Understanding Server-Compromise Impersonation Attacks in Augmented PAKE

辛　星漢 †　　古原　和邦 †　　今井　秀樹 †‡

†（独）産業技術総合研究所情報セキュリティ研究センター
〒 101-0021 東京都千代田区外神田 1-18-13

seonghan.shin@aist.go.jp
‡ 中央大学理工学部電気電子情報通信工学科
〒 112-8551 東京都文京区春日 1-13-27

あらまし　Augmented PAKE (Password-Authenticated Key Exchange) プロトコルはパスワードのみで認証付き鍵共有をしながらさらに Server-Compromise Impersonation (SCI) 攻撃にも安全性を有するものである。現在、複数の augmented PAKE プロトコルが ISO/IEC JTC1/SC 27 11770-4 で標準化され、IEEE P1363.2 working group でも標準化が検討されている。本稿では、Server-Compromise Impersonation (SCI) 攻撃を再考察することで二つの augmented PAKE プロトコルが実は SCI 攻撃に安全ではないことを示す。

Understanding Server-Compromise Impersonation Attacks in Augmented PAKE

SeongHan Shin†　　Kazukuni Kobara†　　Hideki Imai†‡

†Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science and Technology (AIST)
1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan

seonghan.shin@aist.go.jp
‡Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

Abstract　An augmented PAKE (Password-Authenticated Key Exchange) protocol is said to be secure against server-compromise impersonation attacks if an attacker who obtained client's password verification data from a server cannot impersonate the client without performing off-line dictionary attacks on the password verification data. Until now, several augmented PAKE protocols have been standardized in ISO/IEC JTC1/SC 27 11770-4 and are being standardized in IEEE P1363.2 working group. In this paper, we revisit server-compromise impersonation attacks by showing that two augmented PAKE protocols (claimed to be secure) are actually insecure against server-compromise impersonation attacks. More specifically, we present generic server-compromise impersonation attacks on the two augmented PAKE protocols.

1 Introduction

Since the appearance of [1, 2], PAKE (Password-Authenticated Key Exchange) protocols have been designed to provide password-only authentication and establishment of temporal session keys secure against active attacks as well as off-line dictionary attacks on passwords. Several PAKE protocols have been standardized in ISO/IEC JTC 1/SC 27 11770-4 [7] and are being standardized in IEEE P1363.2 working group [6] (current draft version is D26).

In general, PAKE protocols can be classified into 'balanced' PAKE and 'augmented' PAKE [6, 7]: in the former case a client and a server share a common password; and in the latter case a client remembers his/her password and a server has password verification data (derived by applying a one-way function to the password). Since password verification data has the same entropy of the password, the off-line dictionary attacks are inevitable if server is compromised. Nonetheless, an augmented PAKE protocol may be preferable because it provides extra protection for server compromise. That is, the ultimate goal in improving resistance to server compromise is to make the off-line dictionary attacks the best one an attacker can do. According to [4],

Definition 1.1 *An augmented PAKE protocol is said to be* secure against server-compromise impersonation attacks *if an attacker who obtained the password verification data must at least perform an off-line dictionary attack to gain any advantage in impersonating the client.*

Actually, there has been a significant amount of works on augmented PAKE protocols. See [6] for some augmented PAKE protocols (e.g., B-SPEKE, AuthA, PAK-X, PAK-Y, PAK-Z, PAK-Z+, SRP and AMP).

1.1 Our Contributions

There are two augmented PAKE protocols where the first one [8] was proposed in the IEEE Communications Letters and the second one [9] was submitted to the IEEE P1363.2 standard working group [6]. In this paper, we show that these two augmented PAKE protocols [8, 9] (claimed to be secure) are actually insecure against server-compromise impersonation attacks. More specifically, we present *generic* server-compromise impersonation attacks on these augmented PAKE protocols [8, 9].

