Abstract. Recently, Tseng et al. proposed an improvement on Peyravian and Zunic's protected password transmission scheme and protected password changing scheme to remove some security flaws. However, as we will point out in this paper, any adversary can intercept the request for changing the password sent by a legal user and modify it with a wrong password. Furthermore, we shall also propose an improved version of their protected password changing scheme to help it out of the trouble.

Key words: Authentication, cryptography, discrete logarithm, password.

1. Introduction
The rapid growth of networks in both number and size encourages more and more computers to link together for sharing various kinds of data and exchanging huge amounts of information. Various resources distributed among the hosts are shared across the network in the form of network services provided by servers. Before providing such services, servers should have the ability to authenticate the users. Otherwise, any adversary can easily impersonate a legal user to login the server. Thus, user authentication is one of the most important security services for secure communication.
Password-based authentication schemes are the most widely used methods for user authentication since they allow people to choose their own passwords without any devices. In practice, people hardly find long random string passwords easy 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. However, the easy-to-remember passwords narrow down the possibilities and make it easier for attackers to succeed. Most of such schemes (Botting, 1997, Horng, 1995, Hwang, 1999, Hwang et al., 2001, Jablon, 1996, Lee et al., 2002, Li et al., 2001) require symmetric or asymmetric cryptosystems to encrypt the passwords before traveling them over public networks. In 2000, Peyravian and Zunic (Peyravian and Zunic, 2000) proposed a pair of schemes that allow users to transmit passwords over public networks and arbitrarily change their passwords without employing any symmetric cryptosystem (such as DES (Smid, 1988), Rindael (Daemen and Rijmen, 2001), and others (Hwang, 2000, Hwang et al., 2002b)) or asymmetric cryptosystem (such as RSA (Changchien et al., 2002, Hwang, 2000, Rivest et al., 1978), ElGamal (ElGamal, 1985, Hwang et al., 2002a), etc.).

Unfortunately, Tseng et al. (Tseng et al., 2001) pointed out that Peyravian and Zunic’s schemes are vulnerable to such attacks as the guessing attack and server spoofing. They proposed improved schemes based on the Diffie-Hellman key agreement scheme (Diffie and Hellman, 1976) to withstand these attacks. At the same time, Hwang and Yeh (Hwang and Yeh, 2002) also pointed out that those attacks threaten the security of Peyravian and Zunic’s schemes and also repaired the security flaws in their schemes based on the server’s asymmetric cryptosystem. However, to avoid the adversary from replacing the original public keys of the servers with her/his own, certificates (e.g. digital signatures) should be introduced in Hwang and Yeh’s schemes. A certificate from a trusted authority is what servers with public keys will ask for before providing services. This means users need large storage spaces for certificates hard to remember, and more bandwidth is also needed for verifying the signatures.

In this paper, we will point out Tseng et al.’s protected password changing scheme is vulnerable to the modification attack; that is, any adversary can intercept the request for changing passwords sent by a legal user and modify it with a wrong password. At the same time, we will propose an improvement on their scheme without using any symmetric or asymmetric cryptosystems.

The organization of the paper is as follows. In the next section, we will briefly review Tseng et al.’s protected password changing scheme, and the modification attack will also be introduced. In Section 3, we will propose our improved version to enhance the security of their scheme. In Section 4, we will analyze
The client first recovers and computes the server first recovers

Section 5.

2. Modification Attack on Tseng et al.’s Scheme

In this section, we shall first briefly review Tseng et al.’s protected password changing scheme and then show how the modification attack can work on their scheme. In the system, the server publishes two large prime numbers $p$ and $q$ such that $q|p - 1$. Let $g$ be a generator with order $q$ in the Galois field $GF(p)$, which is based on the Diffie-Hellman assumption. The client (user) has the identity $id$ and the corresponding password $pw$. The password $pw$ is only secretly known to both the client and the server. The server employs a one-way hash function $H(\cdot)$ to store $id$ and $pw$ as $verify_{idpw\_digest} = H(id, pw)$ in the database. The protected password changing scheme works as follows:

Step 1. Client $\rightarrow$ Server: $id, C_{idpw\_digest}$

The client chooses a random number $c \in [1, q - 1]$ and computes $rc = g^c \mod p$. Then, it computes $C_{idpw\_digest} = H(id, pw) \oplus rc$ ($\oplus$ denotes the exclusive operator) and sends it along with $id$ to the server.

