An Optimized Two Factor Authenticated Key Exchange Protocol in PWLANs

Eun-Jun Yoon and Kee-Young Yoo*

Department of Computer Engineering, Kyungpook National University,
Daegu 702-701, Republic of Korea
Tel.: +82-53-950-5553; Fax: +82-53-957-4846
ejyoon@infosec.knu.ac.kr, yook@knu.ac.kr

Abstract. Recently, Park and Park proposed a new two factor authenticated key exchange (PP-TAKE) protocol that can be applied to low-power PDAs in Public Wireless LANs (PWLANs). The current paper proposes an efficient TAKE protocol based upon the PP-TAKE protocol. The computational cost of the proposed TAKE protocol is less than that of the PP-TAKE protocol, as is the number of steps needed to communicate is one fewer in which needs only three steps.

Keywords: Authentication, Wireless security, Key exchange, Optimized.

1 Introduction

Two factor authentication refers to the authentication of an entity by identifying (1) what a user remembers (the password) and (2) what the user has (a token or wireless terminal) in an integrated fashion. Two factor authentication requires a token as the second factor and a token-reading input device (a token reader). In generally, a token might be a smart card, a USB (Universal Serial Bus)-based smart key, or a wireless device. If a wireless device or USB-based smart key is used as a token in a PWLAN environment [1], no token reader is required. As the token stores within it the user's secret key or certificate data, however, it should be stored in a safe module that provides a certain level of tamper resistance.

In 2004, Park and Park [2] proposed a new mutual authentication and key establishment (PP-TAKE) protocol that can be applied to low-power PDAs in Public Wireless LANs (PWLANs), by using two factor authentication and precomputation. The PP-TAKE protocol provides mutual authentication, identity privacy, and half forward-secrecy. Also, the complex computations that the client must perform include only one symmetric key encryption and five hash functions during the protocol runtime. The number of communication steps in the PP-TAKE protocol, however, has clearly increased. A protocol has two important efficiency metrics - the number of steps and the number of rounds. One step involves the sending of data items from one party to a single destination at one time. A round includes all independent steps that can be sent and received

[*] Corresponding author.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 1000–1007, 2006.

in parallel. Thus, a participant can simultaneously send different messages to different destinations in a single round, and so multiple participants can send messages in a single round [3][4]. Accordingly, many protocols can be executed in less time by rearranging and sending the messages in parallel; such protocols are round efficient versions. The current paper proposes a modified PP-TAKE protocol in which the number of communication steps in the protocol is reduced by one and the computational cost is low. The computational cost of the proposed TAKE protocol is less than that of the PP-TAKE protocol, and the number of steps regarding communication is one fewer, only three steps are needed. Therefore, the proposed TAKE protocol is more efficient than the PP-TAKE protocol, which can be applied to low-power PDAs in Public Wireless LANs (PWLANs).

This paper is organized as follows: In Section 2, we briefly review the PP-TAKE Protocol. In Section 3, we propose an optimized TAKE protocol. In Sections 4 and 5, we analyze the security and efficiency of our proposed TAKE protocol, respectively. Finally, the conclusion is given in Section 6.

2 The Review of the PP-TAKE Protocol

This section briefly reviews Park-Park's TAKE (PP-TAKE) protocol. Some of the notations used in this paper are defined as follows:

- A, B : client (supplicant) and authentication server (AS)
- p : password
- t : symmetric key used in symmetric key encryption
- ID_A : client A's identifier
- $E_K\{\}$, $D_K\{\}$: encrypt and decrypt with symmetric key K
- $H()$: a cryptographic hash function
- sk_A : session key generated by client A
- p, q : a large prime number and prime divisor of $(p-1)$
- g : an element of order q in Z_p^*
- b, g^b : static private key and public key of B
- $\oplus$: a bitwise exclusive or operation

The TAKE protocol is based on DH (Diffie-Hellman) key agreement and can be modified to work in an arbitrary finite group. In terms of phases, the TAKE protocol includes enrollment phase, a precomputation phase and a real-execution phase. Fig. 1 shows the real-execution phase of the PP-TAKE protocol. For simplicity, the operator mod$p$ will be omitted.

The Enrollment Phase: The client and the Authentication Server (AS) determine and share the password $\langle\pi\rangle$ and symmetric key $\langle t\rangle$, that are used for symmetrical algorithms such as 3DES or Rijndael. The AS selects a random number $\langle b\rangle$ from numbers ranging between $[1\ (q-1)]$ as its private key to a specific client, and the selected number is stored in a secure database. The client is informed of AS's public key $\langle g^b\rangle$ and domain parameters $\langle p, q, g\rangle$. The client stores the symmetric key $\langle t\rangle$ in a secure token. Since the AS's public key $\langle g^b\rangle$ and domain parameters $\langle p, q, g\rangle$ can be made public, they do not have to be stored in a secure location.

