INFORMATICA, 2001, Vol. 12, No. 3, 469–476

On the Security of Methods for Protecting Password Transmission *

Yuh-Min TSENG
Department of Information Management, Nan-Kai College of Technology and Commerce
Nantou, Taiwan 542, R.O.C.
e-mail: tym@bear.nkjc.edu.tw

Jinn-Ke JAN, Hung-Yu CHIEN
Institute of Applied Mathematics, National Chung Hsing University
Taichung, Taiwan 402, R.O.C.

Received: October 2000

Abstract. Peyravian and Zunic (2000) proposed a password transmission scheme and a password change scheme over an insecure network. Their proposed solutions do not require the use of any symmetric-key or public-key cryptosystems. However, this article points out that their schemes have several security flaws for practical applications. A slight improvement on their schemes is proposed in this paper to remove the security flaws.

Key words: cryptography, password, hash function, discrete logarithm.

1. Introduction

Computer network technologies have encouraged the distribution of information. Various resources distributed among the hosts are shared across the network in the form of network services provided by servers. Servers in a distributed computer system provide storage of information, users (clients) can request services through the network system. However, an eavesdropper can impersonate a legal user to login into the server later by intercepting the transmitted messages in the public network. Thus, the requirement to provide a secure user authentication scheme is important.

For user authentication in a distributed network, a password-based scheme is still the most popular. There are several password-based schemes (Botting, 1997; Jablon, 1996; Horng, 1995) have been proposed to provide user authentication. These schemes require symmetric-key or public-key cryptosystems to encrypt the passwords while travelling over public networks. Recently, Peyravian and Zunic (Peyravian *et al.*, 2000) first proposed a secure method for protecting passwords while being transmitted over public networks. Then, they also present a password change scheme. Both two schemes do not adopt any symmetric-key or public-key cryptosystems.

*This research was partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC89–2213-E-252–008.

Unfortunately, there are several security flaws in Peyravian-Zunic's schemes. We find out that their schemes can not withstand the dictionary attack (Morris and Thompson, 1979) or the guessing attack, and do not provide the mutual authentication between a client and a server. An attacker can intercept the transmitted messages over the public network and record them. Then, the eavesdropper repeatedly verifies the guessed password for each candidate password. Thus, the eavesdropper can easily obtain the correct password to masquerade the original user.

In this paper, we will present an improvement on the Peyravian-Zunic's protocols to overcome the dictionary attack and enable the client and the server to authenticate each other. In Peyravian-Zunic's protocols, because their schemes adopt two random integers to protect the transmitted passwords, but the two random integers are plainly sent between the client and the server, so that their protocols are vulnerable to the dictionary attack. Therefore, our improvement will adopt the Diffie-Hellman key agreement scheme (Diffie and Hellman, 1976) to construct a common ephemeral integer between the client and the server. An attacker can not compute the common ephemeral integer so that he/she also does not verify the guessed password for each candidate password. This is because the security of the Diffie-Hellman scheme is based upon the difficulty of computing the discrete logarithms in a finite field.

The remainder of this paper is organized as follows. In the next section, we review briefly the Peyravian-Zunic schemes. The security flaws on their schemes are described in Section 3. In Section 4, we show our improvement based on the discrete logarithm problems. In Section 5, we discuss the security analysis and present other properties of the improvement. Section 6 gives our conclusions.

2. Review of Peyravian-Zunic Schemes

In this section, we first review a method for protecting password transmission proposed by Peyravian and Zunic. Then, we also review their another method for changing passwords.

Both methods have the same initialization phase. Servers (or Service Centers) store resources and can provide some access services. They are responsible for user authentication and access control for each user. Clients (or Users) can be defined as individual subscribers who own computer workstations. The client may request the different access services from the different servers. Assume that a client with identity id owns a password pw, which is being shared with the server. Note that the server does not store the password, but stores $idpw\_digest = Hash(id, pw)$, where $Hash(\,)$ is a strong one-way hash function such as SHA-1 scheme (Schneier, 1996).