1.2 Notation

Here, we explain some notation to be used throughout this paper. Let $\mathbb{G}$ be a finite, cyclic subgroup of prime order q of the multiplicative group $\mathbb{Z}_p^\star$ where $p = aq + 1$ is a prime and a is an integer. Let g be a generator of $\mathbb{G}$ where the group operation is denoted multiplicatively. These parameters (p, q, g) are public to everyone. In the aftermath, all the subsequent arithmetic operations are performed in modulo p unless otherwise stated. Let k be the security parameter for hash functions. Let $\{0,1\}^\star$ denote the set of finite binary strings and $\{0,1\}^k$ the set of binary strings of length k. Let $A\|B$ be the concatenation of bit strings of A and B in $\{0,1\}^\star$. If D is a set, then $d \stackrel{R}{\leftarrow} D$ indicates the process of selecting d at random and uniformly over D. We use two different hash functions $\mathcal{H}$ and $\mathcal{H}_j$, for $j = 1, 2, 3$, where $\mathcal{H} : \{0,1\}^\star \rightarrow \mathbb{Z}_q^\star$ and $\mathcal{H}_j : \{0,1\}^\star \rightarrow \{0,1\}^k$. The $\mathcal{H}$ and $\mathcal{H}_j$ are implemented with secure one-way hash functions (e.g., SHA-2 family). Let C and S be the identities of client and server, respectively, with each identity $\mathrm{ID} \in \{0,1\}^\star$.

2 A Generic Server-Compromise Impersonation Attack on Improved EPA

In [8], Kwon et al., showed that the EPA protocol [5] does not provide resistance to server compromise. Then, they proposed an improved EPA protocol [8] and claimed that it is secure against server-compromise impersonation attacks. In this section, we prove that their claim is completely wrong by showing that the improved EPA protocol [8] is insecure against a generic server-compromise impersonation attack.

2.1 The Improved EPA Protocol

First, we explain the improved EPA protocol [8]. Let h be another generator of $\mathbb{G}$ such that its discrete logarithm problem with g should be hard (of course, h is a public parameter).

In the initialization phase, client C registers his/her password verification data ($W \equiv g^{pwb}$, $Z \equiv h^{pwa}$) securely to server S where $pwa = \mathcal{H}(C, pw, 1)$, $pwb = \mathcal{H}(C, pw, 2)$ and pw is the client's password. Then, client C runs the below improved EPA protocol with server S over insecure networks.

Step 1: The client C chooses a random element $x \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and computes $X \equiv g^x \cdot h^{pwa}$ where $pwa = \mathcal{H}(C, pw, 1)$. Then, client C sends the first message (C, X) to server S.

$$C \rightarrow S : (C, X)$$

Step 2: After receiving (C, X), server S chooses a random element $y \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$, and computes $Y \equiv ((X/Z) \cdot W)^y$, $K \equiv ((X/Z) \cdot g)^y$ and $V_S = \mathcal{H}_1(X \| K)$. Then, server S sends the second message (S, Y, V_S) to client C.

$$S \rightarrow C : (S, Y, V_S)$$

Step 3: After receiving (S, Y, V_S), client C computes $K' \equiv Y^{(x+1)/(x+pwb)}$, where $pwb = \mathcal{H}(C, pw, 2)$, and $V_C = \mathcal{H}_2(Y \| K')$. If the received V_S is correct (i.e., $V_S = \mathcal{H}_1(X \| K')$), client C sends the third message V_C to server S. Otherwise, the client terminates the protocol. Finally, client C computes a session key $SK = \mathcal{H}_3(C \| K')$.

$$C \rightarrow S : V_C$$

Step 4: If the received V_C is correct (i.e., $V_C = \mathcal{H}_2(Y \| K)$, server S computes a session key $SK = \mathcal{H}_3(C \| K)$. Otherwise, it terminates the protocol.

Note that the difference of the improved EPA protocol [8] from the original one [5] is the computation of the shared key K and K' in **Step 2** and **Step 3**.

2.2 The Attack

Here, we show that the above improved EPA protocol [8] is insecure against a generic server-compromise impersonation attack.