Client A $(\langle\pi\rangle, \langle t\rangle)$		Authentication Server B $(\langle\pi\rangle, \langle t\rangle, \langle b\rangle)$
Precomputation $\begin{bmatrix} x \in_R Z_q \\ g^x, c = g^{bx} \end{bmatrix}$		
	$\xrightarrow{H(ID_A, g^b)}$	
$f = H(\pi, t, ID_A)$	$\xleftarrow{r}$	$r \in_R Z_q$
$e = E_f\{g^x\}$ $sk_A = H(c, g^x, r, ID_A)$ $M_A = H(sk_A, \pi, t, g^b)$	$\xrightarrow{e, M_A}$	$f = H(\pi, t, ID_A)$ $g^x = D_f\{e\}$ $c = g^{xb}$ $sk_B = H(c, g^x, r, ID_A)$ $M_A \overset{?}{=} H(sk_B, \pi, t, g^b)$
$M_B \overset{?}{=} H(sk_A, \pi, t, ID_A)$	$\xleftarrow{M_B}$	$M_B = H(sk_B, \pi, t, ID_A)$

Session key $sk_A = sk_B = H(c, g^x, r, ID_A)$

Fig. 1. The PP-TAKE protocol

The Precomputation Phase: Precomputation is performed off-line prior to the execution of the protocol. It reduces time and the computational load during the protocol execution. The client's wireless device performs precomputation during idle time or at the time of power on. In order to be more specific, a random number $\langle x\rangle$ is selected from $[1 \sim (q-1)]$, and then g^x and $c = g^{bx}$ are calculated off-line prior to the protocol execution.

The Real-Execution Phase: The real-execution phase performs mutual entity authentication and session key establishment and it consists of the following four steps:

Step 1. $A \rightarrow B$: $H(ID_A, g^b)$
In order to connect to PWLANs service, client (A) sends to $AS(B)$ its identifier ID_A and AS's public key $\langle g^b\rangle$, which has been hashed into $H(ID_A, g^b)$. If the client identifier uses a NAI (Network Access ID) to support global roaming and accounting (ex: userid@realm.com), the user name portion and the g^b are hashed into $H(userid, g^b)$ and the realm portion are sent together as well.

Step 2. $B \rightarrow A$: r
Upon receipt of $H(ID_A, g^b)$, B extracts $\langle H(ID_A, g^b)\rangle$, $\langle ID_A\rangle$, $\langle\pi\rangle$, $\langle t\rangle$, $\langle b\rangle$ from its database. B selects a random number $r \in_R Z_q^*$ and sends it to A.

Step 3. $A \rightarrow B$: e, M_A
Upon receipt of $\langle r\rangle$, A computes $f = H(r, \pi, t)$ and $e = E_f\{g^x\}$, using f as a symmetric key for the encryption of g^x. A then computes the session key $sk_A = H(c, g^x, r, ID_A)$ and generates $M_A = H(sk_A, \pi, t, g^b)$. A sends $\langle e\rangle$ and $\langle M_A\rangle$ to B.

Step 4. $B \rightarrow A$: M_B
Upon reception of $\langle e\rangle$ and $\langle M_A\rangle$, B computes $f = H(r, \pi, t)$ and $g^x = D_f\{e\}$, using $\langle f\rangle$ as a symmetric key for the decryption of $\langle e\rangle$. Then

B computes $c = g^{xb}$ and $sk_B = H(c, g^x, r, ID_A)$ and confirms if $\langle M_A \rangle$, received from A and $H(sk_B, \pi, t, g^b)$ computed by B are identical. If they are identical, A's authentication is successful and B accepts $\langle M_A \rangle$. B then computes $M_B = H(sk_B, \pi, t, ID_A)$ and sends it to A. A checks to see if $\langle M_B \rangle$, that has been received from B and $H(sk_A, \pi, t, ID_A)$, as computed by A, are identical. If they are identical, B's authentication is successful and A accepts $\langle M_B \rangle$. If A and B accept $\langle M_B \rangle$ and $\langle M_A \rangle$ respectively, mutual authentication is successful.

