INFORMATICA, 2005, Vol. 16, No. 2, 203–212

Weaknesses and Improvements of Yang–Chang–Hwang's Password Authentication Scheme

Wei-Chi KU, Hao-Chuan TSAI

Department of Computer Science and Information Engineering, Fu Jen Catholic University
510 Chung Cheng Road, Hsinchuang, Taipei County, Taiwan 242, R.O.C.
e-mail: wcku@csie.fju.edu.tw

Received: May 2004

Abstract. In 2001, Tseng, Jan, and Chien proposed an improved version of Peyravian–Zunic's password authentication scheme based on the Diffie–Hellman scheme. Later, Yang, Chang, and Hwang demonstrated that Tseng–Jan–Chien's scheme is vulnerable to a modification attack, and then described an improved scheme. In this paper, we show that Yang–Chang–Hwang's scheme is still vulnerable to a denial-of-service attack and a stolen-verifier attack. In addition, we also propose an improved scheme with better security.

Key words: password authentication, denial-of-service attack, stolen-verifier attack.

1. Introduction

Password authentication is a popular approach often used to authenticate users logining any kind of servers due to its simplicity and convenience. Password authentication involves the requesting entity, the user, providing a secret to the receiving entity, the server, which then checks the secret against a value stored earlier to confirm the authenticity of the user. To avoid incurring heavy computational overhead to the application system, many password authentication schemes mainly using one-way hash functions as their basic building blocks have been proposed. In 1981, Lamport (Lamport, 1981) initially proposed a password authentication scheme based on one-way hash functions. However, high hash overhead and the necessity for password resetting decrease its suitability for practical use. Additionally, Lamport's scheme is vulnerable to a preplay attack (Mitchell and Chen, 1996). In 1990, Shimizu (Shimizu, 1990) proposed a hash-based password authentication scheme, CINON, in which the one-time characteristic is gained by using two variable random numbers that are changed at each authentication. However, the user has to memorize two variable random numbers. This inconvenience obstructs the deployment of CINON. To eliminate the drawback of CINON, Shimizu, Horioka, and Inagaki (Shimizu *et al.*, 1998) proposed an enhanced scheme, PERM, in which the user doesn't need to memorize any random number. Instead, a random number is stored in the server

for authenticating the user. It is only when the server receives the correct response corresponding to the sent challenge, i.e., the stored random number, he will believe that the user is authentic and then refresh the stored random number. Unfortunately, PERM was found to be vulnerable to a man-in-the-middle attack (Sandirigama *et al.*, 2000) in that the adversary can impersonate the user by modifying two consecutive sessions between the user and the server.

In 2000, Sandirigama, Shimizu, and Noda (Sandirigama *et al.*, 2000) proposed a simple hash-based password authentication scheme, SAS, which was claimed to be superior to previous similar schemes in utilization, processing time, and transmission overhead. However, SAS was later found to be vulnerable to a replay attack, a denial-of-service attack (Lin *et al.*, 2001), and a stolen-verifier attack (Chen and Ku, 2002). To overcome the weaknesses of SAS, Lin, Sun, and Hwang (Lin *et al.*, 2001) proposed an improved version, OSPA. However, OSPA was found to be vulnerable to a stolen-verifier attack (Chen and Ku, 2002) and a man-in-the-middle attack (Tsuji and Shimizu, 2003). Next, Lin, Shen, and Hwang (Lin *et al.*, 2003) proposed an improved version of OSPA by additionally using smart cards. Unfortunately, Ku, Tsai, and Chen (Ku *et al.*, 2003c) found that Lin–Shen–Hwang's scheme is still vulnerable to a denial-of-service attack and a replay attack.

