INFORMATICA, 2005, Vol. 16, No. 2, 285–294

Attacks and Solutions of Yang *et al.*'s Protected Password Changing Scheme

Eun-Jun YOON, Eun-Kyung RYU, Kee-Young YOO

Department of Computer Engineering, Kyungpook National University
1370 Sankyuk-dong, Buk-gu, Daegu 702-701, South Korea
e-mail: {ejyoon,ekryu}@infosec.knu.ac.kr, yook@knu.ac.kr

Received: January 2004

Abstract. Recently, Yang *et al.* proposed an improvement to Tseng *et al.*'s protected password changing scheme that can withstand denial of service attack. However, the improved scheme is still susceptible to stolen-verifier attack and denial of service attack. Accordingly, the current paper demonstrates the vulnerability of Yang *et al.*'s scheme to two simple attacks and presents an improved protected password change scheme to resolve such problems. In contrast to Yang *et al.*'s protected password changing scheme and the existing password change schemes using server's public key, the proposed scheme can securely update user passwords without a complicated process and server's public key.

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

1. Introduction

User authentication is an important part of security, along with confidentiality and integrity, for systems that allow remote access over untrustworthy networks, like the Internet. As such, a remote password authentication scheme authenticates the legitimacy of users over an insecure channel, where the password is often regarded as a secret shared between the remote system and the user. Based on knowledge of the password, the user can use it to create and send a valid login message to a remote system to gain the right to access. Meanwhile, the remote system also uses the shared password to check the validity of the login message and authenticate the user.

Password authentication protocols are very subject to replay, password guessing and stolen-verifier attacks (Lin and Hwang, 2003).

(1) Replay attack: A replay attack is an offensive action in which an adversary impersonates or deceives another legitimate participant through the reuse of information obtained in a protocol.
(2) Guessing attack: A guessing attack involves an adversary simply (randomly or systematically) trying passwords, one at a time, in hope that the correct password is found. Ensuring passwords chosen from a sufficiently large space can resist exhaustive password searches. However, most users select passwords from a small

subset of the full password space. Such weak passwords with low entropy are easily guessed by using the so-called dictionary attack.

(3) Stolen-verifier attack: In most applications, the server stores verifiers of users' passwords (e.g., hashed passwords) instead of the clear text of passwords. The stolen-verifier attack means that an adversary who steals the password-verifier from the server can use it *directly* to masquerade as a legitimate user in a user authentication execution. Note that the main purpose of an authentication scheme against the stolen-verifier attack is to reduce the immediate danger to user authentication. In fact, an adversary who has a password-verifier may further mount a guessing attack on it.

Password change protocols allow an authenticated user to change his/her password. Besides those attacks mentioned above, a password change protocol is very vulnerable to denial of service attacks (Lin and Hwang, 2003).

(1) Denial of service attack: The denial of service attack prevents or inhibits the normal use or management of communications facilities. This attack may act on a specific user. For example, an adversary may perform this attack to cause the server to reject the login of a specific user.

In addition, the following security properties of session key agreement protocols should be considered since they are often desirable in some environments (Menezes *et al.*, 1997).

(1) Implicit key authentication: Implicit key authentication is the property obtained when identifying a party based on a shared session key, which assures that no other entity than the specifically identified entity can gain access to the session key.
(2) Explicit key authentication: Explicit key authentication is the property obtained when both implicit key authentication and key confirmation hold.
(3) Mutual authentication: Mutual authentication means that both the client and server are authenticated to each other within the same protocol, while explicit key authentication is the property obtained when both implicit key authentication and key confirmation hold.
(4) Forward secrecy: Forward secrecy means that if a long-term private key (e.g., user password or server private key) is compromised, this does not compromise any earlier session keys. In password authentication with key distribution, forward secrecy is a highly desirable security feature.

In 2000, Peyravian and Zunic (Peyravian and Zunic, 2000) proposed a protected password authentication scheme based on a one-way hash function to achieve user authentication and arbitrarily change a password. Subsequently, Tseng *et al.* (2001) pointed out that Peyravian-Zunic's scheme was vulnerable to guessing and server spoofing attacks and proposed a new protected password authentication scheme using the Diffie-Hellman key agreement scheme to eliminate the security flaws. Thereafter, in 2003, Yang *et al.* (2003) pointed out that Tseng *et al.*'s scheme was vulnerable to denial of service attack and proposed an improved scheme that could withstand denial of service attack. Yet, Yang *et al.*'s improved scheme is still susceptible to stolen-verifier attack, where obtaining the secret

data stored in a server can allow an illegitimate user to login to the server as a legitimate user and also their password change protocol suffers from a denial of service attack, in which an attacker can easily make the server reject all subsequent login requests of any user.

