Infringing and Improving Password Security of a Three-Party Key Exchange Protocol

Junghyun Nam[†]

February 5, 2008

Abstract

Key exchange protocols allow two or more parties communicating over a public network to establish a common secret key called a *session key*. Due to their significance in building a secure communication channel, a number of key exchange protocols have been suggested over the years for a variety of settings. Among these is the so-called S-3PAKE protocol proposed by Lu and Cao for password-authenticated key exchange in the three-party setting. In the current work, we are concerned with the password security of the S-3PAKE protocol. We first show that S-3PAKE is vulnerable to an *off-line dictionary attack* in which an attacker exhaustively enumerates all possible passwords in an attempt to determine the correct one. We then figure out how to eliminate the security vulnerability of S-3PAKE.

Keywords: Key exchange protocol, secure communication, password, off-line dictionary attack.

1 Introduction

Back in 1992, Bellovin and Merritt [5] considered how two parties, who only share a short password and who are communicating over a public network, come up with a common session key to be used for protecting their subsequent communications. Their set of protocols, known as encrypted key exchange, or EKE, was a great success in showing how one can exchange password-authenticated information while protecting poorly-chosen passwords from the notorious off-line dictionary attacks. Due to the practical significance of password-based authentication, this initial work has been followed by a number of two-party protocols (e.g., [7, 4, 9]) offering various levels of security and complexity.

While two-party protocols for password-authenticated key exchange (PAKE) are well suited for client-server architectures, they are inconvenient and costly for use in

[†]Department of Computer Science, Konkuk University, 322 Danwol-dong, Chungju-si, Chungcheongbuk-do 380-701, Republic of Korea.
E-mail: jhnam@kku.ac.kr

large scale peer-to-peer systems. Since two-party PAKE protocols require each pair of potential communication parties to share a password, a large number of parties result in an even larger number of passwords to be shared. It is due to this problem that three-party models have been often used in designing PAKE protocols (e.g., [12, 1, 2, 10, 8]). In a typical three-party setting, each party (commonly called *client*) does not need to remember and manage multiple passwords, but shares only a single password with a trusted server who then assists clients in establishing a session key by providing authentication services to them. However, this convenience comes at the price of clients' complete trust in the server. Despite this drawback, the three-party model offers an effective, realistic solution to the problem of session key exchange in large peer-to-peer systems, and in fact is assumed by the popular *Kerberos* authentication system [11].

Recently in [10], Lu and Cao proposed a simple three-party PAKE protocol (in short, S-3PAKE) built upon the earlier two-party PAKE protocol due to Abdalla and Pointcheval [3]. The main design goal of S-3PAKE is to provide both efficiency and security without recourse to the use of server's public keys. However, the S-3PAKE protocol turned out to have security vulnerabilities failing to achieve its goal. According to Chung and Ku [8], S-3PAKE is vulnerable to various impersonation attacks and thus does not satisfy implicit key authentication [6]. Besides pointing out the vulnerability, Chung and Ku also make a simple suggestion to fix the S-3PAKE protocol. (See the end of the next section for Chung and Ku's suggestion.)

The above-mentioned impersonation attacks are not the only ones that can compromise the security of the S-3PAKE protocol. We found that S-3PAKE (even with Chung and Ku's suggestion applied) is not secure against an off-line dictionary attack. The present work reports this new (and probably more serious) security problem with S-3PAKE and, in addition, shows how to fix it.

2 The S-3PAKE Protocol

This section reviews the S-3PAKE protocol proposed by Lu and Cao [10]. Let A and B be two clients who wish to establish a session key, and S be a trusted authentication server with which client A (resp. B) has registered a password pw_A (resp. pw_B). Then the S-3PAKE protocol runs among the three parties, S, A and B, with the following public parameters established:

- A cyclic group $\mathbb{G}$ of prime order q generated by an element g.
- Two elements M and N of group $\mathbb{G}$.
- Two one-way hash functions $\mathcal{G}$ and $\mathcal{H}$, where the outputs of $\mathcal{G}$ are the elements of $\mathbb{G}$.

