A Remote User Authentication Scheme Using Hash Functions

Cheng-Chi Lee[‡] Li-Hua Li[†] Min-Shiang Hwang[†]
Department of Information Management, Chaoyang University of Technology[†]
168, Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.
Email: mshwang@mail.cyut.edu.tw
Department of Computer and Information Science, National Chiao-Tung University[‡]
1001 Ta Hsueh Road, Hsinchu 300, Taiwan, R.O.C.

Abstract

Recently, Peyravian and Zunic proposed a simple but efficient password authentication system. Their scheme is based on the collision-resistant hash function, such as SHA-1. Their scheme did not use any cryptosystems (such as DES, RSA, etc.). However, their scheme is vulnerable to guess attack. An attacker can easily obtain a user's password by guessing attack and then impersonate the user to login and access resources in the server. To overcome the vulnerability of their scheme, we propose an improved scheme to enhance security of their scheme in this article.

Keywords: Cryptography, digest, guessing attack, hash function, password.

I Introduction

When a user wants to login and access resource in a remote server, the user must be authenticated that he/she is a legal user. In general, password authentication techniques are used to authenticate whether the user is legal or not. In these techniques, each user has to remember a pair words: user identity id and password pw. When a user wants to login and access resources in a server via networks (Internet), he/she enters his/her id and pw to a client. The client transmits the user's id and pw to the server via Internet. The server then verifies the user's identity id and pw by comparing these words in server's password table. If it holds, the server allows the user login and access resources in the server. However, the transmitted messages over insecure Internet are vulnerable to relatively easy access by an attacker [4, 6, 9, 19]. Without appropriate protection, user's passwords pw are susceptible to intercept by an attacker [7, 10, 16].

Many literatures [1, 2, 3, 5, 8, 11, 12, 13, 14] had been proposed to protect the password over an insecure networks. These schemes are based on cryptosystems, such as DES, RSA, and ElGamal [19], etc.. Recently, Peyavian and Zunic [18] proposed a scheme for protecting passwords while being transmitted and changed over insecure networks. Their scheme is based on the collision-resistant hash function, such as SHA-1 [19]. Their scheme is simple and efficient to implement. However, their scheme is vulnerable to guess attack. An attacker can obtain a user's password by guessing attack. The attacker then impersonates the user to login and access resources in the server. The guessing attack is computationally feasible because most passwords are a meaningful (in dictionary), or short string (i.e., 8 bytes) and easy to be memorized. The guessing attack can be performed off-line [15, 17].

In this article, we show that how guess attack to Peyavian and Zunic's scheme. In order to resist the guessing attack in their scheme, we propose an improved scheme to enhance security. Our scheme is based on the hash functions.

II The Weakness of Peyravian And Zunic's Scheme

Peyavian and Zunic [18] proposed a scheme of protected password transmission over insecure networks. In their scheme, all users can randomly choose user's identity and password. There are three phases in their scheme: registration, user authentication, and change password phases.

In registration phase, each user U_i must to randomly choose identity ID_i and password PW_i, and then calculate a password digest $HPW_i = H(ID_i, PW_i)$, where $H(\cdot)$ is a collision-resistant hash function, such as SHA-1 [19]. U_i sends ID_i and HPW_i to a server. The server stores ID_i and HPW_i in verification table.

In user authentication phase, a user logins to a server for accessing resources; and the server authenticates the user whether legal or not. The procedures of this phase are shown in Figure 1.

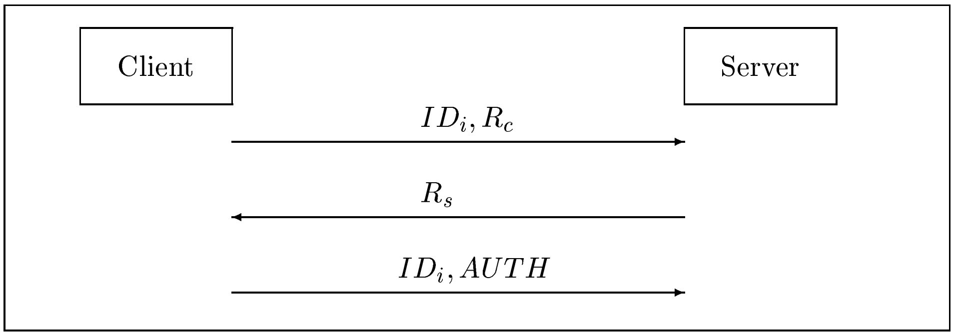

Figure 1: The user authentication phase of Peyravian and Zunic's scheme

Step 1. The user enters his/her identity ID_i and password PW_i to the client. The client randomly chooses a number R_c and then sends ID_i and R_c to server.