Accordingly, the current paper demonstrates that Yang *et al.*'s scheme is vulnerable to stolen-verifier attack and denial of service attack and also presents an improved protected password change scheme to the scheme to isolate such problems. In contrast to Yang's password change scheme and the existing password change schemes using server's public key (Hwang and Yeh, 2002; Ku *et al.*, 2003; Lin and Hwang, 2003), the proposed scheme can securely update user passwords without a complicated process and server's public key.

The remainder of this paper is organized as follows: Section 2 briefly reviews Yang *et al.*'s scheme and demonstrates stolen-verifier attack and denial of service attack with their scheme. The proposed scheme is presented in Section 3, while Section 4 discusses the security of the proposed scheme. Some final conclusions are given in Section 5.

2. Weakness on Yang *et al.*'s Scheme

This section briefly reviews Yang *et al.*'s scheme and then shows how the stolen-verifier attack and denial of service attack can work on their scheme. Some of the notations used in this paper are defined as follows:

- id: public user identity of client;
- pw: secret and possibly weak user password;
- K: strong secret key of server;
- p, q: large prime numbers p and q such that $q|p-1$;
- g: generator with order q in the Galois field $GF(p)$, in which Diffie-Hellman problem is considered hard;
- c, s: session-independent random exponents $\in [1, q-1]$ chosen by client and server, respectively;
- sk: shared session key computed by client and server;
- $H(\cdot)$: strong one-way hash function;
- $\oplus$: bit-wise XOR operation.

2.1. *Review of Yang et al.'s Protected Password Change Scheme*

In Yang *et al.*'s protected password changing scheme, the server stores $vpw = H(id, pw)$ for each client in the database and allows a client to change their old password pw to a new password $newpw$. Yang *et al.*'s protected password change scheme is illustrated in Fig. 1 and their scheme works as follows:

Step (1) Client $\rightarrow$ Server: $id,\ H(id, pw) \oplus g^c, H(id, newpw) \oplus g^c$

The user submits his id and pw to the client. The client then chooses a random number c and computes g^c and chooses a new password $newpw$ and uses g^c to

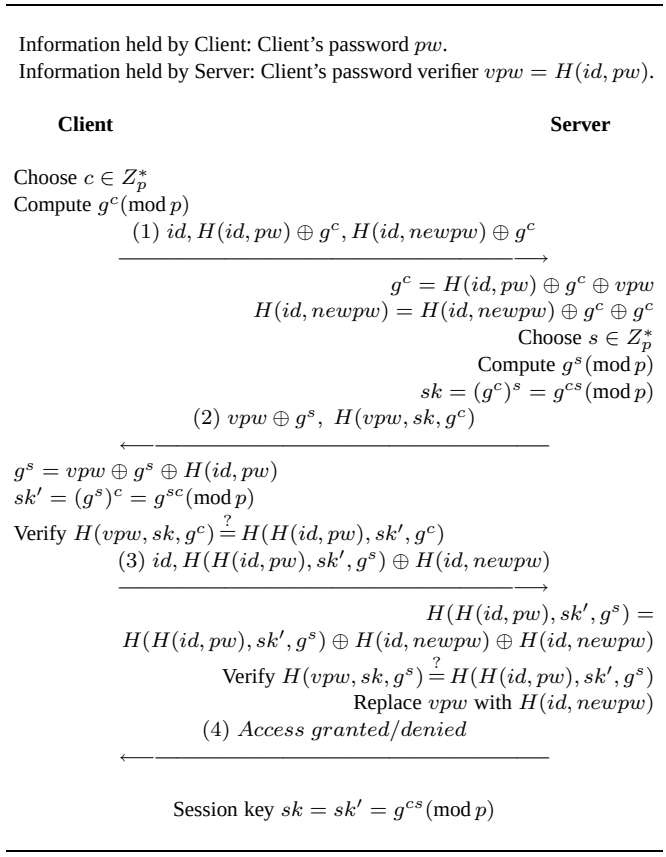

Fig. 1. Yang *et al.*'s protected password changing scheme.

compute $H(id,pw) \oplus g^c$ and $H(id,newpw) \oplus g^c$. The client sends its with the id as a login request to the server.

Step (2) Server → Client: $vpw \oplus g^s,\ H(vpw, sk, g^c)$

The server retrieves g^c from $H(id,pw) \oplus g^c$ by computing $H(id,pw) \oplus g^c \oplus vpw$. Then, the server uses the recovered g^c to obtain $H(id,newpw)$ from $H(id,newpw) \oplus g^c$ by computing $H(id,newpw) \oplus g^c \oplus g^c$. Then, the server chooses a random number s and computes g^s and $sk = (g^c)^s = g^{cs}$. Then, the server uses its own g^s, sk and the recovered g^c to compute $vpw \oplus g^s$ and $H(vpw, sk, g^c)$. The server sends its to the client.

