Security Enhancement for Protecting Password Transmission*

Chou-Chen Yang,[§] Ting-Yi Chang,[§] Jian-Wei Li,[§] Min-Shiang Hwang[†‡]

Graduate Institute of Networking and Communication Engineering[†]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@cyut.edu.tw
Fax: 886-4-23742337

Department of Information Management[‡]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.

Department of Information and Communication Engineering[§]
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.

September 19, 2012

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC91-2213-E-324-003.

‡Responsible for correspondence: Prof. Min-Shiang Hwang.

Security Enhancement for Protecting Password Transmission

Abstract

In 2002, Hwang and Yeh proposed some improved schemes to mend several security flaws in the Peyravian-Zunic password transmission scheme and password change scheme. However, this article will point out that there still exist some security flaws in the Hwang-Yeh schemes; at the same time, we shall also propose some improved versions of their schemes.

Keywords: Authentication, cryptography, password, security.

1 Introduction

The rapid growth of networks in both number and size encourages more and more computers to link together to share various resources and exchange huge amounts of information. With more and more participants joining in, network security has become a more and more important issue. Various resources distributed among the hosts are shared across the network in the form of network services provided by servers. Before users request for services, they should pass the identity authentication by the servers. The password authentication schemes [1, 5, 7, 9, 11, 12, 13] are simple and practical solutions to authenticating users' identities by servers.

In 2000, Peyravian and Zunic [14] offered a pair of schemes that allowed users to transmit passwords over public networks and arbitrarily change their passwords, and the security was based on the one-way hash function. Unfortunately, Tseng et al. [17] have pointed out that the Peyravian-Zunic schemes are vulnerable to the guessing attack and can be frustrated by server spoofing, and they have used the Diffie-Hellman key agreement scheme [3] to help

mend the above security flaws. On the other hand, Hwang and Yeh [6] have also pointed out that the Peyravian-Zunic schemes suffer from not only the guessing attack and server spoofing but also server data eavesdropping. They have also proposed their improved schemes by using the server's public key cryptosystem such as RSA [2, 10, 15] and ElGamal [4, 8] cryptosystems to overcome the security flaws in the Peyravian-Zunic schemes.

However, there are still some security flaws in Hwang and Yeh's password change scheme. Any adversary can intercept the request for changing passwords sent by a legal user and modify it with a wrong password. As a result, the user will not be able to successfully login the server next time. Furthermore, their schemes are vulnerable to server data eavesdropping. If anyone obtains the secret data stored in the server, he/she can easily impersonate a legal user to login the server. In this paper, we will show the security flaws and enhance the security in the Hwang-Yeh schemes.

The organization of the article is as follows. In the next section, we will briefly review the Hwang-Yeh schemes. In Section 3, the security flaws will be shown. In Section 4, we will propose our improvement to enhance the security of their schemes and analyze the security. Finally, we shall conclude this article in Section 5.

2 Review of the Hwang-Yeh Schemes

The main difference between Hwang-Yeh password transmission scheme and password change scheme is that in the latter the client sends a password change request to the server. In the system, the client (or the user) has an identity id and a corresponding password pw. The password is known only to both of the client and the server. The server stores $v\_pw = H(pw)$ instead of pw for each client in the database, where $H(\cdot)$ is a secure one-way hash function. Here, we only present the password transmission scheme.

Step 1. Client$\longrightarrow$Server: id, C_cipher

The client encrypts the random number rc and along with pw with the server's public key PK_S denoted as $C\_cipher = (rc, pw)PK_S$ and send it with id as a login request to the server.

Step 2. Server$\longrightarrow$Client: $S\_auth\_token, S\_rs\_digest$

The server decrypts C_cipher to obtain rc and pw by using its private key. Then, the server computes the hash value $H(pw)$ and checks whether $H(pw) = v\_pw$ holds or not. If it holds, the server chooses a random number rs and computes $S\_auth\_token = rs \oplus rc$ and $S\_rs\_digest = H(rs)$, where $\oplus$ is the exclusive-or (XOR) operation. Then, the server sends $\{S\_auth\_token, S\_rs\_digest\}$ to the client.

Step 3. Client$\longrightarrow$Server: $id, C\_auth\_token, C\_digest\_new\_pw$

The client retrieves rs by computing $S\_auth\_token \oplus rc$ and then verifies the consistency between the retrieved rs and the received S_rs_digest. If the result is positive, the client chooses a new password new_pw and computes $C\_auth\_token = H(rc, rs)$ and $C\_digest\_new\_pw = H(new\_pw) \oplus H(rc{+}1, rs)$. Finally, the client sends $\{id, C\_auth\_token, C\_digest\_new\_pw\}$ to the server.

