Information Technology Journal 6 (4): 602-606, 2007
ISSN 1812-5638

Password-Based Key Agreement Protocol

[1,3]Chunbo Ma, [2]Jun Ao and [1]Jianhua Li
[1]School of Information Security Engineering, Shanghai Jiao Tong University,
Shanghai, 200030, People's Republic of China
[2]State Key Laboratory for Radar Signal Processing,
Xidian University, Xi'an, Shanxi, 710071, People's Republic of China
[3]The State Key Laboratory of Information Security,
Institute of Software of Chinese Academy of Sciences, Beijing, 100049, People's Republic of China

Abstract: In this study, we will review Kim *et al.*'s protocol and Lee *et al.*'s protocol respectively and show that their protocols are not secure as they have claimed. Then, we firstly present our improved password-based key agreement protocol based on Kim *et al.*'s scheme and show that it withstands password guessing attack and illegal modification. Then we propose an improved Verifier-based key agreement protocol based on Lee *et al.*'s scheme and demonstrate that it is secure against Stolen-verifier attack and password guessing attack.

Key words: Password, verifier, key agreement protocol, password guessing attack, stolen-verifier attack

INTRODUCTION

Key agreement protocols allow two or more parties to negotiate a common secret among them over an insecure network. The common secret may subsequently be used to achieve some security requirements, such as confidentiality or data integrity. The first two-party key agreement is the Diffi-Hellman protocol (Diffe and Hellman, 1976). However, the basic Diffie-Hellman protocol is vulnerable to man-in-middle attack since the parties involved in the protocol have no way to authenticate each other.

Password as an important technique to implement symmetric authenticated key agreement protocols has recently received growing attention. Since a pioneering approach that resists the password guessing attacks was introduced in Lomas *et al.* (1989), there have been a number of password-based authenticated key agreement protocols (Seo and Sweeney, 1999; Abdalla *et al.*, 2005; Ryu *et al.*, 2004; Kim *et al.*, 2004) on the framework of Diffie-Hellman scheme. However, traditional password-based protocols are susceptible to dictionary attack, for example Kim *et al.* (2004) protocol, since many users tend to choose memorable passwords of relatively low entropy. In some password-based key agreement protocols, the information drawn from the password is completely shared among the participants. Hence, in the case of some party compromise, the intruder will obtain the entire secret message. Subsequently, the intruder can pretend anyone he likes. To a key agreement protocol running in centralized system, it is also vulnerable to Stolen-Verifier attack in case of Server compromise. In Stolen-Verifier attack, an intruder, who obtains the Verifier from the Server, tries to impersonate a desirable client and to agree upon a session key with the Server. In order to deal with the risk of server compromise, Lee *et al.* (2004) proposed a Verifiable-based key agreement protocol PAKA-X. In this protocol, the client uses a plaintext of the password, while the server stores a Verifier for the password. So the protocol does not allow an adversary who compromises the server to impersonate a user without actually running a dictionary attack on the password file. However, the protocol, PAKA-X, is not secure against Stolen-verifier attack as Lee *et al.* (2004) have claimed.

In this study, we will briefly review Kim *et al.* (2004) password-based key agreement protocol and show its flaws. Then, we present an improved protocol to resist password guessing attack and illegal modification. Next, we will review Lee *et al.* (2004) Verifier-based key agreement protocol and demonstrate that it is susceptible to Stolen-verifier attack. Thereafter, we propose a novel protocol, which withstands Stolen-verifier attack and password guessing attack.

BACKGROUND

Bilinear Maps: Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a

Corresponding Author: Chunbo Ma, School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, 200030, People's republic of China

cyclic multiplicative group of the same order q. Assume that the discrete logarithm in both G_1 and G_2 is intractable. A bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ and satisfies the following properties:

- Bilinear: $e(g^a, p^b) = e(g, p)^{ab}$. For all g, p $\in G_1$ and a, b $\in Z_q$, the equation holds.
- Non-degenerate: There exists p $\in G_1$, if e(g, p) = 1, then g = O.
- Computable: For g, p $\in G_1$, there is an efficient algorithm to compute e(g, p).

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. Pairings and other parameters should be selected in proactive for efficiency and security.