Independent of the above mentioned schemes, Peyravian and Zunic (Peyravian and Zunic, 2000) also proposed a hash-based password authentication scheme, which involves the protected password transmission protocol and the protected password change protocol. Later, Hwang and Yeh (Hwang and Yeh, 2002) showed that Peyravian–Zunic's scheme is vulnerable to an off-line password guessing attack, a server spoofing attack, and a server data eavesdropping attack, and then proposed a modified scheme. However, Hwang–Yeh's scheme is inefficient and inconvenient because of using the public-key cryptosystem. Furthermore, Hwang–Yeh's scheme was found to be vulnerable to a denial-of-service attack and the threat of a replay attack (Ku *et al.*, 2003a). Next, Lee, Li, and Hwang (Lee *et al.*, 2002) proposed another improvement of Peyravian–Zunic's scheme. However, Ku, Chen, and Lee (Ku *et al.*, 2003b) demonstrated that Lee–Li–Hwang's scheme is still vulnerable to a denial-of-service attack, a stolen-verifier attack, and an off-line password guessing attack. In 2001, Tseng, Jan, and Chien (Tseng *et al.*, 2001) also pointed out the weaknesses of Peyravian–Zunic's scheme, and then proposed an improved version based on the Diffie–Hellman scheme (Diffie and Hellman, 1976). However, Tseng–Jan–Chien's scheme was found to be vulnerable to a modification attack (Yang *et al.*, 2003), which is a kind of denial-of-service attacks, and a stolen-verifier attack (Hsieh *et al.*, 2003). Recently, Yang, Chang, and Hwang (Yang *et al.*, 2003) proposed an improved version of Tseng–Jan–Chien's scheme and claimed that their scheme is secure against the password guessing attack, the replay attack, the server spoofing attack, and the modification attack. Unfortunately, we find that Yang–Chang–Hwang's scheme is still vulnerable to a denial-of-service attack and a stolen-verifier attack. In this paper, we will first review Yang–Chang–Hwang's scheme, and then describe its weaknesses. The rest of this paper is organized as follows. In Section 2, we briefly review Yang–Chang–Hwang's scheme. Then, we show the weaknesses of Yang–Chang–Hwang's scheme in

Section 3. Then, we will propose an improved scheme with better resistances to the denial-of-service attack and the stolen-verifier attack in Section 4. Finally, conclusions are made in Section 5.

2. Review of Yang–Chang–Hwang's Scheme

Yang–Chang–Hwang's scheme (Yang *et al.*, 2003) involves the protected password transmission protocol and the protected password change protocol. Its protected password transmission protocol is identical to the one of Tseng–Jan–Chien's scheme while its protected password change protocol was intended to be an improvement of the one of Tseng–Jan–Chien's scheme. For readers' convenience, we will review Yang–Chang–Hwang's scheme before demonstrating its weaknesses. The notations used throughout this paper are described as follows. Notations id and pw represent the identity and the password of the client, respectively. $H(\cdot)$ denotes a one-way hash function and $\oplus$ represents the bitwise XOR operation. Initially, the server publishes two large prime numbers p and q such that $q \,|\, p-1$ and a generator g with order q in the Galois field $GF(p)$. To register as a user, the client computes $hpw = H(id, pw)$, and delivers hpw along with id to the server through a secure channel. Then, the server creates an entry in the database for the client to store hpw as the verifier of pw. The protected password transmission protocol and the protected password change protocol of Yang–Chang–Hwang's scheme can be described as in the following.

2.1. *Protected Password Transmission Protocol*

The protected password transmission protocol is invoked whenever the client requests to login the server by using pw.

Step T1. Client $\longrightarrow$ Server: id, $hpw \oplus rc$.

The client enters id and pw to compute $hpw = H(id, pw)$, and chooses a random number $c \in [1, q-1]$ to compute $rc = g^c \bmod p$ and $hpw \oplus rc$. Next, the client sends id and $hpw \oplus rc$ to the server.

Step T2. Server $\longrightarrow$ Client: $hpw \oplus rs$, $H(hpw, rcs, rc)$.

The server retrieves rc from the second item of the message received in *Step* T1 by using the stored verifier hpw. In addition, the server chooses a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$ and $hpw \oplus rs$. Next, the server computes $rcs = rc^s = g^{cs} \bmod p$, and then uses hpw, the computed rcs, and the retrieved rc to compute $H(hpw, rcs, rc)$. Then, the server sends $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client.

Step T3. Client $\longrightarrow$ Server: id, $H(hpw, rcs, rs)$.