Below is a description of the S-3PAKE protocol.

Step 1. A chooses a random number $x \in \mathbb{Z}_q$, computes $X = g^x$ and $X^* = X \cdot M^{pw_A}$, and sends $A\|X^*$ to B.

Step 2. B selects a random number $y \in \mathbb{Z}_q$, computes $Y = g^y$ and $Y^* = Y \cdot N^{pw_B}$, and sends $A\|X^*\|B\|Y^*$ to S.

Step 3. Upon receiving $A\|X^*\|B\|Y^*$, S first recovers X and Y by computing $X = X^*/M^{pw_A}$ and $Y = Y^*/N^{pw_B}$. Next, S selects a random number $z \in \mathbb{Z}_q$ and computes $\overline{X} = X^z$ and $\overline{Y} = Y^z$. S then computes

$$\begin{aligned} pw_A^* &= \mathcal{G}(A\|S\|X)^{pw_A}, \\ pw_B^* &= \mathcal{G}(B\|S\|Y)^{pw_B}, \\ \overline{X}^* &= \overline{X} \cdot pw_B^*, \\ \overline{Y}^* &= \overline{Y} \cdot pw_A^*, \end{aligned}$$

and sends $\overline{X}^*\|\overline{Y}^*$ to B.

Step 4. After having received $\overline{X}^*\|\overline{Y}^*$, B computes

$$\begin{aligned} pw_B^* &= \mathcal{G}(B\|S\|Y)^{pw_B}, \\ K &= \Big(\frac{\overline{X}^*}{pw_B^*}\Big)^y, \\ \alpha &= \mathcal{G}(A\|B\|K), \end{aligned}$$

and sends $\overline{Y}^*\|\alpha$ to A.

Step 5. With $\overline{Y}^*\|\alpha$ from B, A computes

$$\begin{aligned} pw_A^* &= \mathcal{G}(A\|S\|X)^{pw_A}, \\ K &= \Big(\frac{\overline{Y}^*}{pw_A^*}\Big)^x, \end{aligned}$$

and verifies that α is equal to $\mathcal{G}(A\|B\|K)$. If the verification fails, then A aborts the protocol. Otherwise, A computes the session key $SK = \mathcal{H}(A\|B\|K)$ and sends $\beta = \mathcal{G}(B\|A\|K)$ to B.

Step 6. B verifies the correctness of β by checking that the equation $\beta = \mathcal{G}(B\|A\|K)$ holds. If it holds, then B computes the session key $SK = \mathcal{H}(A\|B\|K)$. Otherwise, B aborts the protocol.

The correctness of S-3PAKE can be easily verified from the equations

$$\begin{aligned} K &= \Big(\frac{\overline{X}^*}{pw_B^*}\Big)^y \\ &= \Big(\frac{\overline{X} \cdot pw_B^*}{pw_B^*}\Big)^y \\ &= g^{xyz} \end{aligned}$$

and

$$\begin{aligned} K &= \Big(\frac{\overline{Y}^*}{pw_A^*}\Big)^x \\ &= \Big(\frac{\overline{Y} \cdot pw_A^*}{pw_A^*}\Big)^x \\ &= g^{xyz}. \end{aligned}$$

As mentioned in the Introduction, Chung and Ku [8] showed that the S-3PAKE protocol described above is vulnerable to various impersonation attacks. Their suggestion to fix the protocol is to include the identities of both A and B when computing pw_A^* and pw_B^* as follows:

$$\begin{aligned} pw_A^* &= \mathcal{G}(A\|B\|S\|X)^{pw_A}, \\ pw_B^* &= \mathcal{G}(B\|A\|S\|Y)^{pw_B}. \end{aligned}$$

However, we note that this suggestion does not help to resist our off-line dictionary attack, as will be clear in the next section.