Complexity assumptions:

- **Computation Diffie-Hellman Problem [CDHP]:** Given g^a, $g^b \in G_1$ for unknowns a,b $\in Z^*_q$ computing $g^{ab} \in G_1$ is intractable.
- **Decision Diffie-Hellman Problem [DDHP]:** Given g^a, g^b, $g^c \in G_1$ for unknowns a, b, c $\in Z^*_q$, deciding whether $e(g, g)^{abc} = W$ is intractable.
- **K-Strong Diffie-Hellman Assumption (Zhang *et al.*, 2004):** Choose x $\in Z^*_q$ uniformly at random and given $\{g, g^x, g^{x^2}, \cdots, g^{x^k}\}$, computing $g^{(c+x)^{-1}}$ is intractable even if the attacker is allowed to adaptively choose c $\in Z^*_q$.

Such group can be found in supersingular elliptic curve or hyperelliptic curve over finite field (Boneh *et al.*, 2001).

KIM *et al*'s PROTOCOL

Review of Kim *et al*'s Protocol: Let n be a large prime and g be a generator with order n-1 in GF(n) as the original Diffie-Hellman protocol. We assume that two participants, Alice and Bob, want to negotiate a session key using their pre-shared secret password. In this scheme, these two participants are allowed to pre-compute two integers W and W^{-1} from password in some predetermined way. They will exchange some message in an authenticated way using the pre-shared password and eventually agree upon a session key. The following is the description on this protocol.

Alice chooses a random number a $\in Z^*_n$, computes $X_A = g^{aW}$ and sends X_A to another participant Bob. Bob chooses a random number b $\in Z^*_n$, computes $X_B = g^{bW}$ and $Y_B = X_A^{W^{-1}} = g^a$ and then sends X_B and Y_B to Alice. After receiving the message from Bob, Alice checks if $Y_B = g^a$. If they match each other, Alice authenticates Bob. Then, Alice computes $Y_A = X_B^{W^{-1}} = g^b$ and the session key $K_{AB} = Y^a = g^{ab}$. Thereafter, Alice sends Y_A to Bob. When Bob receives Y_A from Alice, he checks if $Y_A = g^b$. If they match each other, Bob authenticates Alice. Then, Bob computes the session key $K_{BA} = Y^a_B$.

Weaknesses: Kim *et al.* claim that their protocol has the property of perfect forward secrecy and is powerful against data modification attack. However, their protocol is vulnerable to off-line password guessing attack and illegal modification (Yoon and Yoo, 2005).

Generally speaking, a password has limited entropy and is easy for human to remember. However, this property makes the protocol designed using password susceptible to password guessing attack. Since the two participants, Alice and Bob, exchange message over an open network, it is easy for the attacker to intercept the information (X_A, Y_B). Then the attacker guesses the password and computes W and W^{-1}. Subsequently, the attacker computes Y^W_B and checks if $X_A = Y^W_B$. If the equation doesn't hold, he will repeat above performance and try to find the matching password. Since a password's entropy is low, it is reasonable for an attacker to find the matching password in a polynomial time.

Not only to make password guessing attack on Kim *et al.*'s protocol, the attacker can illegally modify the session key as follows. Upon intercepting message $X_A = g^{aW}$, the attacker can choose a random number t $\in Z^*_n$ and compute $X'_A = (X'_A)^2$. Similarly, when intercepting message (X_B, Y_B), the attacker will replace Y_B with $Y_B^{t^{-1}}$ and X_B with X^t_B, respectively. Thereafter, Alice and Bob can negotiate a session key. Note that the attacker can't get the session $K_{AB} = K_{BA} = g^{abt}$. Note that the attacker can't get the sessi key, even though he has modified it. In other words, the attacker has ability to modify the session key in this attack model, but the modification is not harmful to the security of the protocol.

THE PROPOSED KEY AGREEMENT PROTOCOL

In this section, we will present a password-based key agreement protocol, which can withstand the off-line password guessing attack and illegal modification. In Kim *et al.*'s protocol, the attacker can easily intercept the information (X_A, Y_B) and then implement the password guessing attack. Hence, how to eliminate the information used to verify the guess is the most important issue. In this aspect, we can get some inspiration from (Kwon, 2004). The proposed password-based key

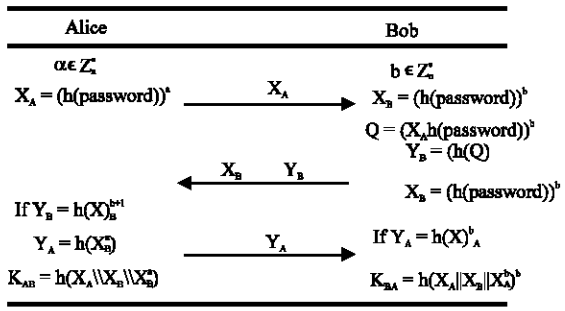

Fig. 1: The proposed protocol

agreement protocol consists of four steps and a strong one-way function h: {0, 1}* → <g> will be used in designing the protocol.