The client retrieves rs from the first item of the message received in *Step* T2 by using hpw, and then computes $rcs = rs^c = g^{sc} \bmod p$. Next, the client uses hpw, rc, and rcs to compute $H(hpw, rcs, rc)$. If the computed result equals the second item of the

message received in *Step* T2, the server is authenticated. Next, the client uses hpw, rcs, and rs to compute $H(hpw, rcs, rs)$, and then sends id and $H(hpw, rcs, rs)$ to the server.

Step T4. Server $\longrightarrow$ Client: Access Granted/Denied.

The server uses rcs, rs, and the stored verifier hpw to compute $H(hpw, rcs, rs)$. If the computed result equals the second item of the message received in *Step* T3, the server grants the client's login request and sends 'Access Granted' to the client. Otherwise, the server rejects the client's login request and sends 'Access Denied' to the client. After successful mutual authentication, the client and the server can compute $H(rcs)$ and use it as the session key for protecting the messages exchanged between them in this session.

2.2. *Protected Password Change Protocol*

The protected password change protocol is invoked whenever the client requests to change his password pw with a new one, say pw_{new}.

Step C1. Client $\longrightarrow$ Server: id, $hpw \oplus rc$, $hpw_{new} \oplus rc$.

The client enters id and pw to compute $hpw = H(id, pw)$ and chooses a random number $c \in [1, q-1]$ to compute $rc = g^c \bmod p$ and $hpw \oplus rc$. In addition, the client chooses his new password pw_{new}, and then computes $hpw_{new} = H(id, pw_{new})$ and $hpw_{new} \oplus rc$. Next, the client sends id, $hpw \oplus rc$, and $hpw_{new} \oplus rc$ to the server.

Step C2. Server $\longrightarrow$ Client: $hpw \oplus rs$, $H(hpw, rcs, rc)$.

The server retrieves rc from the second item of the message received in *Step* C1 by using the stored verifier hpw, and then uses the retrieved rc to retrieve hpw_{new} from the third item of the message received in *Step* C1. Next, the server chooses a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$ and $hpw \oplus rs$. In addition, the server computes $rcs = rc^s = g^{cs} \bmod p$, and then uses the stored verifier hpw, the computed rcs, and the retrieved rc to compute $H(hpw, rcs, rc)$. Next, the server sends $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client.

Step C3. Client $\longrightarrow$ Server: id, $H(hpw, rcs, rs) \oplus hpw_{new}$.

The client retrieves rs from the first item of the message received in *Step* C2 by using hpw, and then computes $rcs = rs^c = g^{sc} \bmod p$. Next, the client uses hpw, rcs, and rc to compute $H(hpw, rcs, rc)$. If the computed result equals the second item of the message received in *Step* C2, the server is authenticated. After successfully authenticating the server, the client uses rcs, hpw, hpw_{new}, and the retrieved rs to compute $H(hpw, rcs, rs) \oplus hpw_{new}$, and then sends id and $H(hpw, rcs, rs) \oplus hpw_{new}$ to the server.

Step C4. Server $\longrightarrow$ Client: Accepted/Denied.

The server uses the previously obtained hpw_{new} to retrieve $H(hpw, rcs, rs)$ from the second item of the message received in *Step* C3, and uses the stored verifier hpw, the previously computed rcs, and rs to compute $H(hpw, rcs, rs)$. If the computed

$H(hpw, rcs, rs)$ equals the retrieved one, the server authenticates the client, and then updates the stored verifier hpw with hpw_{new} and sends 'Accepted' to the client. Otherwise, the server sends 'Denied' to the client.

3. Weaknesses of Yang–Chang–Hwang's Scheme

In this section, we will show that Yang–Chang–Hwang's scheme (Yang *et al.*, 2003) is vulnerable to a denial-of-service attack and a stolen-verifier attack.