Here, we review Peyravian-Zunic's scheme for password transmission as follows.

Step 1. The client sends the identity id and a random integer rc to the server.

Step 2. The server chooses a random value rs and sends it to the client.

Step 3. The client first computes the following values:

$idpw\_digest = Hash(id, pw)$

$auth\_token = Hash(idpw\_digest, rc, rs)$

Then, the client sends id and $auth\_token$ to the server.

Step 4. The server has to use his own $idpw\_digest$, rc and rs to compute the $auth\_token$, and compare it with the received $auth\_token$ from the client. If it holds, the server sends a message to the client giving him access permission.

As for Peyravian-Zunic's password change scheme, they use the same steps as in the password transmission scheme, except for Step 3 and Step 4. The detail modification is presented as follows.

**Step* 3. The client first generates a new password new_pw and computes the following values:

$idpw\_digest = Hash(id, pw)$

$auth\_token = Hash(idpw\_digest, rc, rs)$

$idpw\_digest\_new = Hash(id, new\_pw)$

$auth\_token\_mask = Hash(idpw\_digest, rc + 1, rs)$

$protected\_idpw\_digest\_new = idpw\_digest\_new \mathrm{XOR} auth\_token\_mask$

where XOR is the "exclusive-or" operation. Then, the client sends id, $auth\_token$ and $protected\_idpw\_digest\_new$ to the server.

**Step* 4. The server first has to use his own $idpw\_digest$, rc and rs to compute the $auth\_token$, and compare it with the received $auth\_token$ from the client. If it holds, the server generates $auth\_token\_mask$ to retrieve the $idpw\_digest\_new$ using XOR operation from $protected\_idpw\_digest\_new$.

From the above description, unlike existing solutions, their schemes do not employ any symmetric-key or public-key cryptosystems. Their proposed schemes only use a collision-resistant hash function.

3. Security Flaws

In this section, we present two security flaws on Peyravian-Zunic's password transmission scheme and password change scheme. In Peyravian-Zunic's two schemes, the main difference between them is that the client sends $protected\_idpw\_digest\_new$ to the server for changing password. Thus, we present the security flaws as follows. One is the dictionary attack, another is the forgery server attack.

[*Dictionary attack*]

Because ordinary users seem to have a fundamental inability to remember large passwords, the user-selected passwords are often confined to a very small. However, to use an easily memorized small password, it will be vulnerable to the dictionary attack or the guessing attack because the memorized password belongs to a brute-force searchable space. An attacker can intercept the transmitted messages over the public network and record them. Then, the eavesdropper iterative verifies the guessed password for each candidate password.

In the following, we show that their password transmission scheme is vulnerable to the dictionary attack. Suppose that there is an eavesdropper who records the transmitted

messages id, rc, rs and $auth\_token$ between the client and the server. First, the eavesdropper repeatedly computes $idpw\_digest = Hash(id, pw)$ for each candidate password pw. Then, he/she computes $Hash(idpw\_digest, rc, rs)$ for each generated $idpw\_digest$ and compares the result to $auth\_token$. Thus, the eavesdropper can easily obtain the correct password to masquerade the original user.

For the same reason, Peyravian-Zunic's password change scheme also suffers from the same attack as their password transmission scheme. Since the old $idpw\_digest$ has been revealed, thus the eavesdropper first obtains $auth\_token\_mask$ by computing $Hash(idpw\_digest, rc+1, rs)$. Then, he can retrieve $idpw\_digest\_new$ using XOR operation from $protected\_idpw\_digest\_new$. Therefore, both Peyravian-Zunic's two schemes are vulnerable to the dictionary attack.