Step 2. The server randomly chooses a number R_s and sends it to the client.

Step 3. After receiving R_s, the client calculates $HPW_i = H(ID_i, PW_i)$ and $AUTH = H(HPW_i, R_c, R_s)$. The client sends ID_i and $AUTH$ to the server.

Step 4. After receiving ID_i and $AUTH$, the server retrieves the user's HPW_i from the verification table. The server calculates $H(HPW_i, R_c, R_s)$. Next, the server compares it with $AUTH$ which receives from the client. If it is equal, the server accepts the accessing resources request. Otherwise, it rejects the client's request.

In change password phase, each user can freely change his/her password over insecure networks. The procedures of this phase are shown in Figure 2. Steps 1 and 2 are the same as the user authentication phase. We assume that U_i wants to change his/her password PW_i to $NewPW_i$.

Step 3'. After receiving R_s, the client calculates $HPW_i = H(ID_i, PW_i)$, $NewHPW_i = H(ID_i, NewPW_i)$, $AUTH = H(HPW_i, R_c, R_s)$, and $Mask = NewHPW_i \oplus H(HPW_i, R_c + 1, R_s)$. The client sends ID_i, $AUTH$, and $Mask$ to the server.

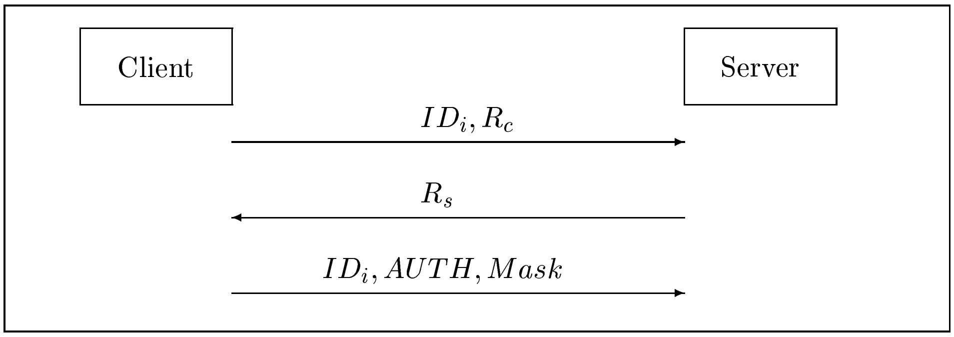

Figure 2: The change password phase of Peyravian and Zunic's scheme

Step 4'. After receiving ID_i, $AUTH$, and $Mask$, the server retrieves the user's HPW_i from the verification table. The server calculates $H(HPW_i, R_c, R_s)$ and compares it with $AUTH$ which receives from the client. If it is equal, the server accepts the client to change the user's password. Next, the server obtains new password digest $NewHPW_i$ as follows: $NewHPW_i = Mask \oplus H(HPW_i, R_c + 1, R_s)$. Then the server replaces the old HPW_i with the new password digest $NewHPW_i$ in verification table.

Next, we discuss the guessing attack in Peyavian and Zunic's scheme. An unauthorized user can intercept the messages ID_i, R_c, R_s, and $AUTH$ of user U_i over an insecure network. The unauthorized user can then guess a candidate password PW' and calculate $AUTH' = H(H(ID_i, PW'), R_c, R_s)$. If $AUTH'$ is equal to $AUTH$, the user's password PW_i is guessed. Otherwise, the unauthorized user try the next candidate password until the $AUTH'$ is equal to $AUTH$. Since most user's passwords are meaningful (in dictionary) or short string (i.e. 8 bytes) for easy memory, the guessing attack is computationally feasible.

Once the unauthorized user obtains the authorized user's password PW_i, he/she can impersonate the authorized user U_i to login and access resources in the server. To remedy the guessing attack, we propose an improvement of their scheme in next section.

III Our Improved Scheme

In this section, we propose an improvement of Peyravian and Zunic's scheme. Our scheme is based on the hash functions. There are also three phases in our scheme: registration, user authentication, and change password phases. The registration phase is the same as that of Peyravian and Zunic's scheme. In user authentication phase, the procedures of the improved scheme are shown in Figure 3.