3.1. *Denial-of-Service Attack*

The denial-of-service attack is an attack leading a legal user can not login the server or the server can not provide service normally. Next, we will demonstrate the way to mount a denial-of-service attack on Yang–Chang–Hwang's scheme. Suppose that the client requests to change his password pw with a new one, say pw_{new}, by invoking the protected password change protocol. During *Step* C1, an adversary, say Eve, can replace the transmitting $hpw_{new} \oplus rc$ with $(hpw_{new} \oplus rc) \oplus re$, where re is a random number selected by Eve. Upon receiving the modified message in *Step* C1, the server will retrieve rc by using the stored verifier hpw, and then use the retrieved rc to retrieve $hpw_{new} \oplus re$ from the received $(hpw_{new} \oplus rc) \oplus re$ (= $(hpw_{new} \oplus re) \oplus rc$). Next, the server will choose a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$, $hpw \oplus rs$, $rcs = rc^s = g^{cs} \bmod p$, and $H(hpw, rcs, rc)$. Then, the server sends $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client in *Step* C2. Upon receiving the message received in *Step* C2, the client will successfully authenticate the server. Then, the client will send id and $H(hpw, rcs, rs) \oplus hpw_{new}$ to the server in *Step* C3. During *Step* C3, Eve can replace the transmitting $H(hpw, rcs, rs) \oplus hpw_{new}$ with $(H(hpw, rcs, rs) \oplus hpw_{new}) \oplus re$. Upon receiving the modified message in *Step* C3, the server will use the previously obtained $hpw_{new} \oplus re$ to retrieve $H(hpw, rcs, rs)$ from the received $(H(hpw, rcs, rs) \oplus hpw_{new}) \oplus re$ (= $(hpw_{new} \oplus re) \oplus H(hpw, rcs, rs)$). In addition, the server will use hpw, rcs, and rs to compute $H(hpw, rcs, rs)$. Since the computed $H(hpw, rcs, rs)$ equals the retrieved one, the server authenticates the client and then changes the stored verifier hpw with $hpw_{new} \oplus re$, which clearly does not equal the expected hpw_{new}. That is, the server is fooled into taking $hpw_{new} \oplus re$ as the client's new verifier. Henceforth, the client's login and password change requests using hpw_{new} will be denied. Since the adversary can easily block the account of any client without using cryptographic techniques, Yang–Chang–Hwang's scheme is vulnerable to a denial-of-service attack.

3.2. *Stolen-Verifier Attack*

In most existing password authentication schemes, the server stores the verifiers of users' passwords rather than users' bare passwords to reduce the risk once the server is compromised. The stolen-verifier attack is an offensive attack that the adversary who has

stolen the user's verifier can impersonate that user and/or perform other attacks. Next, we will describe the way to mount a stolen-verifier attack on Yang–Chang–Hwang's scheme. Clearly, once an adversary, say Eve, has stolen an ever used verifier, she can compute its succeeding verifier if she has also intercepted the corresponding message transmitted in *Step* C1. By iteratively applying the above method, Eve can obtain the client's current verifier hpw. Suppose that Eve has obtained, either directly or indirectly, the client's current verifier hpw. First, Eve can choose a random number e to compute $re = g^e \bmod p$ and $hpw \oplus re$, and then impersonate the client to send id and $hpw \oplus re$ to the server in *Step* T1 of the protected password transmission protocol. Upon receiving the forged message, the server will use the stored verifier hpw to retrieve re from the second received item, and then choose a random number s to compute $rs = g^s \bmod p$, $hpw \oplus rs$, $res = re^s = g^{es} \bmod p$, and $H(hpw, res, re)$. Next, the sever will send $hpw \oplus rs$ and $H(hpw, res, re)$ to Eve. Then, Eve can use the stolen verifier hpw to retrieve rs from the first received item, and then compute $res = rs^e = g^{se} \bmod p$ and $H(hpw, res, rs)$. Next, Eve sends id and $H(hpw, res, rs)$ to the server in *Step* T3. Then, the server will use the stored verifier hpw and the previously computed res and rs to compute $H(hpw, res, rs)$. Since the computed $H(hpw, res, rs)$ equals the second item of the received message, the server will grant Eve's login request, i.e., Eve can successfully impersonate the client to login the server and obtain the resources or services she needs. Similarly, Eve can also use the stolen verifier hpw to change the client's password pw with her own password pw_e by invoking the protected password change protocol. As the client's verifier will be replaced with hpw_e ($= H(id, pw_e)$), Eve can use pw_e to impersonate the client to login the server. In this case, the client's succeeding login requests using pw will be denied.

Alternatively, knowing hpw, Eve can also perform a man-in-the-middle attack as follows. During the client's login process, Eve can use the stolen hpw to impersonate the server to fool the client into establishing a session key with her and simultaneously impersonate the client to fool the server into establishing another session key with her. Therefore, Eve can decrypt all the encrypted messages exchanged between the client and the server. Clearly, such an attack can not be easily detected.