[*Forgery server attack*]

As we reviewed in Section 2, the server only chooses a random value rs and sends it to the client in Peyravian-Zunic's schemes. The server does not provide authenticated messages to the client, and the client can not know whether the server has the same pre-shared password with his own. That is, their schemes do not provide mutual authentication between the server and the client. In such case, an attacker can impersonate a server to provide the wrong information to the honest users. In fact, the internet technologies have encouraged the electronic commerce, and many companies have constructed their own Web sites to provide electronic services. In the following, we give a real example occurred in Taiwan to present the security flaw. An attacker (the opponent company) constructs a forgery Web site with the same views as the real server, and the attacker may put his Web site's address into some search Webs (such as Yahoo, Infoseek). Generally, because users obtain the Web site's address using search Webs, so they may acquire the forgery Web site. In such case, the attacker may provide the wrong information to the honest users and acquire the useful information from users. Therefore, mutual authentication between the client and the server is needed.

4. Improvement on Peyravian-Zunic's Schemes

For removing the weakness of Peyravian-Zunic's schemes, we present a slight modification on their schemes based upon the discrete logarithm problem. The improved schemes do not adopt any symmetric-key and public-key cryptosystems. During the initialization phase, the server chooses and publishes two large prime numbers p and q such that q divides $p-1$. Let g be a generator with order q in the Galois field $GF(p)$. $GF(p)$ means the set of integers $\bmod p$, together with arithmetic operations, and it is a finite field. Assume that a client with identity id owns a password pw, which being shared with the server. Note that the server does not store the password, and instead it stores $idpw\_digest = Hash(id, pw)$.

For the password transmission and the password change schemes, the main difference between them is that the client additionally sends $protected\_idpw\_digest\_new$ to the

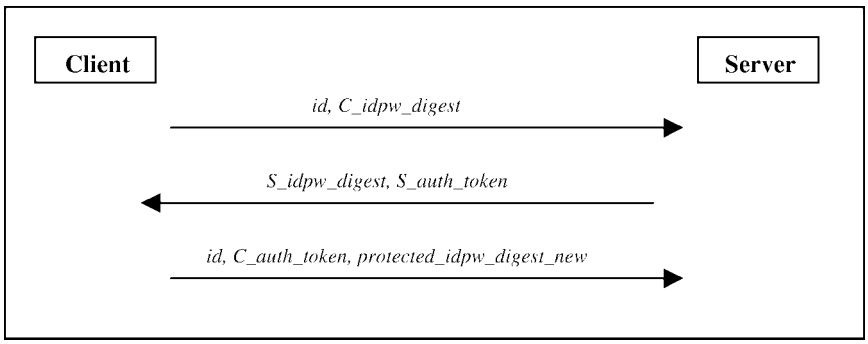

Fig. 1. Password change scheme.

server for changing password. Thus, we only show the password change scheme. As illustrated in Fig. 1, the detailed steps are presented as follows.

Step 1. The client chooses a random integer a in Z_q^*, where Z_q^* is the reduced residue system $\{1, 2, \ldots, q-1\} (\mathrm{mod}\ q)$. Then he/she computes $rc = g^a \bmod p$. Then, he/she computes two values as

$$idpw\_digest = Hash(id, pw) \quad \text{and} \quad C\_idpw\_digest = idpw\_digest \oplus rc \tag{1}$$

where $\oplus$ is the "exclusive-or" (for short, XOR) operation. The client sends the identity id and the value C_idpw_digest to the server.

Step 2. Receiving the messages from the client, the server also chooses a random integer b in Z_q^*, and then computes $rs = g^b \bmod p$ and $rcs = (rc)^b \bmod p = g^{ab} \bmod p$.

The server computes the following values:

$$S\_idpw\_digest = idpw\_digest \oplus rs \tag{2}$$

$$S\_auth\_token = hash(idpw\_digest, rc, rcs) \tag{3}$$

The server sends S_idpw_digest and S_auth_token to the client.