3 Optimized TAKE Protocol

This section proposes an optimized TAKE protocol. An enrollment phase and a precomputation phase are equal to the PP-TAKE protocol. Fig. 2 shows a real-execution phase of the proposed optimized TAKE protocol. Unlike the PP-TAKE protocol, the proposed protocol does not need a symmetric encryption/decryption operation. The proposed protocol uses bitwise exclusive or ($\oplus$) operation for the protection of the precomputed value g^x. The proposed real-execution phase requires only three steps. They are as follows:

The Real-Execution Phase: The real-execution phase performs mutual entity authentication and session key establishment and it consists of the following steps:

Step 1. $A \rightarrow B$: $H(ID_A, g^b), e$

In order to connect to the PWLANs service, client (A) computes $f = H(\pi, t, ID_A)$ and $e = f \oplus g^x$, and sends $H(ID_A, g^b)$ and e to $AS(B)$. If the client identifier uses a NAI (Network Access ID) to support global roaming and accounting (ex: userid@realm.com), the user name portion and the g^b hashed into $H(userid, g^b)$ and the realm portion are sent together as well.

Step 2. $B \rightarrow A$: M_B, r

Upon receipt of $H(ID_A, g^b)$ and e, B extracts $\langle H(ID_A, g^b) \rangle$, $\langle ID_A \rangle$, $\langle \pi \rangle$, $\langle t \rangle$, $\langle b \rangle$ from its database. B selects a random number $r \in_R Z_q^*$ and computes $f = H(\pi, t, ID_A)$ and $g^x = e \oplus f$. B then computes $c = g^{xb}$, $sk_B = H(c, g^x, r, ID_A)$ and $M_B = H(sk_B, \pi, t, ID_A)$, and sends M_B and r to A.

Step 3. $A \rightarrow B$: M_A

Upon receipt of M_B and r, A then computes session key $sk_A = H(c, g^x, r, ID_A)$ and checks to see if $\langle M_B \rangle$, that has been received from B and $H(sk_A, \pi, t, ID_A)$, as computed by A, are identical. If they are identical, B's authentication is successful and A accepts $\langle M_B \rangle$. A then generates $M_A = H(sk_A, \pi, t, g^b)$ and sends it to B. Upon reception of $\langle M_A \rangle$, B confirms if $\langle M_A \rangle$ received from A and $H(sk_B, \pi, t, g^b)$, as computed by B, are identical. If they are identical, A's authentication is successful and B accepts $\langle M_A \rangle$. If A and B accept $\langle M_B \rangle$ and $\langle M_A \rangle$ respectively, mutual authentication is successful.

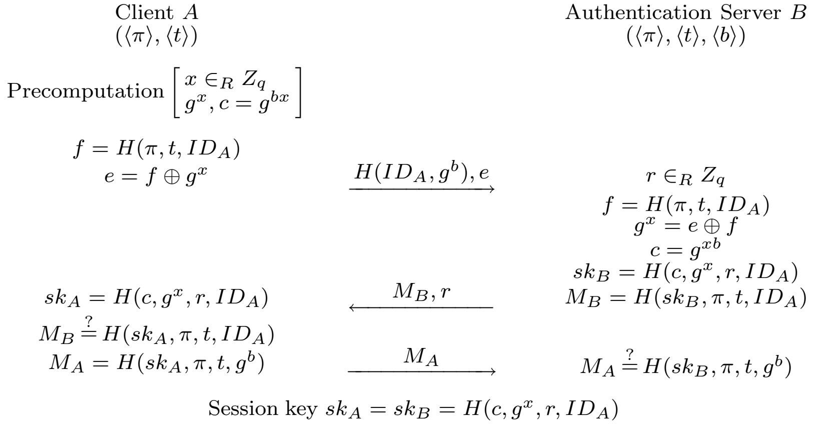
Client A
$(\langle\pi\rangle, \langle t\rangle)$
Authentication Server B
$(\langle\pi\rangle, \langle t\rangle, \langle b\rangle)$
Precomputation $\begin{bmatrix} x \in_R Z_q \\ g^x, c = g^{bx} \end{bmatrix}$
$f = H(\pi, t, ID_A)$
$e = f \oplus g^x$
$\xrightarrow{H(ID_A, g^b), e}$
$r \in_R Z_q$
$f = H(\pi, t, ID_A)$
$g^x = e \oplus f$
$c = g^{xb}$
$sk_B = H(c, g^x, r, ID_A)$
$M_B = H(sk_B, \pi, t, ID_A)$
$\xleftarrow{M_B, r}$
$sk_A = H(c, g^x, r, ID_A)$
$M_B \overset{?}{=} H(sk_A, \pi, t, ID_A)$
$M_A = H(sk_A, \pi, t, g^b)$
$\xrightarrow{M_A}$
$M_A \overset{?}{=} H(sk_B, \pi, t, g^b)$
Session key $sk_A = sk_B = H(c, g^x, r, ID_A)$

Fig. 2. The optimized TAKE protocol

After Step 3, A and B have had it validated that the common secret session key $sk_A = sk_B = H(c, g^x, r, ID_A)$.