Step 1. The user U_i enters his/her identity ID_i and password PW_i to the client. The client calculates $HPW_i = H(ID_i, PW_i)$, where $H(\cdot)$ is a hash function with output message in 512 bits length. The client randomly chooses a number R_c with 512 bits length and sends ID_i and $R_c \oplus HPW_i$ to the server.

Step 2. After receiving ID_i and $R_c \oplus HPW_i$, the server retrieves the user's password digest HPW_i from the verification table and then obtains R_c by computing $(R_c \oplus HPW_i) \oplus HPW_i$. Next, the server randomly generates a number R_s with 512 bits length and

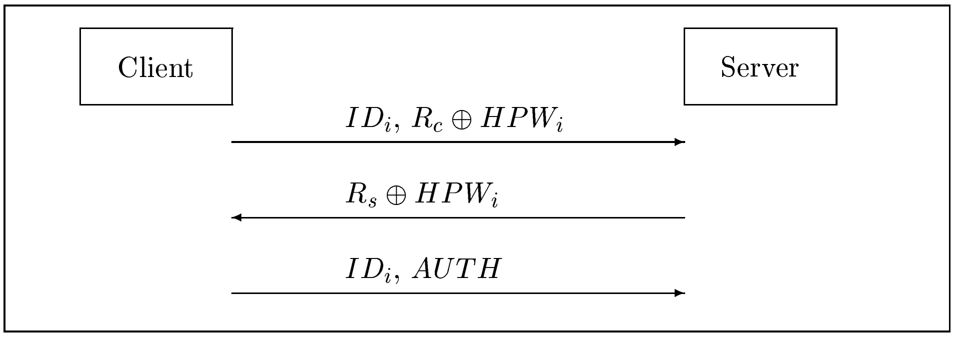

Figure 3: The user authentication phase of our improved scheme

calculates $R_s \oplus HPW_i$ and $AUTH' = H(HPW_i, R_c, R_s)$. The server sends $R_s \oplus HPW_i$ to the client.

Step 3. After receiving $R_s \oplus HPW_i$, the client retrieves R_s by computing $(R_s \oplus HPW_i) \oplus HPW_i$. Next, the client calculates

$$AUTH = H(HPW_i, R_c, R_s). \tag{1}$$

The client sends ID_i and $AUTH$ to the server.

Step 4. After receiving ID_i and $AUTH$, the server compares $AUTH$ with $AUTH'$. If it is equal, the server accepts the accessing resources request. Otherwise, it rejects the client's request.

In change password phase, the procedures of the improved scheme are shown in Figure 4. Steps 1 and 2 are the same as that of the above user authentication phase. We assume that U_i wants to change his/her password PW_i to $NewPW_i$.

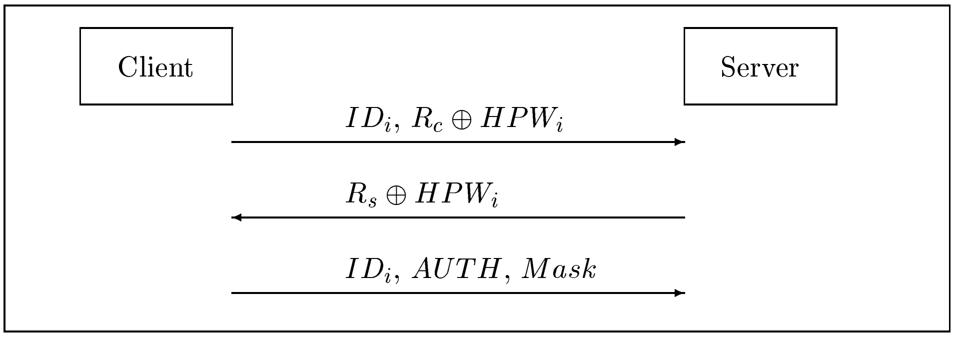

Figure 4: The change password phase of our improved scheme

Step 3'. After receiving $R_s \oplus HPW_i$, the client retrieves R_s by computing $(R_s \oplus HPW_i) \oplus HPW_i$. Next, the client calculates

$$\begin{aligned} NewHPW_i &= H(ID_i, NewPW_i), && (2)\\ AUTH &= H(HPW_i, R_c, R_s), && (3)\\ Mask &= NewHPW_i \oplus H(HPW_i, R_c + 1, R_s). && (4) \end{aligned}$$

The client sends ID_i, $AUTH$, and $Mask$ to the server.