Step 2. Server $\rightarrow$ Client: $S_{idpw\_digest}, S_{auth\_token}$

The server first recovers $rc$ from $C_{idpw\_digest}$ by computing $C_{idpw\_digest} \oplus verify_{idpw\_digest}$. Then, the server chooses a random number $s \in [1, q - 1]$ and computes $rs = g^s \mod p$ and $rcs = (rc)^s = g^{cs} \mod p$. Next, the server computes $S_{idpw\_digest} = verify_{idpw\_digest} \oplus rs$ and $S_{auth\_token} = H(verify_{idpw\_digest}, rcs, rc)$, and then it sends them to the client.

Step 3. Client $\rightarrow$ Server: $id, C_{auth\_token}, C_{new\_idpw\_digest}$

The client first recovers $rs$ from $S_{idpw\_digest}$ by computing $S_{idpw\_digest} \oplus H(id, pw)$. Then it computes $rcs = (rcs)^c = g^{cs} \mod p$ and uses it together with its own $H(id, pw)$ and $rc$ to compute $H(H(id, pw), rcs, rc)$, which is then compared with the received $S_{auth\_token}$ from the server. If they match, the client computes $C_{auth\_token} = H(H(id, pw), rcs, rs)$. Then, the client chooses a new password $new\_pw$ and computes $C_{new\_idpw\_digest} = H(id, new\_pw) \oplus H(id, new\_pw, rcs)$. Finally, the client sends $\{id, C_{auth\_token}, C_{new\_idpw\_digest}\}$ to the server.

Step 4. The server uses its own $verify_{idpw\_digest}, rcs$ and $rs$ to compute $H(verify_{idpw\_digest}, rcs, rs)$ and compares it with the received $C_{auth\_token}$. If they match, the server recovers $H(id, new\_pw)$ from $C_{new\_idpw\_digest}$. Finally, the server sends $\{id, new\_pw\}$ to the client.
The client chooses a random number \( idpw_{\text{digest}} \) by computing \( C_{\text{new idpw digest}} @ H(\text{verify idpw digest, res}) \) and then stores \( \text{verify idpw digest} = H(id, new_pw) \) in the database.

The difference between the protected password transmission scheme and protected password changing scheme is that the client additionally sends \( C_{\text{new idpw digest}} \) to the server for changing passwords in the latter scheme.

Note that the client sends \( \{ id, C_{\text{auth token}}, C_{\text{new idpw digest}} \} \) in Step 3 to the server, and the messages \( C_{\text{auth token}} \) and \( C_{\text{new idpw digest}} \) are used to enable the server to authenticate the client and to obtain \( H(id, new_pw) \), respectively. However, because the two messages are separated, the attacker can replace \( C_{\text{new idpw digest}} \) with a random number \( ra \). After receiving \( \{ id, C_{\text{auth token}}, ra \} \), the server checks the validity of \( C_{\text{auth token}} \). Since \( C_{\text{auth token}} \) is generated by the legal client, the server will accept it. Then, the server computes \( ra @ H(\text{verify idpw digest, res}) \) and stores \( \text{verify idpw digest} = ra @ H(\text{verify idpw digest, res}) \) in the database.

Unfortunately, the client is mistakenly convinced that it has successfully changed from the old password \( pw \) to a new password \( new_pw \). When the client tries to login the server the next time, the server will reject the client’s login request because the client cannot recover \( rs \) from \( S_{\text{idpw digest}} \) by computing \( S_{\text{idpw digest}} @ H(id, new_pw) \) and therefore cannot compute \( C_{\text{auth token}} \) correctly. As a result, the server will conclude that the client is illegal, and the client will not be able to change its password successfully.

3. The Improved Scheme

To get rid of the previously shown security flaw, we shall propose an improved protected password-changing scheme as follows:

Step 1. Client—\( \rightarrow \) Server: \( id, C_{\text{idpw digest}}, C_{\text{new idpw digest}} \)

The client chooses a random number \( c \in [1, q - 1] \) and computes \( rc = g^c \mod p \). Then, the client chooses a new password \( new_pw \) and uses \( rc \) to compute \( C_{\text{idpw digest}} = H(id, pw) @ rc \) and \( C_{\text{new idpw digest}} = H(id, new_pw) @ rc \). The client sends \( \{ id, C_{\text{idpw digest}}, C_{\text{new idpw digest}} \} \) to the server.

Step 2. Server—\( \rightarrow \) Client: \( S_{\text{idpw digest}}, S_{\text{auth token}} \)

The server first recovers \( rc \) from \( C_{\text{idpw digest}} \) by computing \( C_{\text{idpw digest}} @ \text{verify idpw digest} \). Then, the server uses the recovered \( rc \) to obtain \( H(id, new_pw) \) from \( C_{\text{new idpw digest}} \) by computing \( C_{\text{new idpw digest}} @ rc \). Then, the server chooses a random number \( s \in [1, q - 1] \) and computes \( rs = g^s \mod p \)
and \( rcs = (rc)^s = g^{cs} \mod p \). Then, the server uses its own \( rs \), \( rcs \) and the recovered \( rc \) to compute \( S_{idpw\_digest} = verify\_idpw\_digest \oplus rs \) and \( S_{auth\_token} = H(verify\_idpw\_digest, rcs, rc) \). The server sends \( \{S_{idpw\_digest}, S_{auth\_token}\} \) to the client.