Theorem 2.1 *The improved EPA protocol [8] is not an augmented PAKE protocol because it is insecure against server-compromise impersonation attacks.*

We prove this theorem by showing the attack below. In the server-compromise impersonation attacks, an attacker $\mathcal{A}$ is trying to impersonate client C with the obtained password verification data (W, Z) itself, but without doing off-line dictionary attacks on (W, Z). The generic server-compromise impersonation attack of $\mathcal{A}$ is as follows:

Step 1': The attacker $\mathcal{A}$ chooses two random elements $(\alpha, \beta) \stackrel{R}{\leftarrow} (\mathbb{Z}_q^\star)^2$ and computes $X \equiv g^\alpha \cdot W^\beta \cdot Z$. Then, attacker $\mathcal{A}$ sends a message (C, X) to server S.

$$\mathcal{A} \rightarrow S : (C, X)$$

Step 2': After receiving (C, X), server S chooses a random element $y \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$, and computes $Y \equiv$

$((X/Z)\cdot W)^y$, $K \equiv ((X/Z)\cdot g)^y$ and $V_S = \mathcal{H}_1(X\|K)$ as usual. Then, server S sends a message (S, Y, V_S) to client C (actually, to attacker $\mathcal{A}$).

$$S \rightarrow \mathcal{A} : (S, Y, V_S)$$

In order to impersonate client C successfully, attacker $\mathcal{A}$ should find out γ satisfying $Y^\gamma \equiv K$. That is,

$$\begin{aligned} \log_g Y^\gamma &\equiv \log_g K \bmod q \\ (\alpha + pwb\cdot\beta + pwb)y\cdot\gamma &\equiv (\alpha + pwb\cdot\beta + 1)y \\ (\alpha + pwb(\beta+1))\gamma &\equiv \alpha + 1 + pwb\cdot\beta \quad (1) \end{aligned}$$

Since there is no off-line dictionary attacks on (W, Z), the solution of Equation (1) is that $\alpha\cdot\gamma \equiv \alpha + 1 \bmod q$ and $(\beta+1)\gamma \equiv \beta \bmod q$. If the attacker $\mathcal{A}$ chooses (α, β), such that $\alpha + \beta \equiv -1 \bmod q$, the server-compromise impersonation attack is always possible. This means that, by sending $X \equiv g^{-1-\beta}\cdot W^\beta \cdot Z$ in **Step 1'**, the attacker $\mathcal{A}$ can impersonate client C successfully with $K' \equiv Y^{\beta/(\beta+1)} = K$. It can be easily verified that

$$\begin{aligned} K' &\equiv Y^{\beta/(\beta+1)} \equiv (((X/Z)\cdot W)^y)^{\frac{\beta}{\beta+1}} \\ &\equiv \left(g^{-1-\beta}\cdot W^\beta \cdot W\right)^{y\cdot\frac{\beta}{\beta+1}} \\ &\equiv \left(g^{-(\beta+1)}\cdot W^{\beta+1}\right)^{y\cdot\frac{\beta}{\beta+1}} \equiv \left(g^{-1}\cdot W\right)^{y\cdot\beta} \end{aligned}$$

and

$$\begin{aligned} K &\equiv ((X/Z)\cdot g)^y \equiv \left(g^{-1-\beta}\cdot W^\beta \cdot g\right)^y \\ &\equiv \left(g^{-\beta}\cdot W^\beta\right)^y \equiv \left(g^{-1}\cdot W\right)^{y\cdot\beta} \ . \end{aligned}$$

Note that this attack is valid for any element $\beta \in \mathbb{Z}_q^\star$.

3 A Generic Server-Compromise Impersonation Attack on TP-AMP

In [9], Kwon proposed a 3-pass augmented PAKE (called, TP-AMP) protocol and submitted it to the IEEE P1363.2 standard working group. In this section, we show that the TP-AMP protocol [9] is insecure against a generic server-compromise impersonation attack.

3.1 The TP-AMP Protocol

First, we explain the TP-AMP protocol [9] which is a combination of AMP3 (Fig. 3 of [9]) and PAK [3]. Let $\mathcal{G}$ be a full-domain hash (FDH) function $\mathcal{G} : \{0,1\}^\star \rightarrow \mathbb{G}$. As noted in [9], the $\mathcal{G}$ can be replaced with the hash-masking technique, introduced in [3], for the first message sent from client C to server S.

In the initialization phase, client C registers his/her password verification data ($W \equiv g^{pwb}$, $Z = (\mathcal{G}(C, pwb))^{-1}$) securely to server S where $pwb = \mathcal{H}(C, pw)$ and pw is the client's password. Then, client C runs the below TP-AMP protocol with server S over insecure networks.