4. An Improved Version of Yang–Chang–Hwang's Scheme

In this section, we will describe an improved version of Yang–Chang–Hwang's scheme with better security strength.

4.1. *The Improved Scheme*

As in Yang–Chang–Hwang's scheme, the server initially publishes two large prime numbers p and q such that $q \mid p - 1$ and a generator g with order q in $GF(p)$. To register as a user, the client computes $hpw = H(id, pw)$ and delivers hpw along with id to the server through a secure channel. Then, the server computes $enc\_hpw = hpw \oplus H(x, id)$, where x is a secret storage key of the server, and creates an entry in the database to store

enc_hpw for the client. It is assumed that x is under strict protection. The improved protected password transmission protocol and protected password change protocol can be described as in the following.

4.1.1. *Improved Protected Password Transmission Protocol*
The improved protected password transmission protocol is identical to the one of Yang–Chang–Hwang's scheme except that *Step* T2 is slightly changed into *Step* T2′ as follows:

Step T2′. Server $\longrightarrow$ Client: $hpw \oplus rs$, $H(hpw, rcs, rc)$.

The server computes $H(x, id)$ and then uses the result to retrieve hpw from the stored enc_hpw ($= hpw \oplus H(x, id)$). Next, the server uses the retrieved hpw to retrieve rc from the second item of the message received in *Step* T1. In addition, the server chooses a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$ and $hpw \oplus rs$. Next, the server computes $rcs = rc^s = g^{cs} \bmod p$, and then uses hpw, rcs, and rc to compute $H(hpw, rcs, rc)$. Then, the server sends $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client. As in Yang–Chang–Hwang's scheme, the client and the server can compute $H(rcs)$ after successful mutual authentication and then use it as the session key for protecting the messages exchanged between them in this session.

4.1.2. *Improved Protected Password Change Protocol*
Steps C1, C2, C3, and C4 of Yang–Chang–Hwang's scheme are changed into C1′, C2′, C3′, and C4′, respectively, as follows:

Step C1′. Client $\longrightarrow$ Server: id, $hpw \oplus rc$.

The client enters id and pw to compute $hpw = H(id, pw)$ and chooses a random number $c \in [1, q-1]$ to compute $rc = g^c \bmod p$ and $hpw \oplus rc$. Next, the client sends id and $hpw \oplus rc$ to the server.

Step C2′. Server $\longrightarrow$ Client: $hpw \oplus rs$, $H(hpw, rcs, rc)$.

The server computes $H(x, id)$ and then uses the result to retrieve hpw from the stored enc_hpw ($= hpw \oplus H(x, id)$). Next, the server uses the retrieved hpw to retrieve rc from the second item of the message received in *Step* C1′. In addition, the server chooses a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$ and $hpw \oplus rs$. Next, the server computes $rcs = rc^s = g^{cs} \bmod p$, and then uses hpw, rcs, and rc to compute $H(hpw, rcs, rc)$. Then, the server sends $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client.

Step C3′. Client $\longrightarrow$ Server: id, $H(hpw, rcs, rs) \oplus hpw_{new}$, $H(rcs, hpw_{new})$.

The client retrieves rs from the first item of the message received in *Step* C2′ by using hpw, and then computes $rcs = rs^c = g^{sc} \bmod p$. Next, the client uses hpw, rcs, and rc to compute $H(hpw, rcs, rc)$. If the computed result equals the second item of the message received in *Step* C2′, the server is authenticated. Then, the client chooses his new password pw_{new}, and then computes $hpw_{new} = H(id, pw_{new})$. Next, the client uses rcs, hpw, hpw_{new}, and the retrieved rs to compute $H(hpw, rcs, rs) \oplus hpw_{new}$ and

$H(rcs, hpw_{new})$, and then sends id, $H(hpw, rcs, rs) \oplus hpw_{new}$, and $H(rcs, hpw_{new})$ to the server.

Step C4′. Server ⟶ Client: Accepted/Denied.