4 Security Analysis

This section analyzes the security of the proposed TAKE protocol. Here, the following security properties must be considered in the proposed [2]: identity protection, explicit mutual authentication, session key establishment, forward-secrecy, resistance to an off-line dictionary attack, key confirmation, and non-repudiation. The followings are used to analyze the security properties in the proposed protocol.

(1) *The proposed TAKE protocol provides identity protection:* To ensure the privacy of personal communication, it is necessary to protect a client's identity from passive attacks such as eavesdropping. Also, identity protection is particularly useful for the client to whom a dynamic IP address is allocated by the DHCP. In Step 1 of the proposed TAKE protocol, upon receiving an ID request from the AP, the client sends $H(ID_A, g^b)$ instead of its real identity ID_A, to prevent passive attackers such as eavesdroppers, from knowing the client's identity. The AS, however, needs to be able to match the pseudonym of the client to its real identity ID_A.

(2) *The proposed TAKE protocol provides explicit mutual authentication:* Since attackers can launch Man-in-the-Middle (MitM) attacks by installing a rouge AP or a rouge radio NIC between the client and the Authentication Server (AS), explicit mutual authentication between the client and the network is necessary to prevent MitM attacks. The client must know the password $\langle\pi\rangle$, the symmetric key $\langle t\rangle$, the session key $\langle sk_A\rangle$, and the AS's public key $\langle g^b\rangle$ in order to compute the M_A for client authentication, while the AS must know the password $\langle\pi\rangle$, the symmetric key $\langle t\rangle$, the session key $\langle sk_B\rangle$, and its private key $\langle b\rangle$ in order to compute the M_B for AS authentication.

The session key is explicitly authenticated by mutual confirmation values M_A and M_B, respectively. Therefore, the proposed TAKE protocol provides explicit mutual authentication.

(3) *The proposed TAKE protocol provides session key establishment:* The session key must be established between the client and the AS, so that it can support the dynamic WEP key. For a random challenge, random numbers $\langle x \rangle$ and $\langle r \rangle$, which are separately generated by each entity, are different every time. Therefore, the established session keys $\langle sk_A \rangle$ and $\langle sk_B \rangle$ are freshness and randomness.

(4) *The proposed TAKE protocol can provide full forward-secrecy:* Forward-secrecy should be provided to ensure that attackers cannot compute previous session keys from the sessions which were eavesdropped on previously, even when a long-term secret keying material of the entity participating in the protocol has been revealed. If $\langle ID_A \rangle$, $\langle \pi \rangle$, $\langle t \rangle$, and $\langle g^b \rangle$, possessed by a client, are all exposed to an attacker, the attacker may learn about g^x by computing $e \oplus f$. The value $c = g^{xb}$, however, is hard to compute due to the DDH (Decision Diffie-Hellman) problem [5]. Therefore, forward-secrecy on the client's side is ensured. On the other hand, if $\langle b \rangle$, $\langle \pi \rangle$, and $\langle t \rangle$, which are stored on the AS side, and the $\langle ID_A \rangle$ of a client are all exposed to an attacker, the attacker can compute a session key. Therefore, forward-secrecy on the AS side is not provided. In order to provide full forward secrecy, the Diffie-Hellman key exchange algorithm can be used to compute the session key. In this approach, we let $c = g^{xr}$, where x and r are the random exponents chosen by the client and AS separately, and $sk_A = sk_B = H(c, g^x, r, ID_A, g^b)$. Then, the protocol can provide full forward secrecy. The client, however, is required to perform, in addition, one exponentiation operation in a real-execution phase of the protocol as a trade-off.

(5) *The proposed TAKE protocol can resist an off-line dictionary attack:* It should be guaranteed that secret information (passwords and keys) shared between a client and the AS is resistant to an off-line dictionary attack. As the symmetric key $\langle t \rangle$ with high entropy, the password $\langle \pi \rangle$, and a random number $\langle r \rangle$ are hashed into $\langle f = H(r, \pi, t) \rangle$ and are used as a key for the protection of g^x, therefore, off-line dictionary attacks stand little chance of success. In other words, an attacker cannot help but guess the password $\langle \pi \rangle$, symmetric key $\langle t \rangle$ and random value $\langle g^x \rangle$ at the same time.

