Security Analysis and Improvement of a Gateway-Oriented Password-Based Authenticated Key Exchange Protocol

Jin Wook Byun, Dong Hoon Lee* and Jong In Lim

August 16, 2006

Abstract

Recently, Abdalla et al. proposed a gateway-oriented password-based authenticated key exchange (GPAKE) scheme among a client, a gateway, and an authentication server. A password is only shared between the client and the authentication server, not to the gateway. The security goal of GPAKE is securely establishing a session key between the client and the gateway by the help of the authentication server without revealing any information of client’s password to the gateway. The GPAKE scheme is formally claimed to be satisfied with the security goal in the presence of a malicious gateway. In this article, however, we show that a malicious gateway of GPAKE is still able to gain information of password by performing an undetectable on-line password guessing attack.

Index Terms: Gateway-oriented communication, undetectable on-line password guessing attack, three-party password-based authentication

1 Introduction

To communicate securely over an insecure public network it is essential that secret keys are exchanged securely. Password-based authenticated key exchange (PAKE) scheme allows two or more parties holding a same memorable password to agree on a common secret value (a session key) over an insecure open network. Over the years, there have been a great deal of research on efficient and provably secure password-based authenticated key exchange schemes in the two-party or n-party settings. However, the setting such that all participants trying to agree on a secret key have a same password is not practical since a password is not a common secret but a secret depending on an individual. Thus, in recent years, password-based authenticated key exchange using different passwords have received a lot of attentions in the literature [1, 4, 5, 6].

In 1995, Steiner et al. first proposed a three-party password-based encrypted key exchange (3P-EKE) scheme in which two clients have distinct password to authenticate each other, and

*Corresponding author. Email address: donghlee@korea.ac.kr (Dong Hoon Lee), byunstar@korea.ac.kr (Jin Wook Byun), jilim@korea.ac.kr (Jong In Lim). The authors are with Graduate School of Information Security (GSIS), Center for Information Security Technologies (CIST), Korea university, Seoul, Korea.
a server mediates between two clients to allow them to authenticate and agree a session key. Unfortunately, this scheme was found to be flawed by Ding and Horster [6]. They first classified password guessing attacks into three classes as follows, and showed that the 3P-EKE scheme was susceptible to an undetectable on-line password guessing attack.

- Detectable on-line password guessing attack (DOPGA) : An attacker first guesses a password, and try to verify the password using responses from a server in an on-line manner. If the password is not valid one then the server can detect a failed guess. This detection enables the server to prevent this type of attack by itself. For instance, if the number of failure exceeds three times, then the server may close the corresponding account, and strongly suggest a change of current password to the client. Therefore, in fact, this attack is not important consideration in designing a secure PAKE scheme.

- Off-line password guessing attack (OPGA) : An attacker uses a guessed password to verify the correctness of the password in an off-line manner. Thus, the attacker can freely guess a password and then verify it is correct or not without limitation. This attack is the most critical one among three types of attacks.

- Undetectable on-line password guessing attack (UOPGA) : A malicious insider client first guesses a password of one of the clients and uses his guess in an on-line transaction. The malicious client verifies correctness of his guess using responses of the server. Note that a failed guess never be noticed by the server and other clients. Thus, the malicious client can get sufficient information of password by participating the protocol legally and undetectably many times.

Consequently, the important security requirements of three-party based PAKE are to attain security against the OPGA and the UOPGA.

In 2000, Lin et al. showed that 3P-EKE is not only vulnerable to an UOPGA, but also vulnerable to an OPGA [4]. They also proposed a new 3PEKE scheme where the server uses its public key to resolve both an OPGA and an UOPGA. The server’s public key actually puts a high burden on the client for verifying, and they redesigned the 3PEKE scheme without the server’s public key [5].

In 2005, Abdall et al. proposed a gateway-oriented password-based authenticated key exchange (GPAKE) scheme which is also three-party are involved, but slightly different from the existing models in aspects of authentication setting and security goal. The GPAKE scheme consists of a client, a gateway, and an authentication server. The authentication server and the client priorly share a password for authentication, but a common session key is generated between the gateway and the client. The security goal is to securely generate a session key between the client and the gateway preventing leakage of information of password from the gateway.
In this letter, we review the GPAKE scheme and show that it does actually leak information of password to a malicious gateway. Especially, we show that the GPAKE scheme is susceptible to an undetectable on-line password guessing attack by a malicious gateway. We also give a countermeasure against the attack by letting the client generate a message authentication code of keying material.

The rest of article is organized as follows. In Section 2, we review Abdalla et al.’s GPAKE scheme. In Section 3, we demonstrate that the GPAKE scheme is not secure in the presence of a malicious gateway and present a countermeasure to prevent the malicious gateway. In Section 4, we conclude the paper.