Step 4. Server$\longrightarrow$Client: access granted or access denied

The server computes the hash value $H(rc, rs)$ and checks whether $H(rc, rs) = C\_auth\_token$ holds or not. If it holds, the server can obtain $H(new\_pw)$ by computing $C\_digest\_new\_pw \oplus H(rc + 1, rs)$ and then store $v\_pw = H(new\_pw)$ in the database.

Obviously, by employing the public key cryptosystem on the server's side to protect the transmitted password, Hwang and Yeh have effectively avoided the guessing attack and server spoofing that treated the Peyravian-Zunic schemes.

3 Security Flaws of the Hwang-Yeh Schemes

In this section, we will show Hwang and Yeh's password change scheme is vulnerable to the modification attack, and both the password transmission scheme and the password change scheme are vulnerable to server data eavesdropping.

Modification attack:

Upon seeing $\{id, C\_auth\_token, C\_digest\_new\_pw\}$ sent by the client in Step 3, the adversary replaces $C\_digest\_new\_pw$ with a random number ra. After receiving $\{id, C\_auth\_token, ra\}$, the server first computes the hash value $H(rc, rs)$ and checks whether $H(rc, rs) = C\_auth\_token$ holds or not. Since C_auth_token is computed by the client, the equation $H(rc, rs) = C\_auth\_token$ checked by the server will turn out positive. Then, the server computes $ra \oplus H(rc + 1, rs)$ and stores $v\_pw = ra \oplus H(rc + 1, rs)$ in place of $H(pw)$ in the database.

However, the client is under the impression that it has successfully changed from an old password pw to a new password new_pw. Once the client logins to the server the next time, it sends $\{id, C\_cipher = (rc, new\_pw)PK_S\}$ to the server in Step 1. In Step 2, the server decrypts the message to obtain rc and new_pw with its private key. Then, the server computes the hash value $H(new\_pw)$ and check whether $H(new\_pw) = v\_pw$ holds or not. However, $H(new\_pw)$ is not equal to v_pw because $v\_pw = ra \oplus H(rc + 1, rs)$. The server will reject the client's login request.

Server data eavesdropping:

Hwang and Yeh claimed that their schemes could avoid the security flaw where the secret data $v\_pw = H(pw)$ gets eavesdropped by the attacker who desires forge the login request to pass the authentication. In practice, people would find the long random string password difficult to use and remember. It would

be much more user-friendly if the password is a meaningful string that people can recognize easily such as a natural language phrase. On the other hand, however, natural language phrases narrow down the possibilities for attackers. When an attacker somehow acquires the secret data $v\_pw = H(pw)$ stored in the server, she/he can verify the correctness of the guessed password $guess\_pw$ by checking whether $H(guess\_pw) = v\_pw$ holds or not. If the passwords are guessed right, she/he can easily forge the login request to pass the authentication.

4 The Improvement on the Hwang-Yeh Schemes

In this section, we will propose an improved scheme to overcome the previously shown security flaws in the Hwang-Yeh schemes and analyze the security of our improved schemes.

4.1 The Improved Schemes

To overcome the problem of server data eavesdropping in the Hwang-Yeh schemes, the server stores $v\_pw = MAC_K(pw)$ instead of $H(pw)$ for each client in the database, where $MAC_K(\cdot)$ is a MAC generation function and the keying materials as its key K [16]. The password change protection scheme has the following steps.

Step 1. Client$\longrightarrow$Server: id, C_cipher

The client encrypts the random number rc along with pw and new_pw with the server public key PK_S as $C\_cipher = (rc, pw, new\_pw)PK_S$ and its id as a login request to the server.

Step 2. Server$\longrightarrow$Client: $S\_auth\_token, S\_rs\_digest$

The server decrypts C_cipher to obtain rc, pw and new_pw with its private key. Then, the server computes $MAC_K(pw)$ and check whether $MAC_K(pw) = v\_pw$ holds or not. If it holds, the server chooses a random

number rs and computes $S\_auth\_token = rs \oplus rc$ and $S\_rs\_digest = H(rs)$. Then, the server sends $\{S\_auth\_token, S\_rs\_digest\}$ to the client.

Step 3. Client$\longrightarrow$Server: id, C_auth_token

The client retrieves rs by computing $S\_auth\_token \oplus rc$ and then verifies the consistency between the retrieved rs and the received S_rs_digest. If the result is positive, the client computes $C\_auth\_token = H(rc, rs)$. Finally, the client sends $\{id, C\_auth\_token\}$.

Step 4. Server$\longrightarrow$Client: access granted or access denied

The server first computes the hash value $H(rc, rs)$ and then checks whether $H(rc, rs) = C\_auth\_token$ holds or not. If it holds, the server can ensure the client is legal. Then, the new_pw decrypted in Step 1 will be stored as $v\_pw = (H(pw))K_S$ in the server's database.

The difference between our password transmission scheme and password change scheme is that in the latter one the client encrypts new_pw in Step 1 to the server.