Step 3. Client\textarrowright Server: \( id, C_{auth\_token} \)

The client first recovers \( rs \) from \( S_{idpw\_digest} \) by computing \( S_{idpw\_digest} \oplus H(id, pw) \). Then it computes \( rcs = (rs)^c = g^{cs} \mod p \) and uses it together with its own \( H(id, pw) \) and \( rc \) to compute \( H(H(id, pw), rcs, rc) \), which is then compared with the received \( S_{auth\_token} \) from the server. If they hold, the server is authenticated and the client uses its own \( rcs, H(id, new\_pw) \) and the recovered \( rs \) to compute \( C_{auth\_token} = H(H(id, pw), rcs, rs) \oplus H(id, new\_pw) \). Finally, the client sends \( \{id, C_{auth\_token}\} \) to the server.

Step 4. The server uses its own \( verify\_idpw\_digest, rcs \) and \( rs \) to compute \( H(verify\_idpw\_digest, rcs, rs) \). Then, the server uses the recovered \( H(id, new\_pw) \) in the Step 1 to compute \( C_{auth\_token} \oplus H(id, new\_pw) \) and compares the result with \( H(verify\_idpw\_digest, rcs, rs) \). If they hold, the server stores \( verify\_idpw\_digest = H(id, new\_pw) \) in the database.

Obviously, our improved scheme does not employ any symmetric or asymmetric cryptosystems and does not increase the client’s or server’s computational complexity. Without sending \( C_{new\_idpw\_digest} \) and \( C_{auth\_token} = H(H(id, pw), rcs, rs) \), the scheme is the same as Tseng et al.’s protected password transmission scheme.

4. Security Analysis

In this section, we shall first list the security definitions, and then several possible attacks will be analyzed to demonstrate the security of our improved scheme.

**Definition 1:** A weak secret (password) is a value of low entropy \( w(k) \), which can be guessed in a polynomial time.

**Definition 2:** The Diffie-Hellman assumption is that given \( g^c \mod p \) and \( g^s \mod p \), to compute \( g^{cs} \mod p \) is hard.

**Definition 3:** A secure one-way hash function \( Y = H(X) \) is one where given \( X \) to compute \( Y \) is easy and given \( Y \) to compute \( X \) is hard.

**Replay attacks:**

The attacker intercepts \( \{id, C_{idpw\_digest}, C_{new\_idpw\_digest}\} \) sent by the client in Step 1 and uses it to masquerade as the client to send the login re-
quest to the server. However, he/she has no ability to make a correct response $C_{auth\_token}$ in Step 3 because the random number $rs$ generated by the server is different every time. On the other hand, since the message sent by the server and the client is different, the attacker cannot intercept any message between them and then replay it to the other party.

**Guessing attacks:**
Ding and Horster (Ding and Horster, 1995) divided password-guessing attacks into three classes: (1) Detectable on-line guessing attacks, (2) Undetectable on-line guessing attacks, and (3) Off-line guessing attacks. Because the detectable on-line guessing attacks can be prevented by letting the server take appropriate intervals between trials, we shall only separately show that the latter two guessing attacks will fail in our scheme as follows.

**Undetectable on-line guessing attacks:**
The attacker attempts to use a guessed password $pu'$ in an on-line transaction. She/he verifies the correctness of her/his guess by observing the response of the server. A failed guess cannot be detected by the server. The attacker disguises as the client and cheats the server. She/he can generate $C_{idpw\_digest} = H(id, pu') \oplus (ra = g^a \mod p)$ and $C_{new\_idpw\_digest} = ra \oplus H(id, new\_pu')$ by herself/himself and get $S_{idpw\_digest} = verify_{idpw\_digest} \oplus rs$, $S_{auth\_token} = H(verify_{idpw\_digest}, H(id, pu') \oplus ra \oplus H(id, pu))^b \mod p, H(id, pu') \oplus ra \oplus H(id, pu)$ sent from the server in Step 2. In order to verify $S_{auth\_token}$, the attacker must control $ras = g^{as} \mod p$ by finding $rs = rs \oplus H(id, pw) \oplus H(id, pu')$ and $ra = ra \oplus H(id, pw) \oplus H(id, pu')$ simultaneously. However, the probability is $2^{-w(k)}$. Even if $pu'$ is guessed, the attacker cannot verify $S_{auth\_token}$ and reply the correct $C_{auth\_token}$ to the server. On the other hand, the attacker still cannot mount the on-line guessing attacks for guessing new_pw because no message from the server includes new_pw. Therefore, no undetectable on-line guessing attacks can do any harm to our scheme.