(6) *The proposed TAKE protocol provides key confirmation:* It should be confirmed to a legitimate user participating in the protocol that he or she actually shares a common secret session key with an entity with which communication is intended. The proposed TAKE protocol includes the session key in M_A and M_B, in order to confirm the keys.

(7) *The proposed TAKE protocol can provide non-repudiation:* Fraudulent clients must know that the accounting is correct but that illegal access is not paid for. The proposed TAKE protocol doesn't employ a digital signature, so it doesn't support non-repudiation which provides proof of the integrity and origin of data. Using the two factor authentication, however, makes it more

difficult for fraudulent clients to deny the use of the PWLANs service than using the single factor authentication.

5 Efficiency Analysis

The computational costs of the PP-TAKE protocol and the proposed TAKE protocol, in the precomputation and real-execution phases, are summarized in Table 1. The followings are used to analyze the efficiency properties in the proposed protocol.

(1) *The proposed TAKE protocol uses a low computational load:* In general, a protocol requires a low computational load that can be borne by even low-power devices such as PDAs and precomputation, in order to minimize on-line computation operations. In the precomputation phase of the PP-TAKE protocol and the proposed TAKE protocol, the client is required to perform two exponentiation operations. In a real-execution phase, PP-TAKE requires a one-time exponentiation operation, a one-time symmetric encryption operation, a one-time symmetric decryption operation and a nine-time hash operation, but the proposed TAKE protocol does not need symmetric encryption/decryption operations. It requires a total of one exponentiation operation, nine hash operations and two bitwise exclusive-or operations. On the client side of the PP-TAKE protocol, the computational load is one symmetric key encryption, and five hashes, but the computational load of the proposed TAKE protocol is one bitwise exclusive-or, and four hashes.

(2) *The proposed TAKE protocol uses a minimum number of message exchanges:* In terms of network resource efficiency and network delay, it is advantageous to have as few communication rounds as possible. Therefore, the number of messages that are exchanged between client and AS should be kept to a minimum. The PP-TAKE protocol requires four steps in order to perform mutual authentication and key establishment, while the proposed TAKE protocol requires only three steps.

(3) *The proposed TAKE protocol uses a minimum communication bandwidth:* The protocol message should be as short as possible. Among the five messages, three are hash output bits, one is random number bits and the other is the bitwise exclusive or encryption output bit of g^x.

Table 1. A comparisons of the computational costs

	PP-TAKE Protocol		Proposed TAKE Protocol	
	Client	Server	Client	Server
Precomputation Phase	2Exp	·	2Exp	·
Real-Execution Phase	1Sym + 5Hash	1Exp + 1Sym + 4Hash	5Hash + 1Xor	1Exp + 4Hash + 1Xor
# of Steps	4 Steps		3 Steps	

Exp: Exponentiation operation; Sym: Symmetric key encryption/decryption operation; Hash: Cryptographic hash operation; Xor: Bitwise exclusive-or (⊕) operation.

As shown in the efficiency properties (1)-(3), it is obvious that our protocol is more efficient than the PP-TAKE protocol for both the client and server, respectively.

6 Conclusions

The current paper proposed an efficient TAKE protocol, which is based upon the PP-TAKE protocol. The computational cost of the proposed TAKE protocol is less than that of the PP-TAKE protocol, and the number of steps in the communication process is one fewer that what is normally needed (only three steps are required). Furthermore, the security requirements of the proposed TAKE protocol are the same as the original PP-TAKE protocol, as described in literature [2]. Therefore, the proposed TAKE protocol is more efficient than the PP-TAKE protocol.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. IEEE.: Standard for local and metropolitan area networks-Port based network access control. IEEE Std 802.1x. (June 2001)
2. Park, Y.M., Park, S.K.: Two factor authenticated key exchange (TAKE) protocol in public wireless LANs. IEICE Trans. Commun., Vol. E87-B, No. 5. (May 2004) 1382-1385
3. Gong, L.: Efficient network authentication protocols: lower bounds and implementations. Distrib Comput. Vol. 9. No. 3. (1995) 131-145
4. Lee, T.F., Hwang, T., Lin, C.L.: Enhanced three-party encrypted key exchange without server public keys. Computers & Security. Vol. 23. (2004) 571-577
5. Boneh, D.: The decision Diffie-Hellman problem. Proc. Third Algorithmic Number Theory Symposium. (1998) 48-63