2 Review of Abdalla et al.’s GPAKE Scheme

2.1 Notation

We use the notations in Table 1 throughout this paper.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C, G, S$</td>
<td>identities of a client, a gateway, and an authentication server.</td>
</tr>
<tr>
<td>$pw$</td>
<td>a password shared by a client and an authentication server.</td>
</tr>
<tr>
<td>$sk$</td>
<td>a generated session key between a client and a gateway.</td>
</tr>
<tr>
<td>$\text{MAC}_k$</td>
<td>a message authentication code algorithm such that $\text{MAC}_k : {0, 1}^* \rightarrow {0, 1}^l$.</td>
</tr>
<tr>
<td>$H_1, H_2$</td>
<td>cryptographic hash functions (e.g., SHA-1).</td>
</tr>
<tr>
<td>$\mathbb{G}$</td>
<td>a finite cyclic group has a generator $\langle g \rangle$ of $l$ bit prime order $q$.</td>
</tr>
<tr>
<td>$\mathcal{G}$</td>
<td>a full domain hash function such that $\mathcal{G} : {0, 1}^* \rightarrow \mathbb{G}$.</td>
</tr>
</tbody>
</table>

Table 1: Notation

2.2 Description of GPAKE Scheme

The GPAKE scheme consists of a client $C$, a gateway $G$, and an authentication server $S$. The communication channel between the gateway and the server is assumed to be authenticated and private. We illustrate a framework of GPAKE in the figure 1.

(1) $C$ first selects a random number $x \in \mathbb{Z}_q^*$ and computes $X^* = g^x \cdot PW$ where $PW = \mathcal{G}(C, G, pw)$. $C$ sends $(C, X^*)$ to the gateway $G$.

(2) On receiving values $(C, X^*)$, $G$ selects a random number $y \in \mathbb{Z}_q^*$ and computes $Y = g^y$, and sends messages $(C, X^*, Y)$ to the authentication server $S$.

(3) On receiving values $(C, X^*, Y)$, $S$ selects a random number $s \in \mathbb{Z}_q^*$ and computes $\overline{X} = X^*/PW$, $\overline{Y} = Y^*$ where $X = X^*/PW$. $S$ sends $(\overline{X}, \overline{Y})$ back to $G$. 

3
3 Security Vulnerability of GPAKE and its Countermeasure

3.1 Undetectable On-line Password Guessing Attack on GPAKE

In the following, we demonstrate an undetectable on-line password guessing (UOPG) attack against the GPAKE scheme where a malicious gateway of GPAKE is able to legally gain information about the password by repeatedly and indiscernibly asking queries to the authentication server.

- **Step 1.** Let $G_A$ be a malicious gateway mediating between $S$ and $C$. Upon receiving $(C, X^*)$ from the client in Flow (1) of the GPAKE scheme in Figure 1, $G_A$ first guesses a password $pw'$ and then computes $PW' = G(C, G_A, pw')$.

- **Step 2.** $G_A$ randomly generates $Y'$ and $AuthG'$, then sends $(G_A, Y', AuthG')$ to $C$ as a message of Flow (4) in Figure 1. $C$ would compute its authenticator $AuthG'' = H_2(C, G_A, X^*, Y', K)$ and compare it with the given $AuthG'$. Even $C$ may detect only once that the authenticator $AuthG'$ is invalid, it really does not mean that $S$ detects a failure of $G_A$’s malicious trial.

- **Step 3.** On the other hand, $G_A$ establishes an authenticated and private channel with $S$, and then repeatedly performs the followings without being noticed by $S$.

(4) $G$ computes $K = X^y = g^{xy}$ and $AuthG = H_2(C, G, X^*, Y, K)$. $G$ also generates a session key $sk = H_1(C, G, X^*, Y, K)$. $G$ sends $(G, Y, AuthG)$ to the client $C$.

(5) On receiving values $(G, Y, AuthG)$, $C$ also computes $K = Y^x$ and checks whether $AuthG$ is $H_2(C, G, X^*, Y, K)$ or not. If the check succeeds then $C$ generates a session key $sk = H_1(C, G, X^*, Y, K)$.
− With the guessed password $pw'$ and $X^*$, $G_A$ first computes $X' = \frac{X^*}{PW} = g^{x'}$ for an unknown element $x' \in Z_q^*$. $G_A$ then selects a random element $y \in Z_q^*$ and computes $Y' = (X')^y = g^{x'y}$ and $X'^* = g^{x'} \cdot PW'$. $G_A$ sends $(C, X'^*, Y')$ to $S$ in Flow (2) of Figure 1.

− Upon receiving $(C, X'^*, Y')$, $S$ selects a random value $s \in Z_q^*$ and computes $X' = (X'^*)^s$ and $\overline{Y} = (Y')^s = g^{x's}$. $S$ sends $(\overline{X'}, \overline{Y'})$ back to $G_A$ in Flow (3).

− $G_A$ checks if $X' = (\overline{Y'})^{-y}$. If the check passes, then $G_A$ confirms that the guessed password $pw'$ is the correct one.

It is clear that if $pw' = pw$ and hence $PW' = PW$, $X' = (\overline{Y'})^{-y}$ since

$$
\begin{align*}
\overline{X'} &= (X'^*)^s = (\frac{X^*}{PW})^s = g^{x's} \\
(\overline{Y'})^{-y} &= g^{x's/y} = g^{xs}.
\end{align*}
$$

Consequently, the above attack obviously shows that the GPAKE scheme does not prevent the leakage of information of the password from the malicious gateway $G_A$.

