藉由密碼認證之金鑰協議方法之改進
Improved Authenticated Key Agreement Scheme via Password Authentication

Hung-Min Sun (孫宏民)
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan 70101
hmsun@mail.ncku.edu.tw

Bin-Tsan Hsieh(謝濱燦)
Institute of Information Management
Chaoyang University of Technology
Wufeng, Taichung County, Taiwan 413
bintsan@ms27.hinet.net

Abstract

Recently Kwon and Song proposed an authenticated key agreement scheme for sharing g^{xy} of the Diffie-Hellman scheme via password authentication. In this paper, we propose an improvement on Kwon and Song's scheme, which can reduce the computational cost of their scheme.

Keywords: ***Cryptography, Key Agreement, Password Authentication***

1 Introduction

The Diffie-Hellman key agreement scheme [1] can be used to share a session key between two communication parties. However, due to its flexibility, the Diffie-Hellman key agreement scheme suffers from the man-in-the-middle attack. Hence if the Diffie-Hellman key agreement scheme is used as a key agreement scheme between a client and a server, user authentication is required for this scheme to prevent man-in-the-middle attacks. As a password authentication scheme is still the most popular one for user authentication in a distributed environment, several authenticated key agreement schemes [2-6] based on password were proposed in the past. These schemes can provide the property of perfect forward secrecy that a compromised password doesn't reveal an old session key [4] via the Diffie-Hellman problem. Note that based on password, it seems that there exist some trivial methods to share a session key between a client and a server, however, these methods cannot provide perfect forward secrecy.

Recently Kwon and Song [7] proposed an authenticated key agreement scheme for sharing g^{xy} of the Diffie-Hellman scheme via password authentication. Compared with A-EKE [3], their scheme needs only four steps that are less than seven steps required in A-EKE. Compared with other four-step schemes — B-SPEKE [5] and SRP [6], Kwon and Song's scheme has minimal constraints: p is a non-smooth prime and g is a primitive root of $GF(p)$, while B-SPEKE [5] need that p is a safe prime against a small subgroup confinement [4] and only a recommendation is sufficient for SRP [6]. In this paper, we propose an improvement on Kwon and Song's scheme, which can reduce the computational cost of their scheme.

The remainder of this paper is organized as follows. In section 2, we briefly review Kwon and Song's authenticated key agreement scheme. In section 3, we propose an improvement on Kwon and Song's authenticated key agreement scheme. Finally, we conclude this paper in section 4.

2 Kwon and Song's Authenticated Key Agreement Scheme

System Steup: Let $f(\ )$ be a simple mapping function which extends the length of a preimage to that of a secret exponent, e.g. 200 bits, in order to make a discrete logarithm problem infeasible. Let $h(\ )$ be a one-way hash function and p be a large prime modulus. Let + and $\oplus$ stand for the multiplication operator and the exclusive-or operator respectively. Each user keeps a secret s as his password and the server stores a triple $(id, t \oplus s, g^v \pmod p)$ for each user where $v = f(t+s)$ and t is a random number chosen by the server. Here only $g^v \pmod p$ needs to be kept in secret. Note that for security requirement, no password is directly stored in the server (this concept is commonly accepted and widely used in practical, e.g. Unix system).

For providing both authentication and key agreement, a user inputs his id and password s. Then a client application A sends the id to the server B and then chooses a random number $x \in_R Z_{p-1} \setminus \{0\}$:

Step 1. $A \rightarrow B : id$

A computes $g^x \pmod p$ and keeps it while waiting for Step 2. B reads $(id, t \oplus s, g^v \pmod p)$ from the storage and hence gets $g^v \pmod p$. Then B chooses a random number $y \in_R Z_{p-1} \setminus \{0\}$, computes $g^y + g^v \pmod p$, and sends $t \oplus s$ and $g^y + g^v \pmod p$ to A.