Step 4'. After receiving ID_i, $AUTH$, and $Mask$, the server retrieves the user's HPW_i from the verification table, and then compares $AUTH$ with $AUTH'$. If it is equal, the server accepts the client to change the user's password and obtains new password digest $NewHPW_i$ as follows:

$$NewHPW_i = Mask \oplus H(HPW_i, R_c + 1, R_s). \quad (5)$$

Then the server replaces the old HPW_i with the new password digest $NewHPW_i$ in verification table.

IV Security Analysis

It is difficult to guess user's password in our improved scheme without knowing R_c, R_s, and the user's password digest HPW_i. If an attacker wants to obtain PW_i, he/she must to guess the triple values R_c, R_s, and PW_i simultaneously such that $AUTH = H(H(ID_i, PW_i), R_c, R_s)$. Since R_c and R_s are a long random number (i.e., 512 bits in length), the probability of guessing R_c, R_s, and PW_i is less than $\frac{1}{2^{512}} \times \frac{1}{2^{512}} \times \frac{1}{2^{64}}$ (assume that the user's password PW_i is 8 bytes in length).

In addition, an attacker may obtain $(R_c \oplus HPW_i)$ and $(R_s \oplus HPW_i)$ from Steps 1 and 2 in the user authentication phase, respectively. The attacker only can obtain $R_c \oplus R_s$ by computing $(R_c \oplus HPW_i) \oplus (R_s \oplus HPW_i)$. It is also difficult to guess R_c and R_s without knowing HPW_i. The probability of guessing R_c and R_s is less than $\frac{1}{2^{512}} \times \frac{1}{2^{512}}$. Since R_c and R_s are unknown to an unauthorized user, any attacker cannot guess the user's password PW_i.

Since attackers cannot obtain user's password PW_i, he/she cannot impersonate the user U_i to login and access resources in the server. Therefore, our improved scheme is more secure than Peyravian and Zunic's scheme.

V Conclusion

We have shown that there is a weakness in Peyravian and Zunic's scheme. An unauthorized user can use guessing attack to obtain an authorized user's password and then login and access resources in the server. To remedy the weakness, we have proposed an improvement of Peyravian and Zunic's scheme in Section III. Our improved scheme is based on the hash functions.

ACKNOWLEDGEMENTS

Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC90-2213-E-324-005.

References

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

[2] C. C. Chang and S. J. Hwang, "Using smart cards to authenticate remote passwords," *Computers and Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[3] C. C. Chang and W. Y. Liao, "A remote password authentication scheme based upon ElGamal's signature scheme," *Compuper & Security*, vol. 13, no. 2, pp. 137–144, 1994.

[4] Chin-Chen Chang and Min-Shiang Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[5] S. Halevi and H. Krawczyk, "Public key cryptography and password protocols," in *Proceedings of 5th ACM Conference on Computer and Communications Security*, pp. 122–131, 1998.

[6] Min-Shiang Hwang, "A new redundancy reducing cipher," *International Journal of Informatica*, vol. 11, no. 4, pp. 435–440, 2000.

[7] Min-Shiang Hwang, "Cryptanalysis of remote login authentication scheme," *Computer Communications*, vol. 22, no. 8, pp. 742–744, 1999.

[8] 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.

[9] 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, 2002.

[10] 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.

[11] Min-Shiang Hwang and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

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

[13] J. K. Jan and Y. Y. Chen, "'paramita wisdom' password authentication scheme without verification tables," *The Journal of Systems and software*, vol. 42, pp. 45–57, 1998.

[14] Cheng-Chi Lee, Min-Shinag Hwang, and Wei-Pang Yang, "A Flexible Remote User Authentication Scheme Using Smart Car," *ACM Operating Systems Review*, vol. 36, no. 3, pp. 46–52, 2002.

[15] G. Li, M. A. Lomas, R. M. Needham, and J. H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 648–656, June 1993.

[16] 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 Networls*, vol. 12, no. 6, pp. 1498–1504, 2001.

[17] R. Morris and K. Thompson, “Password security: A case history,” *Communications of the ACM*, vol. 22, pp. 594–597, Nov. 1979.

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

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