Step 1: Alice chooses a random number $a \in Z^*_n$, computes, $X_A = (h(password))^a$ and sends X_A to Bob.

Step 2: Bob chooses a random number $b \in Z^*_n$, computes $X_B = (h(password))^b$ and $Q = (X_A \cdot h(password))^b$ using the pre-shared password. Then Bob sends X_B and $Y_B = h(Q)$ to Alice.

Step 3: Upon receiving the message (X_B, Y_B) from Bob, Alice checks if Y_B is equal to $h(X^{a+1}_B)$. If they match each other, Alice authenticates Bob. Subsequently, Alice computes $Y_A = h(X^a_B)$ and sends it to Bob.

Step 4: After receiving the message Y_A, Bob checks if Y_A is equal to $h(X^b_A)$. If they match each other, Bob authenticates Alice.

Step 5: Alice computes the session key $K_{AB} = h(X_A \| X_B \| X^a_B)$ and Bob computes $K_{BA} = h(X_A \| X_B \| X^b_A)$ as the session key.

The proposed protocol can be illustrated in Fig. 1.

Upon successfully implementing above protocol, the two participants, Alice and Bob, will share a common secret session key. Thereafter, they can establish a secure communication channel using this session key.

SECURITY ANALYSIS

Withstand password guessing attack: The dictionary attack can be classified two kinds, i.e. on-line guessing attack and off-line guessing attack. To the on-line password guessing attack, the participants can defeat the attacker by choosing appropriate trail intervals. In an off-line guessing attack, the attacker should repeatedly guess the password and verify its correctness by the message gathered in an off-line manner. In our proposed protocol,

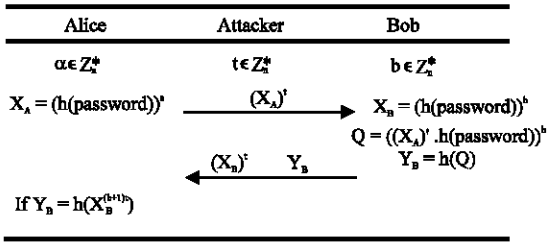

Fig. 2: The illegal modification

the attacker can obtain (X_A, Y_A) and (X_B, Y_B) through the network. However, a random number, a or b, is involved in each value. In other words, an attacker who obtains the message exchanged between Alice and Bob can't find appropriate information to verify his guess on the password. Therefore, we say that the protocol is secure against password guessing attack.

Withstand illegal modification: Since an attacker can intercept the message exchanged over the network, he has ability to modify the message and send it to desirable receiver. However, although the attacker modifies them, the proposed protocol can detect this attack. Assume that the attacker chooses a random number and the illegal modification can be illustrated in Fig. 2.

In the case of illegal modification, Alice computes

$$X_B^{(a+1)t} = (h(password))^{b(a+1)t} = (h(password))^{abt+bt}$$

Bob computes

$$Q = (h(password))^{(ta+1)b} = (h(password))^{abt+b}$$

Since h is a strong one-way function, the attacker can't modify Y_B as he wish. Obviously, $Y_B \neq h(X_B^{(a+1)t})$ and the illegal modification can be detected.

LEE *et al.*'s KEY AGREEMENT PROTOCOL

Lee *et al.* (2004) presented a Verifier-based key agreement protocol in 2004. They claimed that their protocol was secure in the case of server compromise. It means that if the attacker intrudes the server, he can't get enough information to impersonate a user without actually running a dictionary attack on the password file.