Step 1: The client C chooses a random element $x \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ and computes $X \equiv g^x \cdot Z^{-1}$ where $Z^{-1} = \mathcal{G}(C, pwb)$ and $pwb = \mathcal{H}(C, pw)$. Then, client C sends the first message (C, X) to server S.

$$C \rightarrow S : (C, X)$$

Step 2: After receiving (C, X), server S chooses a random element $y \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$, and computes $Y \equiv (X\cdot Z\cdot W)^y$, $K \equiv (X\cdot Z\cdot g)^y$ and $V_S = \mathcal{H}_1(C\|S\|X\|Y\|K)$.[1] Then, server S sends the second message (S, Y, V_S) to client C.

$$S \rightarrow C : (S, Y, V_S)$$

Step 3: After receiving (S, Y, V_S), client C computes $K' \equiv Y^{(x+1)/(x+pwb)}$, where $pwb = \mathcal{H}(C, pw)$, and $V_C = \mathcal{H}_2(C\|S\|X\|Y\|K')$. If the received V_S is correct (i.e., $V_S = \mathcal{H}_1(C\|S\|X\|Y\|K')$), client C sends the third message V_C to server S. Otherwise, the client terminates the protocol. Finally, client C computes a session key $SK = \mathcal{H}_3(C\|S\|X\|Y\|K')$.

$$C \rightarrow S : V_C$$

[1] The TP-AMP protocol [9] does not work correctly unless $c \leftarrow m\nu \bmod p$ and $d \leftarrow mg \bmod p$ should be $c \leftarrow m'\nu \bmod p$ and $d \leftarrow m'g \bmod p$, respectively.

Step 4: If the received V_C is correct (i.e., $V_C = \mathcal{H}_2(C\|S\|X\|Y\|K)$), server S computes a session key $SK = \mathcal{H}_3(C\|S\|X\|Y\|K)$. Otherwise, it terminates the protocol.

3.2 The Attack

In this subsection, we show that the above TP-AMP protocol [9] is insecure against a generic server-compromise impersonation attack.

Theorem 3.1 *The TP-AMP protocol [9] is not an augmented PAKE protocol because it is insecure against server-compromise impersonation attacks.*

We prove this theorem by showing the attack below. In the server-compromise impersonation attacks, an attacker $\mathcal{A}$ is trying to impersonate client C with the obtained password verification data (W, Z) itself, but without doing off-line dictionary attacks on (W, Z). The generic server-compromise impersonation attack of $\mathcal{A}$ is as follows:

Step 1': The attacker $\mathcal{A}$ chooses two random elements $(\alpha, \beta) \stackrel{R}{\leftarrow} \left(\mathbb{Z}_q^\star\right)^2$ and computes $X \equiv g^\alpha \cdot W^\beta \cdot Z^{-1}$. Then, attacker $\mathcal{A}$ sends a message (C, X) to server S.

$$\mathcal{A} \rightarrow S : (C, X)$$

Step 2': After receiving (C, X), server S chooses a random element $y \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$, and computes $Y \equiv (X \cdot Z \cdot W)^y$, $K \equiv (X \cdot Z \cdot g)^y$ and $V_S = \mathcal{H}_1(C\| S\|X\|Y\|K)$ as usual. Then, server S sends a message (S, Y, V_S) to client C (actually, to attacker $\mathcal{A}$).

$$S \rightarrow \mathcal{A} : (S, Y, V_S)$$

In order to impersonate client C successfully, attacker $\mathcal{A}$ should find out γ satisfying $Y^\gamma \equiv K$. That is,

$$\begin{aligned} \log_g Y^\gamma &\equiv \log_g K \bmod q \\ (\alpha + pwb \cdot \beta + pwb)y \cdot \gamma &\equiv (\alpha + pwb \cdot \beta + 1)y \\ (\alpha + pwb(\beta + 1))\gamma &\equiv \alpha + 1 + pwb \cdot \beta \quad (2) \end{aligned}$$