Step 2. $B \rightarrow A : t \oplus s, g^y + g^v \pmod p$

B computes $(g^v)^y \pmod p$ and keeps it while waiting for Step 3. A obtains t from his password s and $t \oplus s$, and computes $v = f(t+s)$ and $g^v \pmod p$. So A can recover $g^y \pmod p$ from $g^v \pmod p$ and $g^y + g^v \pmod p$. Then A computes $(g^y)^v \pmod p$ and $(g^y)^x \pmod p$, and adds

$(g^y)^v \pmod p$ to $g^x \pmod p$. At last, A computes a hash image $h(g^y + g^v \pmod p, (g^y)^x \pmod p)$ and sends the following message to B.

Step 3. $A \rightarrow B : (g^y)^v + g^x \pmod p,$
$h(g^y + g^v \pmod p, (g^y)^x \pmod p))$

B can recover $g^x \pmod p$ from Step 3 because $(g^y)^v \pmod p = (g^v)^y \pmod p$. B computes $(g^x)^y \pmod p$ and $h(g^y + g^v \pmod p, (g^x)^y \pmod p)$. Then he checks whether $h(g^y + g^v \pmod p, (g^x)^y \pmod p) \stackrel{?}{=} h(g^y + g^v \pmod p, (g^y)^x \pmod p)$. If it holds, then B computes a response hash image and sends it to A:

Step 4. $B \rightarrow A : h((g^y)^v + g^x \pmod p$
$, (g^x)^y \pmod p))$

A computes $h((g^y)^v + g^x \pmod p, (g^y)^x \pmod p))$ and compares it with $h((g^y)^v + g^x \pmod p, (g^x)^y \pmod p))$. If they match each other, then A and B are able to agree on $g^{xy} \pmod p$ of the Diffie-Hellman scheme and compute a new session key from it, say $K=h(g^{xy} \pmod p)$.

3 An Improvement on Kwon and Song's Scheme

In this section, we propose an improvement on Kwon and Song's scheme.

Similar to Kwon and Song's scheme, a user inputs his *id* and password *s*. Then a client application *A* sends the *id* to the server *B* and then chooses a random number $x \in_R Z_{p-1} \setminus \{0\}$:

Step 1. $A \rightarrow B : id$

A computes $g^x \pmod p$ and keeps it while waiting for Step 2. *B* reads $(id, t \oplus s, g^v \pmod p)$ from the storage and hence gets $g^v \pmod p$. Then *B* chooses a random number $y \in_R Z_{p-1} \setminus \{0\}$, computes $g^y - g^v \pmod p$, and sends the following message to *A*:

Step 2. $B \rightarrow A : t \oplus s, g^y - g^v \pmod p$

A obtains *t* from his password *s* and $t \oplus s$, and computes $v = f(t + s)$ and $g^v \pmod p$. So *A* can recover $g^y \pmod p$ by $g^y = (g^y - g^v) + g^v \pmod p$. *A* computes $g^x - g^v \pmod p$, $(g^y)^x \pmod p$ and a hash image $h((g^y)^x \pmod p, g^v + g^y \pmod p))$, and then sends the following message to *B*.

Step 3. $A \rightarrow B : g^x - g^v \pmod p,$
$h((g^y)^x \pmod p, g^v + g^y \pmod p))$

B recovers $g^x \pmod p$ by $g^x = (g^x - g^v) + g^v \pmod p$, computes $(g^x)^y \pmod p$ and $h((g^x)^y \pmod p, g^v + g^y \pmod p)$. Then he checks whether $h((g^x)^y \pmod p, g^v + g^y \pmod p) \stackrel{?}{=} h((g^y)^x \pmod p, g^v + g^y \pmod p)$ which is from *A*. If it holds, then *B* computes a response hash image and sends it to *A*:

Step 4. $B \rightarrow A : h((g^x)^y \pmod p, g^v + g^x \pmod p))$.

A computes $h((g^y)^x \pmod p, g^v + g^x \pmod p))$ and compares it with $h((g^x)^y \pmod p, g^v + g^x \pmod p))$ which is from *B*. If they match each other, then *A* and *B* are able to agree on $g^{xy} \pmod p$ of the Diffie-Hellman scheme and compute a new session key from it, say $K=h(g^{xy} \pmod p)$.