**Off-line guessing attacks:**
Since easy-to-remember passwords do not have enough entropy, the attacker can guess the password have off-line, independent of the server. The attacker has the knowledge of $C_{idpw\_digest}, S_{idpw\_digest}, C_{auth\_token}$, and $S_{auth\_token}$. He/she first guesses a password $pu'$ and then finds $rc = rc \oplus H(id, pw) \oplus H(id, pu')$ and $rs = rs \oplus H(id, pw) \oplus H(id, pu')$. However, the adversary has to break the Diffie-Hellman assumption to find $rcs$ and then
verify his/her guess. Without knowing $rc$, $rs$ and $res$, the attacker cannot guess $new\_pw$ from $C_{new\_idpw\_digest}$ in Step 1 and $C_{auth\_token}$ in Step 3.

**Server spoofing:**
Only the real server has the ability to recover $rc$ from $C_{idpw\_digest}$ and then make a correct response $S_{auth\_token}$ in Step 2. An illegal client cannot recover $rs$ from $S_{idpw\_digest}$ and then make a correct response $C_{auth\_token}$ in Step 3. The improved scheme can also achieve the property of mutual authentication.

**Modification attack:**
The attacker tries to make the client convinced that it has successfully changed the password from $pw$ to $new\_pw$. In fact, the $verify\_idpw\_digest$ stored in the server’s database is a random number. The attacker modifies $C_{new\_idpw\_digest}$ to a random number $ra$ in Step 1. After receiving $\{id, C_{idpw\_digest}, ra\}$, the server recovers $rc$ and uses it to obtain $ra \oplus rc$ and then sends $\{S_{idpw\_digest}, S_{auth\_token}\}$ to the client in Step 2. After receiving $\{S_{idpw\_digest}, S_{auth\_token}\}$, the client first authenticates the server and then sends $\{id, C_{auth\_token}\}$ to the server in Step 3. After receiving $\{id, C_{auth\_token}\}$, the server first uses the recovered $ra \oplus rc$ to compute $C_{auth\_token} \oplus (ra \oplus rc)$ and then compares it with $H(verify\_idpw\_digest, res, rs)$. Obviously, $C_{auth\_token} \oplus (ra \oplus rc)$ is not equal to $H(verify\_idpw\_digest, res, rs)$ because $C_{auth\_token} \oplus (ra \oplus rc) = H(H(id, pw), res, rs) \oplus H(id, new\_pw) \oplus (ra \oplus rc)$. However, even if the attacker can compute $H(H(id, pw), res, rs) \oplus (ra \oplus rc)$ as $C_{auth\_token}$ and send it to the server in Step 3, she/he cannot break the Diffie-Hellman assumption to obtain $rc, rs$ and $res$, and get $H(id, pw)$.

After the above discussions, we have come to the conclusion that our improved scheme can withstand replay attacks, guessing attacks, and server spoofing, and it can work successfully against modification attacks. Moreover, a session key between the client and the server can be established as $H(res)$. When the password is compromised, the attacker still cannot reveal an old session key. The scheme can still live up to the requirement of perfect forward secrecy (Jablon, 1996).

5. Conclusion
In this paper, we have shown that Tseng et al.’s protected password changing scheme is vulnerable to the modification attack. In addition, we have proposed
our improved scheme to effectively mend the security flaw without employing any symmetric or asymmetric cryptosystems.

Acknowledgements The authors wish to thank many anonymous referees for their suggestions to improve this paper. Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC90-2213-E-324-004.

REFERENCES

Daemen, J. and V. Rijmen (2001), Rijndael, the advanced encryption standard. Dr. Dobb’s Journal, 26(3), 137–139.
Security of Improvement on Methods for Protecting Password Transmission


C.-C. Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. He has been an associate professor in the Dept. of Information and Communication Engineering since 1994. His current research interests include network security, mobile computing, and distributed system.

T.-Y. Chang received the B.S. in Information Management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, Republic of China, in 2001, and his M.S. in Department and Graduate Institute of Computer Science and Information Engineering from CYUT, in 2003. He is currently pursuing his Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, Republic of China. His current research interests include information security, cryptography, and mobile communications.

M.-S. Hwang was born on August 27, 1960 in Tainan, Taiwan, Republic of China (ROC.). He received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He obtained the 1997, 1998, 1999, 2000, and 2001 Distinguished Research
Awards of the National Science Council of the Republic of China. He is currently a professor of the department of Management Information Systems, National Chung Hsing University, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 80 articles on the above research fields in international journals.