved version of the Sao-Sweeny SAKA scheme. However, we will present another modification attack on the Tseng SAKA scheme, and it will still successfully break the Ku-Wang SAKA scheme as well as the others. On the other hand, we will show that the Tseng SAKA scheme and the Ku-Wang SAKA scheme are also weak in front of the dictionary attack.

Table 1: Summary of previously proposed schemes in security

	Seo-Sweeney [6]	Tseng [8]	Lin et al. [4]	Ku-Wang [3]
Withstand Man-in-Middle Attack	Yes	Yes	Yes	Yes
Withstand Dictionary Attack	No	**No**	No	**No**
Withstand Replay Attack	No	Yes	Yes	Yes
Withstand Backward Replay Attack	No	No	Yes	Yes
Withstand Modification Attack	**No**	**No**	**No**	**No**
Perfect Forward Secrecy	No	Yes	Yes	**No**

Modification Attack

Assume that Eve, who is an adversary, tries to fool Alice and Bob into believing a wrong session key in the Tseng SAKA scheme. Eve first prepares a value $e \bmod p-1$ and its inversion $e^{-1} \bmod p-1$. In the key establishment phase, upon seeing X_1 sent by Alice in Step (e. 1), Eve replaces it with $X_1' = (X_1)^e = g^{aQe} \bmod p$. Then, Alice performs Step (e. 3) and Bob performs Steps (e. 2) and (e. 4). Alice and Bob will separately obtain the session keys $Key_1 = Y^a = g^{ab} \bmod p$ and $Key_2 = X^b = g^{abe} \bmod p$, where $Y = (Y_1)^{Q^{-1}} = g^b \bmod p$ and $X = (X_1')^{Q^{-1}} = g^{ae} \bmod p$. Next, they separately verify the validity of the session keys Key_1 and Key_2. After receiving Y in Step (v. 1), the check equation $Y = g^b \bmod p$ will hold in Step (v. 3) on Bob's side, so he will believe that he and Alice have agreed on a common session key. Upon seeing X is sent by Bob in Step (v. 2), Eve replaces it with $X' = (X)^{e^{-1}} = g^a \bmod p$. The check equation $X' = g^a \bmod p$ will hold in Step (v. 3) on Alice's side, and she will too believe that she and Bob have agreed on a common session key. However, $Key_1 = g^{ab} \bmod p$ is not equal to $Key_2 = g^{abe} \bmod p$.

In the Ku-Wang SAKA scheme, Eve performs the same work in the key establishment phase. In the key verification phase, upon seeing K_1 is sent by Alice in Step (v. 1), Eve replaces it with $K_1' = (K_1)^e = (g^{abQ})^e \bmod p$. The check equation $Key_2 = (K_1')^{Q^{-1}} = g^{abe} \bmod p$ in Step (v. 2) will hold. Bob will believe that he has obtained the correct X_1 and Alice has obtained the correct Y_1. Then, Bob sends $X = g^{ae} \bmod p$ to Alice. Eve replaces it with $X' = (X)^{e^{-1}} = g^a \bmod p$. The check equation $X_1' = g^a \bmod p$ in Step (v. 3) will hold, which will make Alice believe that she has obtained the correct Y_1 and Bob has obtained the correct X_1. However, $Key_1 = g^{ab} \bmod p$ is not equal to $Key_2 = g^{abe} \bmod p$.

As a matter of fact, the modification attack can easily be mounted to break all the existing SAKA-related methods [4, 6]. Because all the SAKA schemes have the common weakness, any adversary can use some value to replace the original value sent by Alice in the key establishment phase and then use its inversion to make Bob return to the original value sent to Alice in the key verification phase. This will male Alice and Bob believe the wrong session key.

Dictionary Attack

Assume that Eve, who is an adversary, tries to reveal the secret values Q and Q^{-1} shared between Alice and Bob by mounting the dictionary attack. In the Tseng SAKA scheme, Eve first intercepts $X_1 = g^{aQ} \bmod p$ sent by Alice in Step (e. 1) and masquerades as Bob to send $Y_1 = g^e \bmod p$ in Step (e. 2). Alice computes $Y = Y_1^{Q^{-1}} = g^{eQ^{-1}} \bmod p$ and $Key_1 = Y^a = g^{eQ^{-1}a} \bmod p$. In Step (v. 1), Alice sends Y to Bob. After intercepting Y, Eve can verify the correctness of the guessing password by checking whether $Y^Q = g^e \bmod p$

holds or not because $Y^Q = Y_1^{Q^{-1}Q} = g^e \bmod p$.

For the same reason, in the Ku-Wang SAKA scheme, Eve performs the same work in the key establishment phase. In the key verification phase, Alice computes $K_1 = (Key_1)^Q = g^{ae} \bmod p$ and sends it to Bob in Step (v .1). After intercepting K_1, Eve can verify the correctness of the guessing password by checking whether $K_1 = (X_1)^{Q^{-1}} \bmod p$ holds or not because $K_1 = (X_1)^{Q^{-1}} = g^{ae} \bmod p$.

In both schemes, if the password S is poorly chosen, the adversary can determine Q or Q^{-1} by using the equations to verify if the guessing password is correct. On the other hand, in the Ku-Wang SAKA scheme, when a password is compromised, the old session key Key_1 can be recoverd by computing $K_1^{Q^{-1}} = Key_1 = g^{ab} \bmod p$. Therefore, their scheme cannot provide perfect forward secrecy.

4 Conclusion

In this article, we have presented the modification attack and the dictionary attack to subvert the security of the Tseng scheme and the Ku-Wang scheme. As we have proved, the Ku-Wang scheme is weak against the modification attack and the dictionary attack; moreover, the modification attack can be used to break all the existing SAKA-related schemes.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[2] Bin-Tsan Hsieh, Hung-Min Sun, and Tzonelih Hwang, "Cryptanalysis of enhancement for simple authenticated key agreement algorithm," *IEE Electronics Letters*, vol. 38, no. 1, pp. 20–21, 2002.

[3] Wei-Chi Ku and Sheng-De Wang, "Cryptanalysis of modified authenticated key agreement protocol," *IEE Electronics Letters*, vol. 36, no. 21, pp. 1770–1771, 2000.

[4] Iuon-Chang Lin, Chin-Chen Chang, and Min-Shiang Hwang, "Security enhancement for the simple authentication key agreement algorithm," in *The Twenty-Fourth Annual International Computer Software and Applications Conference(COMPSAC)'2000*, pp. 113–115, 2000.

[5] Eric Jui-Lin Lu, Cheng-Chi Lee, and Min-Shiang Hwang, "Cryptanalysis of some authenticated key agreement protocols," *International Journal of Computational and Numerical Analysis and Applications*, vol. 3, no. 2, pp. 151–157, 2003.

[6] D. Seo and P. Sweeney, "Simple authenticated key agreement algorithm," *IEE Electronics Letters*, vol. 35, no. 13, pp. 1073–1074, 1999.

[7] H. Sun, "On the security of simple authenticated key agreement algorithm," in *Proceedings of the Management Theory Workshop'2000*, 2000.

[8] Yuh-Min Tseng, "Weakness in simple authenticated key agreement protocol," *IEE Electronics Letters*, vol. 36, no. 1, pp. 48–49, 2000.