4.2 Security Analysis

In this section, several attacks will be analyzed to demonstrate the security of our improved schemes.

Replay attack:

The attacker intercepts $\{id, C\_cipher\}$ sent by the client in Step 1 and uses it to masquerade as the client to send the login request the next time. However, the random number rs generated by the server is different every time, and the random number rc is protected under the server's public key. Without knowing rc and rs, the attacker cannot make a correct response C_auth_token to the server in Step 3.

Guessing attack:

The random numbers rc and rs separately generated by the client and the server are protected as $C\_cipher = (rc, pw)PK_S$ and $S\_auth\_token = rs \oplus rc$. No one can reveal rc from C_cipher without knowing the server's private key. Hence, the attacker cannot verify the correctness of the guessed password by checking $(rc, guess\_pw)PK_S = C\_cipher$. For the same reason, in the password change scheme, the guessing attack will also fail.

Server spoofing:

Just like Hwang-Yeh schemes, our improved schemes use the server's public key to ensure that only the real server can decrypt C_cipher. Only a real server can obtain rc from C_cipher and use it to compute the response in Step 2. After verifying the identity of the server, the client sends C_auth_token to the server to achieve mutual authentication.

Modification attack:

The messages C_auth_token and $C\_digest\_new\_pw$ in the Hwang-Yeh password change scheme are used by the server to authenticate the client and obtain the new password, respectively. Only when the adversary modifies $C\_digest\_new\_pw$ to a random number and uses the original C_auth_toke generated by the client can he/she pass the authentication. In our improved scheme, the new password new_pw is also encrypted by using the server's public key in Step 1. Therefore, the adversary is unable to choose a random number to replace new_pw.

Server data eavesdropping:

We know that the servers are always the targets of attack. The attacker may

acquire $v\_pw = MAC_K(pw)$ stored in the server. However, without knowing the secret key K, the attacker cannot forge the login request to pass the authentication because pw is encrypted by the server's secret key K_S and therefore the correctness of the guessed password cannot be verified by checking $MAC_K(guess\_pw) = v\_pw$.

5 Conclusion

In this article, we have shown that the Hwang-Yeh password change scheme is vulnerable to the modification attack, and both their password transmission scheme and password change scheme are weak against server data eavesdropping. In addition, we have proposed our improved schemes to effectively avoid the security flaws in the Hwang-Yeh schemes.

References

[1] J. Botting, "Security on the Internet: Authenticating the user," *Telecommunications*, vol. 31, no. 12, pp. 77–80, 1997.

[2] S. Wesley Changchien, Min-Shiang Hwang, and Kuo-Feng Hwang, "A batch verifying and detecting multiple RSA digital signatures," *International Journal of Computational and Numerical Analysis and Applications*, vol. 2, no. 3, pp. 303–307, 2002.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[4] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[5] Gwoboa Horng, "Password authentication without using password table," *Information Processing Letters*, vol. 55, pp. 247–250, 1995.

[6] Jing-Jang Hwang and Tzu-Chang Yeh, "Improvement on Peyravian-Zunic's password authentication schemes," *IEICE Trans. on Communications*, vol. E85-B, pp. 823–825, 2002.

[7] Min-Shiang Hwang, "A remote password authentication scheme based on the digital signature method," *International Journal of Computer Mathematics*, vol. 70, pp. 657–666, 1999.

[8] Min-Shiang Hwang, Chin-Chen Chang, and Kuo-Feng Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.

[9] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *International Journal of Informatica*, vol. 12, no. 2, pp. 297–302, 2001.

[10] Min-Shiang Hwang, Iuon-Chung Lin, and Kuo-Feng Hwang, "Cryptanalysis of the batch verifying multiple RSA digital signatures," *Informatica*, vol. 11, no. 1, pp. 15–19, 2000.

[11] D. P. Jablon, "Strong password only authenticated key exchange," *Computer Communication Review*, vol. 26, pp. 5–26, Oct. 1996.

[12] Cheng-Chi Lee, Min-Shiang Hwang, and Wei-Pang Yang, "A flexible remote user authentication scheme using smart cards," *ACM Operating Systems Review*, vol. 36, no. 3, pp. 46–52, 2002.

[13] Li-Hua Li, Iuon-Chung Lin, and Min-Shiang Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[14] M. Peyravian and N. Zunic, "Methods for protecting password transmission," *Computers & Security*, vol. 19, no. 5, pp. 466–469, 2000.

[15] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[16] Bruce Schneier, *Applied Cryptography, 2nd Edition*. New York: John Wiley & Sons, 1996.

[17] Yuh-Min Tseng, Jinn-Ke Jan, and Hung-Yu Chien, "On the security of methods for protecting password transmission," *International Journal of Informatica*, vol. 12, no. 2, pp. 1–8, 2001.