Security Considerations

(1) The improved scheme also satisfies the property of perfect forward secrecy due to the Diffie-Hellman problem: It is clear that a password compromise only reveal old $g^x \pmod p$ and $g^y \pmod p$. It doesn't reveal $g^{xy} \pmod p$ and the old session key $K=h(g^{xy} \pmod p)$ because *x* and *y* are unknown to an intruder.

(2) The improved scheme is secure against the man-in-the-middle attacks because both $g^x \pmod p$ and $g^y \pmod p$ are unknown to a middle person. Moreover, any attempts to modify messages in the steps of the proposed scheme will be detected due to the hash images.

(3) No verifiable information can be used for guessing *v* (or $g^v \pmod p$). An intruder can know only $g^y - g^v \pmod p$ and $g^x - g^v \pmod p$, but both $g^y \pmod p$ and $g^x \pmod p$ are unknown to the intruder. Note that although $g^x - g^y \pmod p$ can be computed from these both, but this is not vulnerable to security. Even with the knowledge of $g^{xy} \pmod p$, the intruder still cannot guess *v* (or $g^v \pmod p$) by $h((g^y)^x \pmod p, g^v + g^y \pmod p))$ or $h((g^x)^y \pmod p, g^v + g^x \pmod p))$ because both $g^y \pmod p$ and $g^x \pmod p$ are still unknown to the intruder. Such an unverifiability disables off-line guessing attacks such as partition attacks or arbitrary exponent attacks [3,5].

Efficiency considerations

The total amount of execution time for the four-step schemes can be evaluated by the number of modular exponentiations which are computed by both in parallel, i.e., $E(client : server)$. In Kwon and Song's scheme, the modular exponentiations

computed by the client and the server in parallel are:

between Step1 and Step2 : $E(g^x : g^y)$

between Step2 and Step3 : $E(g^v : (g^v)^y)$, $E((g^y)^v :)$, and $E((g^y)^x :)$

between Step3 and Step4 : $E(: (g^x)^y)$.

Therefore there are five parallel modular exponentiations required for Kwon and Song's.

In the improved scheme, the modular exponentiations computed by the client and the server in parallel are:

between Step1 and Step2 : $E(g^x : g^y)$

between Step2 and Step3 : $E(g^v :)$ and $E((g^y)^x :)$

between Step3 and Step4 : $E(: (g^x)^y)$.

So, there are only four parallel modular exponentiations required for the improved scheme.

4 Conclusions

In this paper, we propose an improvement on Kwon and Song's authenticated key agreement scheme via password authentication. The improved scheme is a four-step protocol which is the same as Kwon and Song's scheme, and also provides the perfect forward secrecy and the capability of preventing man-in-the-middle attacks. Compared with Kwon and Song's scheme in a term of efficiency, the improved scheme requires only four parallel modular exponentiations which are less than that of five required for Kwon and Song's scheme.

Reference

[1] DIFFIE, W., and HELLMAN, M.: 'New directions in cryptography', IEEE Trans., 1976, IT-22, (6), pp. 644-654

[2] BELLOVIN, S., and MERRIT, M.: 'Encrypted key exchange: password-based protocols secure against dictionary attacks'. IEEE Comp. Society Symp. On Research in Security and Privacy, 1992, pp. 72-84

[3] BELLOVIN, S., and MERRIT, M.: 'Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise'. ACM Conf. Comp. And Comm. Security, 1993, pp. 244-250

[4] JABLON, D.: 'Strong password-only authenticated key exchange', ACM Comput. Commun. Rev., 1996, 20, (5) pp. 5-26

[5] JABLON, D.: 'Extended password key exchange protocols'. WETICE Workshop on Enterprise Security, 1997

[6] WU, T.: 'Secure remote password protocol'. Internet Society Symp. Network and Distributed System Security, 1998

[7] KWON, T. and SONG, J.: 'Secure agreement scheme for g^{xy} via password authentication'. Electronics Letters, 1999, 35, (11), pp.892-893