3 Off-Line Dictionary Attack on S-3PAKE

Lu and Cao [10] claim that their S-3PAKE protocol is secure against off-line dictionary attacks. They support this claim with the argument that says: *S-3PAKE does not leak any information that allows to verify the correctness of password guesses, because $\mathcal{G}$ is a one-way hash function and x, y and z are all random numbers.* This argument may hold if there only exist honest clients who stick to the protocol specification. But, there could be malicious clients who deviate from the protocol. Indeed, we found that S-3PAKE is not secure against an off-line dictionary attack in the presence of a malicious client.

Assume that B is a malicious client, and wants to find out the password of client A. Then the following description represents our off-line dictionary attack mounted by B against A's password.

Phase 1. The attacker B runs the protocol with the server S while playing dual roles of B itself and the victim A.

1. B selects two random numbers $x, y \in \mathbb{Z}_q$ and computes X^* and Y^* as

$$\begin{aligned} X^* &= M^x, \\ Y^* &= M^y \cdot N^{pw_B}. \end{aligned}$$

   B then sends S the forged message $A\|X^*\|B\|Y^*$ alleging that A and B want to establish a session key.

2. Following the protocol, S will send $\overline{X}^* \| \overline{Y}^*$ to B in response to $A \| X^* \| B \| Y^*$. But notice here that $\overline{X}^*$ and $\overline{Y}^*$ are set equal to, respectively, $M^{(x-pw_A)z} \cdot pw_B^*$ and $M^{yz} \cdot pw_A^*$ because S computes them as

$$\begin{aligned}
\overline{X}^* &= \overline{X} \cdot pw_B^* \\
&= X^z \cdot pw_B^* \\
&= \Big(\frac{X^*}{M^{pw_A}}\Big)^z \cdot pw_B^* \\
&= \Big(\frac{M^x}{M^{pw_A}}\Big)^z \cdot pw_B^* \\
&= M^{(x-pw_A)z} \cdot pw_B^*
\end{aligned}$$

and

$$\begin{aligned}
\overline{Y}^* &= \overline{Y} \cdot pw_A^* \\
&= Y^z \cdot pw_A^* \\
&= \Big(\frac{Y^*}{N^{pw_B}}\Big)^z \cdot pw_A^* \\
&= \Big(\frac{M^y \cdot N^{pw_B}}{N^{pw_B}}\Big)^z \cdot pw_A^* \\
&= M^{yz} \cdot pw_A^*
\end{aligned}$$

where $pw_A^* = \mathcal{G}(A \| S \| M^{(x-pw_A)})^{pw_A}$ and $pw_B^* = \mathcal{G}(B \| S \| M^y)^{pw_B}$.

Phase 2. Using $\overline{X}^*$ and $\overline{Y}^*$ from S, B now guesses possible passwords and checks them for correctness.

1. First, B computes $pw_B^* = \mathcal{G}(B \| S \| M^y)^{pw_B}$ and

$$\begin{aligned}
K &= \Big(\frac{\overline{X}^*}{pw_B^*}\Big)^y \\
&= \Big(\frac{M^{(x-pw_A)z} \cdot pw_B^*}{pw_B^*}\Big)^y \\
&= M^{(x-pw_A)yz}.
\end{aligned}$$

2. Next, B makes a guess pw'_A for the password pw_A and computes $pw'^*_A = \mathcal{G}(A \| S \| M^{(x-pw'_A)})^{pw'_A}$ and

$$\begin{aligned}
K' &= \Big(\frac{\overline{Y}^*}{pw'^*_A}\Big)^{(x-pw'_A)} \\
&= \Big(\frac{M^{yz} \cdot pw_A^*}{pw'^*_A}\Big)^{(x-pw'_A)}.
\end{aligned}$$

3. B then verifies the correctness of pw'_A by checking that K is equal to K'. Notice that if pw'_A and pw_A are equal, then the equation $K = K'$ ought to be satisfied.