Since there is no off-line dictionary attacks on (W, Z), the solution of Equation (2) is that $\alpha \cdot \gamma \equiv \alpha + 1 \bmod q$ and $(\beta + 1)\gamma \equiv \beta \bmod q$. If the attacker $\mathcal{A}$ chooses (α, β), such that $\alpha + \beta \equiv -1 \bmod q$, the server-compromise impersonation attack is always possible. This means that, by sending $X \equiv g^{-1-\beta} \cdot W^\beta \cdot Z^{-1}$ in **Step 1'**, the attacker $\mathcal{A}$ can impersonate client C successfully with $K' \equiv Y^{\beta/(\beta+1)} = K$. It can be easily verified that

$$\begin{aligned} K' &\equiv Y^{\beta/(\beta+1)} \equiv ((X \cdot Z \cdot W)^y)^{\frac{\beta}{\beta+1}} \\ &\equiv \left(g^{-1-\beta} \cdot W^\beta \cdot W\right)^{y \cdot \frac{\beta}{\beta+1}} \\ &\equiv \left(g^{-(\beta+1)} \cdot W^{\beta+1}\right)^{y \cdot \frac{\beta}{\beta+1}} \equiv \left(g^{-1} \cdot W\right)^{y \cdot \beta} \end{aligned}$$

and

$$\begin{aligned} K &\equiv (X \cdot Z \cdot g)^y \equiv \left(g^{-1-\beta} \cdot W^\beta \cdot g\right)^y \\ &\equiv \left(g^{-\beta} \cdot W^\beta\right)^y \equiv \left(g^{-1} \cdot W\right)^{y \cdot \beta} . \end{aligned}$$

Note that this attack is valid for any element $\beta \in \mathbb{Z}_q^\star$. Also, one can notice that this attack is very similar to the one, shown in Section 2, and does not matter its implementation of $\mathcal{G}$.

4 Concluding Remarks

In this paper, we have shown that two augmented PAKE protocols [8, 9] (claimed to be secure) are actually insecure against server-compromise impersonation attacks. More specifically, we have presented *generic* server-compromise impersonation attacks on [8, 9]. It is imperative to understand server-compromise impersonation attacks well and clearly since several augmented PAKE protocols are being considered as IEEE standard candidates in the IEEE P1363.2 working group [6].

参考文献

[1] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based

Protocols Secure against Dictionary Attacks", In *Proc. of IEEE Symposium on Security and Privacy*, pp. 72-84, IEEE Computer Society, 1992.

[2] S. M. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: A Password-based Protocol Secure against Dictionary Attacks and Password File Compromise", In *Proc. of ACM CCS'93*, pp. 244-250, ACM Press, 1993.

[3] V. Boyko, P. MacKenzie, and S. Patel, "Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman", In *Proc. of EUROCRYPT 2000*, LNCS 1807, pp. 156-171, Springer-Verlag, 2000.

[4] C. Gentry, P. MacKenzie, and Z. Ramzan, "A Method for Making Password-Based Key Excahnge Resilient to Server Compromise", In *Proc. of CRYPTO 2006*, LNCS 4117, pp. 142-159, Springer-Verlag, 2006.

[5] Y. H. Hwang, D. H. Yum, and P. J. Lee, "EPA: An Efficient Password-Based Protocol for Authenticated Key Exchange", In *Proc. of ACISP 2003*, LNCS 2727, pp. 452-463, Springer-Verlag, 2003.

[6] IEEE P1363, "IEEE P1363.2: Password-Based Public-Key Cryptography", `http://grouper.ieee.org/groups/1363/passwdPK/index.html`.

[7] ISO/IEC JTC 1/SC 27 11770-4, "Information Technology−Security Techniques−Key Management−Part 4: Mechanisms based on Weak Secrets", 2006. Available at `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=45306`.

[8] T. Kwon, Y. H. Park, and H. J. Lee, "Security Analysis and Improvement of the Efficient Password-based Authentication Protocol", *IEEE Communications Letters*, Vol. 9, No. 1, pp. 93-95, January 2005.

[9] T. Kwon, "Summary of AMP (Authentication and Key Agreement via Memorable Passwords)", IEEE P1363.2: Password-Based Public-Key Cryptography, Submissions to IEEE P1363.2, August 2003. Available at `http://grouper.ieee.org/groups/1363/passwdPK/contributions/ampsummary.pdf`.