### 3.2 The modified GPAKE and its Security Analysis

The vulnerability to the UOPG attack described above actually stems from an absence of authentication of message $X^* = g^{x} \cdot PW$ in the GPAKE scheme. To remedy this vulnerability in the modified GPAKE scheme, we let $C$ create a message authentication code (MAC) of $X^*$ and enable $S$ to verify the MAC of $X^*$, where a MAC key is assumed to be securely shared between $C$ and $S$. To establish a MAC key, a 2-PAKE scheme is executed between $C$ and $S$. We illustrate the modified GPAKE scheme in Figure 2 which works as follows.

1. The client $C$ and the authentication server $S$ priorly execute a secure 2-PAKE scheme such as one in [3], and establish a common secret key $k$ as a MAC key. $C$ selects a random number $x \in Z_q^*$ and computes $X^* = g^{x} \cdot PW$ and its MAC, $\delta = MAC_k(X^*)$. $C$ then sends $(C, X^*, \delta)$ to the gateway $G$.

2. Upon receiving $(C, X^*, \delta)$, $G$ selects a random number $y \in Z_q^*$ and computes $Y = g^y$, and sends messages $(C, X^*, \delta, Y)$ to the authentication server $S$.

3. Upon receiving $(C, X^*, \delta, Y)$, $S$ first verifies a validity of $\delta = MAC_k(X^*)$. If the check passes, $S$ selects a random number $s \in Z_q^*$ and computes $\overline{X} = X^s$ and $\overline{Y} = Y^s$ where $X = X^*/PW$. $S$ sends $(\overline{X}, \overline{Y})$ back to $G$. The rest of executions are identical to the original GPAKE scheme.

The following theorem shows that the above countermeasure is secure against an UOPG attack.
\[ x \in \mathbb{Z}_q^* \]
\[ X^* = g^x \cdot PW \]
\[ \delta = \text{MAC}_k(X^*) \]

(1) \[ C, X^*, \delta \rightarrow y \in \mathbb{Z}_q^* \]
\[ Y = g^y \]

(2) \[ C, X^*, \delta, Y \rightarrow \text{Verify} \]
\[ s \in \mathbb{Z}_q^* \]
\[ X = X^*/PW \]
\[ X^* = X^s \]
\[ Y = Y^s \]

(3) \[ K = \text{MAC}_k(Y^s) \]

(4) \[ G, Y, \text{AuthG} \]

K = Y^s
Check AuthG
Compute sk

Figure 2: The modified GPAKE scheme

(a) A GPAKE Scheme

(b) A Modified GPAKE Scheme

Figure 3: A GPAKE scheme and its countermeasure
Theorem 1. If 2-PAKE is a secure 2-party password-based authenticated key exchange and \( \text{MAC}_k(\cdot) \) is a secure MAC algorithm such that \( \text{MAC}_k : \{0,1\}^* \rightarrow \{0,1\}^l \), then the modified GPAKE scheme is secure against UOPG attacks.

Proof. Let \( G_A \) be a malicious gateway performing an UOPG attack against the modified GPAKE scheme. To undetectably pass a verification process of MAC algorithm by \( S \) in Figure 2, \( G_A \) would make attempts to generate a valid MAC value of \( X^{x'} = g^{x'} \cdot PW' \) where \( pw' \) and \( x' \) are repeatedly generated by \( G_A \) itself. The probability of successfully generating the valid MAC value consists of the following two cases.

• Case 1 : \( G_A \) may obtain a secret key \( k \) generated by 2-PAKE scheme. However this probability is negligible since a secure 2-PAKE scheme means that the secret key \( k \) is known to nobody but the participants (client and authentication server) under passive or active attackers.

• Case 2 : Let \( FG \) be an event that \( G_A \) fortunately guesses a valid MAC pair \( (m, \text{MAC}_k(m)) \) without knowing the key \( k \). However this probability is also negligible since if the MAC algorithm is a secure one as a secure cipher block chaining MAC in [2] and \( G_A \) queries at most \( q \) times, then the probability \( \Pr[FG] \) is at most \( \frac{q}{2^l} \).

By the above two cases, \( G_A \) never generate valid message authentication codes with non-negligible probability, and accordingly \( G_A \) cannot go through the verification process without being detected by \( S \).

4 Concluding Remarks

In this paper, we have demonstrated that the GPAKE scheme by Abdalla et al. was susceptible to an undetectable on-line password guessing attack. And also, we presented a countermeasure for the attack by applying an algorithm of message authentication code to both sides of a client and an authentication server. Inevitably, additional rounds between \( C \) and \( S \) were required to negotiate a MAC key.

References


[5] Chun-Li Lin, Hung-Min Sun, Michael Steiner, and Tzonelih Hwang, “Three-party encrypted key ex-
change Without Server Public-Keys”, *In IEEE Communications Letters*, Vol. 5, No. 12, pp. 497-499,