4. B repeats steps 2 and 3 of this phase until a correct password is found.

This off-line dictionary attack may lead to devastating losses of passwords, because it can be mounted against any registered client and does not even require the participation of the victim.

4 Countermeasure

The vulnerability of S-3PAKE to the off-line dictionary attack above is attributed to the fact that pw_A is used as the exponent of M in the computations of $X^* = X \cdot M^{pw_A}$ and $X = X^*/M^{pw_A}$. This fact gives the attacker B the idea of setting the values of X^* and Y^* equal to M^x and $M^y \cdot N^{pw_B}$. Fortunately, a slight modification of how X^* and Y^* are computed can prevent the off-line dictionary attack. Employing the technique used for Abdalla and Pointcheval's 3PAKE protocol [2], we recommend the following changes to S-3PAKE:

- Let $\overline{pw}_A = \mathcal{G}(pw_A)$ and $\overline{pw}_B = \mathcal{G}(pw_B)$. Then A computes X^* as $X^* = X \cdot \overline{pw}_A$ instead of as $X^* = X \cdot M^{pw_A}$, and B computes Y^* as $Y^* = Y \cdot \overline{pw}_B$ instead of as $Y^* = Y \cdot N^{pw_B}$.

- Accordingly, the way S recovers X and Y is modified to $X = X^*/\overline{pw}_A$ and $Y = Y^*/\overline{pw}_B$. (Of course, S should abort the protocol immediately if any of X and Y is found to be equal to 1. Otherwise, the protocol still suffers from an off-line dictionary attack similar to the one described above.)

The rest of the protocol remains unchanged. Notice that the public parameters M and N are no longer necessary in our new version of S-3PAKE.

With our modification applied, the S-3PAKE protocol becomes secure against off-line dictionary attacks. No matter what value X^* takes, the value of $\overline{X}^*$ returned from S is of no help in verifying password guesses. It is because $\overline{X}^*$ is now computed as $\overline{X}^* = (X^*/\overline{pw}_A)^z \cdot pw_B^*$ where $\overline{pw}_A$ is directly raised to a random secret power z.

References

[1] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. PKC '05*, LNCS vol. 3386, pp. 65–84, 2005.

[2] M. Abdalla and D. Pointcheval, "Interactive Diffie-Hellman assumptions with applications to password-based authentication," in *Proc. FC '05*, LNCS vol. 3570, pp. 341–356, 2005.

[3] M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," in *Proc. CT-RSA '05*, LNCS vol. 3376, pp. 191–208, 2005.

[4] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT '00*, LNCS vol. 1807, pp. 139–155, 2000.

[5] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proc. 1992 IEEE Symposium on Research in Security and Privacy*, pp. 72–84, 1992.

[6] C. Boyd and A. Mathuria, "Protocols for authentication and key establishment," Springer-Verlag, 2003.

[7] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman," in *Proc. EUROCRYPT '00*, LNCS vol. 1807, pp. 156–171, 2000.

[8] H.-R. Chung and W.-C. Ku, "Three weaknesses in a simple three-party key exchange protocol," *Information Sciences*, vol. 178, no. 1, pp. 220–229, 2008.

[9] J. Katz, R. Ostrovsky, and M. Yung, "Efficient password-authenticated key exchange using human-memorable passwords," in *Proc. EUROCRYPT '01*, LNCS vol. 2045, pp. 475–494, 2001.

[10] R. Lu and Z. Cao, "Simple three-party key exchange protocol," *Computers & Security*, vol. 26, no. 1, pp. 94–97, 2007.

[11] J. Steiner, C. Newman, and J. Schiller, "Kerberos: an authentication service for open network systems," in *Proc. 1998 USENIX Winter Conference*, pp. 191–202, 1998.

[12] M. Steiner, G. Tsudik, and M. Waidner, "Refinement and extension of encrypted key exchange," *ACM SIGOPS Operating Systems Review*, vol. 29, no. 3, pp. 22–30, 1995.