Step 3. The client first uses his own $idpw\_digest$ to get rs from the value S_idpw_digest, and computes $rcs = (rs)^a \bmod p = g^{ab} \bmod p$. Then he/she computes $hash(idpw\_digest, rc, rcs)$ and compares it with the received S_auth_token from the server. If it holds, the server is authenticated. Moreover, the client generates a new password new_pw and computes the following four values:

$$C\_auth\_token = Hash(idpw\_digest, rs, rcs) \tag{4}$$

$$idpw\_digest\_new = Hash(id, new\_pw) \tag{5}$$

$$auth\_token\_mask = Hash(idpw\_digest, rcs) \tag{6}$$

$$protected\_idpw\_digest\_new = idpw\_digest\_new \oplus auth\_token\_mask \tag{7}$$

Then, the client sends id, C_auth_token and $protected\_idpw\_digest\_new$ to the server.

Step 4. The server first has to use his own $idpw\_digest$, rc and rcs to compute the $Hash(idpw\_digest, rs, rcs)$, and compare it with the received C_auth_token from the client. If it holds, the client is authenticated and the server then generates $auth\_token\_mask$ to retrieve the $idpw\_digest\_new$ using the "exclusive-or" operation from the value $protected\_idpw\_digest\_new$.

Note that if the client does not want to change his password, the message $protected\_idpw\_digest\_new$ is not needed to be sent to the server in Step 3.

5. Security Analysis and Discussion

First, let us discuss our improvement how to withstand the dictionary attack or guessing attack. The improved schemes are based on the difficulty of computing discrete logarithm problems. The Diffie-Hellman scheme (Diffie and Hellman, 1976) is well known to be representative of them. In Steps 1 and 2, random integers rc and rs are generated under the condition of $rc = g^a \bmod p$ and $rs = g^b \bmod p$, where p is a large prime and g is primitive root. An attacker selects a candidate password pw', and gets rc' and rs' from C_idpw_digest and S_idpw_digest, respectively. However, the attacker can not compute $rcs' = (rs')^a \bmod p = (rc')^b \bmod p = g^{ab} \bmod p$, so he/she can not compute $hash(idpw\_digest, rc', rcs')$ and $hash(idpw\_digest, rs', rcs')$ to validate the candidate password pw'. Therefore, the improved schemes are secure against the dictionary attack.

Let us consider the property of mutual authentication in our improved schemes. Since in Step 3 the client may authenticate the server by comparing whether $hash(idpw\_digest, rc, rcs)$ is equal to S_auth_token or not. On the other hand, the server also authenticates the client by comparing whether $hash(idpw\_digest, rs, rcs)$ is equal to C_auth_token or not. Thus, the client and the server authenticate each other.

Moreover, our improved schemes not only provide functions of both password transmission and password change, but also a session key between the client and the server has been established. The client and the server are able to agree on $rcs = (rs)^a \bmod p = g^{ab} \bmod p$, and compute a session key $hash(rcs)$. The improved schemes can provide the property of perfect forward secrecy (Jablon, 1996) via the Diffie-Hellman scheme (1976). The perfect forward secrecy means that a compromised password does not reveal an old session key. Obviously, a password compromise may reveal $rc = g^a \bmod p$ and $rs = g^b \bmod p$, can not reveal $rcs = g^{ab} \bmod p$.

As we mentioned earlier, some schemes (Botting, 1997; Jablon, 1996; Horng, 1995) encrypt the passwords with symmetric-key or public-key cryptosystems to protect the password while travelling over public networks. Our improved schemes only base on the discrete logarithm problems, but do not use these cryptosystems. In fact, there are several Password-based key agreement schemes (Tseng, 2000; Seo and Sweeney, 1999; Kwon and Song, 1999) via password authentication to provide mutual authentication and the session key establishment. Although these schemes do not adopt any symmetric-key and public-key cryptosystems, but they also do not provide a method for changing an old password to a new password. It is necessary for users to change their old passwords to new passwords.