Review of Lee *et al.*'s protocol: We assume that there exists a initialization in which the user Alice chooses a password, computes $v = g^{h(ID_A, ID_S, password)}$ and then sends v to the Server as the Verifier. In addition, there exists a strong one-way function h: {0, 1}* → Z^*_q. We briefly review Lee *et al.*'s protocol as follows:

Alice chooses a random number $a \in Z^*_q$, computes $X_A = g^a \oplus v$ and then sends (ID_A, X_A) to Server. Upon receiving the message from Alice, Server chooses a random number $b \in Z^*_q$, computes $X_S = (v)^b \oplus v$, $K_S = (X_A \oplus v)^b = g^{ab}$, $V'_A = h(A, X_S, K_S)$ and $V_S = h(ID_S, X_A, K_A)$ and then sends X_S and V_S to Alice. After receiving X_S from Server, Alice computes $K_A = (X_S \oplus v)^{a \cdot h(ID_A, ID_S, password)^{-1}} = g^{ab}$, $VA = h(ID_A, X_S, K_A)$ and $V_S = h(ID_S, X_A, K_A)$ and then sends V_A to Server. Upon receiving from Alice, Server verifies whether $V_A = V'_A$. If it holds, Server authenticates Alice and computes the common session key $K = h(K_A) = h(g^{ab})$. Similarly, after receiving from Server, Alice verifies $V_S = B'_S$. If it holds, Alice authenticates Server and computes the common session key $K = h(K_S) = h(g^{ab})$.

Weakness: Lee *et al.*'s claimed that their protocol was secure in the case of server compromise. However (Shim and Seo, 2005) pointed out that the protocol was vulnerable to Stolen Verifier attack. In other words, given the Verifier, the attacker can impersonate a legitimate client to negotiate a session key with the Server. The flaw of the protocol is that Serve hasn't an efficient means to verify the message claimed to be sent by Alice. Motivated by Lee *et al.*'s protocol, we present a Verifier-based authenticated two-party protocol, which withstands Stolen Verifier attack.

THE PROPOSED KEY AGREEMENT PROTOCOL

Let G_1 and G_2 be two groups that support a bilinear map. Define a strong one-way function $h: \{0, 1\}^* \rightarrow Z^*_q$. We assume that there exists a initialization in which the user Alice chooses a password, computes $v = g^{h(ID_A, ID_S, password)}$ and then sends v to the Server as the Verifier. The proposed protocol consists of the following steps.

Step 1: Alice chooses a random number $a \in Z^*_q$ and computes $X_{A1} = g^a$ and then sends it to Server.

Step 2: Server chooses a random umber $b \in Z^*_q$ and computes $X_{S1} = v^b$ and then sends it to Alice.

Step 3: Upon receiving the message from Server, Alice computes $\alpha = (X_{S1})^{ah(ID_A, ID_S, password)^{-1}}$ and $X_{A2} = h(\alpha)$ and then sends X_{A2} to Server.

Step 4: After receiving the message from Alice, Server computes, $V_A = h(X_{A1})^b)$ and verifies whether $V_A = X_{A2}$. If it holds, Server authenticates Alice and computes $X_{S2} = v^{b^2}$ and then sends it to Alice. Thereafter, Server computes the session key $K = h(ID_A, ID_S, X^b_{A1})$.

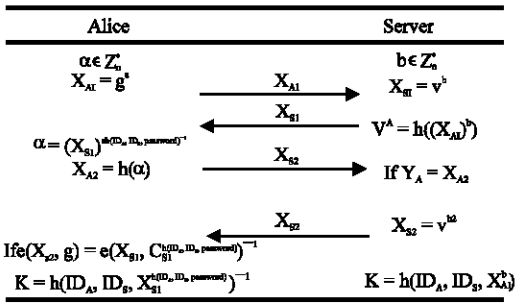

Fig. 3: The proposed protocol

Step 5: After receiving the message from Server, Alice verifies whether $e(X_{S2}, g) = e(X_{S1}, X_{S1}^{h(ID_A, ID_S, password)^{-1}})$. If it holds, Alice authenticates Server and computes the common session key $K = h(ID_A, ID_S, X_{S1}^{ah(ID_A, ID_S, password)^{-1}})$.

After successfully implementing above protocol, the two parties, Alice and Server, will agree upon a common session key $K = h(ID_A, ID_S, g^{ab})$. The protocol can be illustrated in Fig. 3.

SECURITY ANALYSIS

Withstand stolen verifier attack: We assume that an attacker, Eve, has intruded Server and gotten the Verifier. As we have mentioned above, Eve's aim is to impersonate Alice to negotiate a session key with Server. We have the following theorem.