The server uses hpw, rcs, and rs to compute $H(hpw, rcs, rs)$ and then uses the computed result to retrieve hpw_{new} from the second item of the message received in *Step* C3′. Next, the server uses rcs and the retrieved hpw_{new} to compute $H(rcs, hpw_{new})$. If the computed result equals the third item of the message received in *Step* C3′, the server authenticates the client, and then updates the stored enc_hpw with $enc\_hpw_{new} = hpw_{new} \oplus H(x, id)$ and sends 'Accepted' to the client. Otherwise, the server rejects the client's login request and sends 'Denied' to the client.

4.2. *Security Analysis of The Improved Scheme*

The resistances of the improved scheme to the replay attack, the password guessing attack, and the server spoofing attack are similar to the ones of Yang–Chang–Hwang's scheme (Yang *et al.*, 2003), and we omit the proofs for clearness. Next, we will show that the improved scheme can additionally resist the denial-of-service attack and the stolen-verifier attack. Let Eve denote the adversary.

4.2.1. *Resistance to Denial-of-Service Attack*

Suppose that the client requests to change his password pw with pw_{new} by invoking the improved protected password change protocol. Upon receiving the client's request sent in *Step* C1′, the server will compute $H(x, id)$ to retrieve hpw from the stored enc_hpw. Next, the server will use hpw to retrieves rc from the received $hpw \oplus rc$. In addition, the server will choose a random number $s \in [1, q-1]$ to compute $rs = g^s \bmod p$, $hpw \oplus rs$, $rcs = rc^s = g^{cs} \bmod p$, and $H(hpw, rcs, rc)$, and then send $hpw \oplus rs$ and $H(hpw, rcs, rc)$ to the client in *Step* C2′. Upon receiving the message sent in *Step* C2′, the client will successfully authenticate the server and send id, $H(hpw, rcs, rs) \oplus hpw_{new}$, and $H(rcs, hpw_{new})$ to the server in *Step* C3′. If Eve can replace the transmitting $H(hpw, rcs, rs) \oplus hpw_{new}$ and $H(rcs, hpw_{new})$ with $(H(hpw, rcs, rs) \oplus hpw_{new}) \oplus re$ and $H(rcs, hpw_{new} \oplus re)$, where re is a random number selected by Eve, respectively, she can fool the server into updating the stored enc_hpw with $hpw_{new} \oplus re \oplus H(x, id)$ so that the client's subsequent login requests and password change requests will be denied. However, Eve can not produce $H(rcs, hpw_{new} \oplus re)$ because she can not obtain rcs and hpw_{new} by analyzing the protocol messages. Thus, the improved scheme can resist the denial-of-service attack.

4.2.2. *Resistance to Stolen-Verifier Attack*

Suppose that Eve has stolen enc_hpw. As $enc\_hpw = hpw \oplus H(x, id)$, Eve can derive hpw only if she knows $H(x, id)$, which implies that she knows x, the secret storage key of the server. However, since x is under strict protection as assumed, it is infeasible for Eve to derive hpw in this way. Therefore, the improved scheme can prevent from the stolen-verifier attack.

5. Conclusion

Herein, we have shown that Yang–Chang–Hwang's password authentication scheme, which was intended to be an improved version of Tseng–Jan–Chien's password authentication scheme, is still vulnerable to a denial-of-service attack and a stolen-verifier attack. As analyzed, the security flaws of Yang–Chang–Hwang's scheme are due to two problems. First, the integrity of the transmitted messages from the user to the server in the protected password change protocol are not well protected. The adversary can modify the transmitted messages to block the account of the user without being detected by the server. Secondly, although the server stores the verifier of the user's password rather than the user's bare password to reduce the risk once the server is compromised, the adversary can still use the stolen verifier to impersonate the user. Furthermore, we have described an improved version of Yang–Chang–Hwang's scheme with better resistances to the denial-of-service attack and the stolen-verifier attack.

Acknowledgement

We are grateful to the referees for their valuable comments. This research was supported by the National Science Council, Republic of China, under Grant NSC-92-2213-E-030-013.

References

Chen, C.M., and W.C. Ku (2002). Stolen-verifier attack on two new strong-password authentication protocols. *IEICE Transactions on Communications*, **E85-B**(11), 2519–2521.

Diffie, W., and M. Hellman (1976). New direction in cryptography. *IEEE Transactions on Information Theory*, **22**(6), 472–492.