6. Conclusions

We have presented the security weaknesses on the Peyravian-Zunic's schemes. And we have proposed the improvement on their schemes that overcomes the security weaknesses, which provides mutual authentication between the client and the server, and can withstand the dictionary attack. We have demonstrated that the improvement is based upon the difficulty of calculating discrete logarithms in a finite field.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable and constructive suggestions. Part of this research was supported by National Science Council, Taiwan, R.O.C., under contract no. NSC89–2213-E-252–008.

References

Botting, J. (1997). Security on the internet: Authenticating the user. *Telecommunications*, **31**(12), 77–80.

Diffie, W., M.E. Hellman (1976). New directions in cryptography. *IEEE Trans. on Info. Theory*, **22**(6), 644–654.

Horng, T.L. (1995). Password authentication using triangles and straight lines. *Computers and Mathematics with Applications*, **30**(9), 63–71.

Jablon, D.P. (1996). Strong password only authenticated key exchange. *Computer Communication Review*, **26**(5), 5–26.

Kwon, T., J. Song (1999). Secure agreement scheme for g^{xy} via password authentication. *Electronics Letters*, **35**(11), 892–893.

Morris, D., K. Thompson (1979). Password security: a case history. *Communications of ACM*, **22**(11), 594–597.

Peyravian, M., N. Zunic (2000). Methods for protecting password transmission. *Computers & Security*, **19**(5), 466–469.

Schneier, B. (1996). it Applied Cryptography, 2^{nd} ed., John Wiley & Sons Inc.

Seo, D.H., P. Sweeney (1999). Simple authenticated key agreement algorithm. *Electronics Letters*, **35**(13), 1073–1074.

Tseng, Y.M. (2000). Weakness in a simple authenticated key agreement protocol. *Electronics Letters*, **36**(1), 48–49.

Y.-M. Tseng received the B.S. degree in Computer Science and Engineering from National Chiao Tung University, Taiwan, Republic of China, in 1988; and the M.S. degree in Computer and Information Engineering from National Taiwan University in 1990, and the Ph.D. in Applied Mathematics from National Chung Hsing University in 1999. He is currently an Associate Professor of the Department of Information Management, Nan-Kai College of Technology and Commerce, Taiwan. He is a member of the Chinese Association for Information Security (CCISA). His research interests include cryptography, mobile communication security, network security, and image encryption.

J.-K. Jan was born in Taiwan in 1951. He received the B.S. degree in Physics from Catholic Fu Jen University, Taiwan, Republic of China, in 1974 and the M.S. degree in Information and Computer Science from Tokyo University in 1980. He studied Software Engineering and Human-Computer Interface at the University of Maryland, College Park, MD, during 1984–1986. He is presently a professor of the Department of Applied Mathematics at National Chung Hsing University, Taiwan. He is currently also the editor of Information and Education, and an executive member of the Chinese Association for Information Security. His research interests include computer cryptography, network security, human factors of designing software and information systems, database security, and coding theory.

H.-Y. Chien received the B.S. degree in Computer Science from National Chiao Tung University, Hsinchu, Taiwan, in 1988 and the M.S. degree in Computer and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1990. He is currently pursuing his doctoral degree in applied mathematics at National Chung Hsing University. He is a member of the Chinese Association for Information Security. His research interests include cryptography, network security, electronic commerce.

Apie metodus slaptažodžių persiuntimui apsaugoti

Yuh-Min TSENG, Jinn-Ke JAN, Hung-Yu CHIEN

Peyravian ir Zunic (Peyravian *et al.*, 2000) pasiūlė slaptažodžių persiuntimo ir keitimo schemas neapsaugotuose tinkluose. Jų pasiūlytas sprendinys nereikalauja naudoti jokių simetrinio rakto ar viešo rakto kriptosistemų. Šiame straipsnyje parodyta, kad aukščiau minėtos schemos turi keletą apsaugos "skylių", taikant tas schemas praktikoje. Pasiūlytas būdas šioms apsaugos "skylėms" pašalinti.