Theorem: Suppose CDH assumption holds and then our key agreement protocol is secure against Stolen Verifier attack.

Proof: In this scenario, Eve is allowed to choose a random number $a \in Z^*_n$ and compute $X_{A1} = g^a$. We suppose that the attacker, Eve, has ability to impersonate Alice. In other words, Eve should output two values X_{A1} and X_{A2} which satisfy $V_A = X_{A2}$.

Since h is a strong one way function, given g^t and g, Eve should compute g^b and then use this value to compute X_{A2}, where Verifier v is denoted by g^t. Obviously, it is contrary to our complexity assumptions described earlier.

There exists alternative way for Eve to pose as Alice. Eve can collect messages g^t, g^{bt} and g^{b^2t} and then try to output. However, by the assumptions mentioned earlier, it is intractable.

With above mentioned, we can say that an attacker can't pose as Alice even though he obtains the Verifier stored in Server and tries to make Stolen Verifier attack.

Withstand dictionary attack: To the on-line password guessing attack, the participants can defeat the attacker by choosing appropriate trail intervals. In an off-line guessing attack, the attacker should repeatedly guess the password and verify its correctness by the message gathered in an off-line manner. In our protocol, the attacker is allowed to collect any message exchanged over network. It means that the attacker can obtain g^a, g^{bt}, $h(g^{ab})$ and g^{b^2t}. Since a, $b \in Z^*_n$ are random numbers uniformly distributed in Z^*_n, the off-line Dictionary attack is defeated. In addition, given g^{bt} and g^{b^2t}, an attacker can't output g^b by our assumptions. Therefore, we say that the proposed protocol is secure against Dictionary attack.

Withstand man-in-middle attack: The pre-shared password and Verifier are used to prevent the man-in-middle attack. Since an attacker doesn't have the password or Verifier, he can't pretend Alice to exchange information with Bob.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewer for his valuable comments.

CONCLUSIONS

Using password to design authenticated key agreement protocols has attracted more and more attention in recently few years. In practice, password-based schemes are suitable for implementation in many scenarios, especially those where no device is capable of securely storing high-entropy long-term secret key. As we have mentioned, since password has limited entropy and is susceptible to dictionary attack, man-in-middle attack and so on, one should be careful in designing password-based protocols.

REFERENCES

Abdalla, M., O. Chevassut and D. Pointcheval, 2005. One-time verifier-based encrypted key exchange. In PKC'05, LNCS, 3386: 47-64.

Boneh, D., B. Lynn and H. Shacham, 2001. Short signatures from the Weil pairing. Advances in Cryptology-Asiacrypt'2001, Gold Coast, Australia. Lecture Notes in Computer Sci., 2248: 514-532.

Diffe, W. and M. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theory, 22: 644-654.

Kim, Y.S., E.N. Huh, J.H. Hwang and B.W. Lee, 2004. An efficient key agreement protocol for secure authentication. ICCSA 2004. LNCS, 3043: 746-754.

Kwon, T., 2004. Practical Authenticated Key Agreement Using Passwords. ISC 2004, LNCS, 3255: 1-12.

Lee, S.W., W.H. Kim, H.S. Kim and K.Y. Yoo, 2004. Efficient password-based authenticated key agreement protocol. In ICCSA. LNCS, 3046: 617-626

Lomas, T., L. Gong, J. Saltzer and R. Needham, 1989. Reducing risks from poorly chosen keys. ACM SIGOPS Operat. Syst. Rev., 23: 14-18.

Ryu, E.K., K.W. Kim and K.Y. Yoo, 2004. An authenticated key agreement protocol resistant to a dictionary attack. In: ICCSA 2004. LNCS, 3046: 603-610.

Seo, D.H. and P. Sweeney, 1999. Simple authenticated key agreement algorithm. Elect. Lett., 35: 1073-1074.

Shim, K.A. and S.H. Seo, 2005. Security analysis of password authenticated key agreement protocols. In CANS 2005, LNCS, 3810: 49-58

Yoon E.J. and K.Y. Yoo, 2005. New efficient simple authenticated key agreement protocol. COCOON 2005, LNCS, 3595: 945-954.

Zhang, F., R. Safavi-Naini and W. Susilo, 2004. An efficient signature scheme from bilinear pairings and its applications. Practice and Theory in Public Key Cryptography-PKC 2004, LNCS, 2947: 277-290.