Hsieh, B.T., H.M. Sun and T. Hwang (2003). On the security of some password authentication protocols. *Informatica*, **14**(2), 195–204.

Hwang, J.J., and T.C. Yeh (2002). Improvement on Peyravian–Zunic's password authentication schemes. *IEICE Transactions on Communications*, **E85-B**(4), 823–825.

Ku, W.C., C.M. Chen and H.L. Lee (2003a). Cryptanalysis of a variant of Peyravian–Zunic's password authentication scheme. *IEICE Transactions on Communications*, **E86-B**(5), 1682–1684.

Ku, W.C., C.M. Chen and H.L. Lee (2003b). Weaknesses of Lee–Li–Hwang's hash-based password authentication scheme. *ACM Operating Systems Review*, **37**(4), 19–25.

Ku, W.C., H.C. Tsai and S.M. Chen (2003c). Two simple attacks on Lin–Shen–Hwang's strong-password authentication protocol. *ACM Operating Systems Review*, **37**(4), 26–31.

Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, **24**(11), 770–772.

Lee, C.C., L.H. Li and M.S. Hwang (2002). A remote user authentication scheme using hash functions. *ACM Operating Systems Review*, **36**(4), 23–29.

Lin, C.L., H.M. Sun and T. Hwang (2001). Attacks and solutions on strong-password authentication. *IEICE Transactions on Communications*, **E84-B**(9), 2622–2627.

Lin, C.W., J.J. Shen and M.S. Hwang (2003). Security enhancement for optimal strong password authentication protocol. *ACM Operating Systems Review*, **37**(2), 7–12.

Mitchell, C.J., and L. Chen (1996). Comments on the S/KEY user authentication scheme. *ACM Operating Systems Review*, **30**(4), 12–16.

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

Sandirigama, M., A. Shimizu and M.T. Noda (2000). Simple and secure password authentication protocol (SAS). *IEICE Transactions on Communications*, **E83-B**(6), 1363–1365.

Shimizu, A. (1990). A dynamic password authentication method by one-way function. *IEICE Transactions*, **J73-D-I**(7), 630–636.

Shimizu, A., T. Horioka and H. Inagaki (1998). A password authentication methods for contents communication on the Internet. *IEICE Transactions on Communications*, **E81-B**(8), 1666–1673.

Tseng, Y.M., J.K. Jan and H.Y. Chien (2001). On the security of methods for protecting password transmission. *Informatica*, **12**(3), 469–477.

Tsuji, T., and A. Shimizu (2003). An impersonation attack on one-time password authentication protocol OSPA. *IEICE Transactions on Communications*, **E86-B**(7), 2182–2185.

Yang, C.C., T.Y. Chang and M.S. Hwang (2003). Security of improvement on methods for protecting password transmission. *Informatica*, **14**(4), 551–558.

W.C. Ku was born in Taiwan in 1967. In 2000, he received the PhD degree in Electrical Engineering from National Taiwan University. In 2001, Dr. Ku joined the faculty of the Department of Computer Science and Information Engineering at Fu Jen Catholic University, where he is currently an associate professor. His research interests include cryptography and information security.

H.C. Tsai was born in Taiwan in 1978. In 2002, he received the BS degree in Mathematics from SooChow University, Taiwan. He is currently a master's student in computer science and information engineering at Fu Jen Catholic University. His research interests include cryptography and information security.

Yang–Chang–Hwang slaptažodžio autentiškumo schemos trūkumai ir pagerinimai

Wei-Chi KU, Hao-Chuan TSAI

Tseng, Jan ir Chien 2001 buvo pasiūlyta pagerinta Peyravian–Zunic slaptažodžio autentiškumo schema, pagrįsta Diffie–Hellman schema. Vėliau, Yang, Chang ir Hwang pademonstravo, kad Tseng–Jan–Chien schema yra pažeidžiama modifikavimo ataka, ir pasiūlė pagerintą schemą. Šiame straipsnyje parodoma, kad Yang–Chang–Hwang schema yra pažeidžiama serviso-paneigimo ir pavogto-tikrintojo atakų. Dėl to geresniam saugumui pasiūlyta pagerinta schema.