Step (3) Client → Server: $id,\ H(H(id,pw), sk', g^s) \oplus H(id,newpw)$

The client retrieves g^s from $vpw \oplus g^s$ by computing $vpw \oplus g^s \oplus H(id,pw)$ and computes $sk' = (g^s)^c = g^{sc}$ and $H(H(id,pw), sk', g^c)$, then verifies the consistency between the retrieved $H(H(id,pw), sk', g^c)$ and the received $H(vpw, sk, g^c)$. If the result is positive, the client computes $H(H(id,pw), sk', g^s) \oplus H(id,newpw)$ and sends this client authentication token with the id to the server.

Step (4) Server → Client: $Access\ granted\ /\ denied$

The server computes the hash value $H(vpw, sk, g^s)$ using its own copies of sk and g^s. Then, the server retrieves $H(H(id, pw), sk', g^s)$ from $H(H(id, pw), sk', g^s) \oplus H(id, newpw)$ using the recovered $H(id, newpw)$ in the Step (1) and checks whether $H(vpw, sk, g^s) = H(H(id, pw), sk', g^s)$ holds or not. If it holds, the server replaces vpw with $H(id, newpw)$.

Once the server grants the client's login request, the final session key can be computed by the client as $(g^c)^s$ and by the server as $(g^s)^c$.

2.2. *Stolen-Verifier Attack on Yang et al.'s Scheme*

Servers are always the targets of attacker, because numerous customers' secrets are stored in their databases. The hash value $vpw = H(id, pw)$ of the user password stored in the server can be eavesdropped and then used to masquerade as the original user. Yang *et al.* (2003) did not explain about stolen-verifier attack, where obtaining the secret data $vpw = H(id, pw)$ stored in a server can allow an illegitimate user to login to the server as a legitimate user. If attacker stolen on the password verifier $vpw = H(id, pw)$ in the server, he can chooses a random number c' and computes $g^{c'}$ and chooses a new password $newpw'$ and uses $g^{c'}$ to computes client password digest $vpw \oplus g^{c'}$ and client new password digest $H(id, newpw') \oplus g^{c'}$ in Step (1). Then the attacker sends its with the id as a login request to the server and can masquerade the original user.

2.3. *Denial of Service Attack on Yang et al.'s Scheme*

Usually, the server closes a login session if the number of error login attempts of an account exceeds a limited value (e.g., 3 times). Even so, such a user's account is still workable and later login requests will pass as long as the correct password is provided. However, the Yang *et al.*'s password change protocol suffers from a denial of service attack, in which an attacker can easily make the server reject all subsequent login requests of any user. In Step (1) of Yang *et al.*'s password change protocol, an attacker can simply replace new password digest $H(id, newpw) \oplus g^c$ with current client password digest $H(id, pw) \oplus g^c$. After receiving the replaced messages $\{id, H(id, pw) \oplus g^c, H(id, pw) \oplus g^c\}$, the server retrieves g^c from $H(id, pw) \oplus g^c$ by computing $H(id, pw) \oplus g^c \oplus vpw$. Then, the server uses the recovered g^c to obtain current password verifier $H(id, pw)$ from $H(id, pw) \oplus g^c$ by computing $H(id, pw) \oplus g^c \oplus g^c$. In Step (3), an attacker can replace A_2 with $H(H(id, pw), sk', g^s) \oplus H(id, newpw)$ by computing as follows using captured message in Step (1):

$$\begin{aligned} A_1 &= H(id, pw) \oplus g^c \oplus H(id, newpw) \oplus g^c = H(id, pw) \oplus H(id, newpw), \\ A_2 &= H\big(H(id, pw), sk', g^s\big) \oplus H(id, newpw) \oplus A_1 \\ &= H\big(H(id, pw), sk', g^s\big) \oplus H(id, pw). \end{aligned}$$

After receiving the replaced messages $\{id, A_2\}$, the server computes the hash value $H(vpw, sk, g^s)$ using its own copies of sk and g^s. Then, the server retrieves

$H(H(id,pw),sk',g^s)$ from A_2 using the recovered $H(id,pw)$ in the Step (1) and checks whether $H(vpw,sk,\ g^s) = H(H(id,pw),sk',g^s)$ holds or not. Because it holds, the server will pass the authentication and update a new password verifier as $H(id,pw)$. The value $H(id,pw)$ is not equal to $H(id,newpw)$, and therefore all subsequent login requests of that user will be rejected until that user has re-registered to the server.

3. Proposed Protected Password Changing Scheme

This section proposes an improved user password changing scheme to overcome the above mentioned problems with Yang *et al.*'s scheme. The server stores $vpw = H(id,pw) \oplus K$ using the server's secret key K instead of $H(id,pw)$ for each client in the database to overcome stolen-verifier attack. The password change protocol allows a client to change their old password pw to a new password $newpw$. The proposed protected password change scheme is illustrated in Fig. 2.

Step (1) Client $\rightarrow$ Server: $id,\ H(id,pw) \oplus g^c, H(id,newpw) \oplus g^c$

The user submits his id and pw to the client. The client then chooses a random number c and computes g^c and chooses a new password $newpw$ and uses g^c to

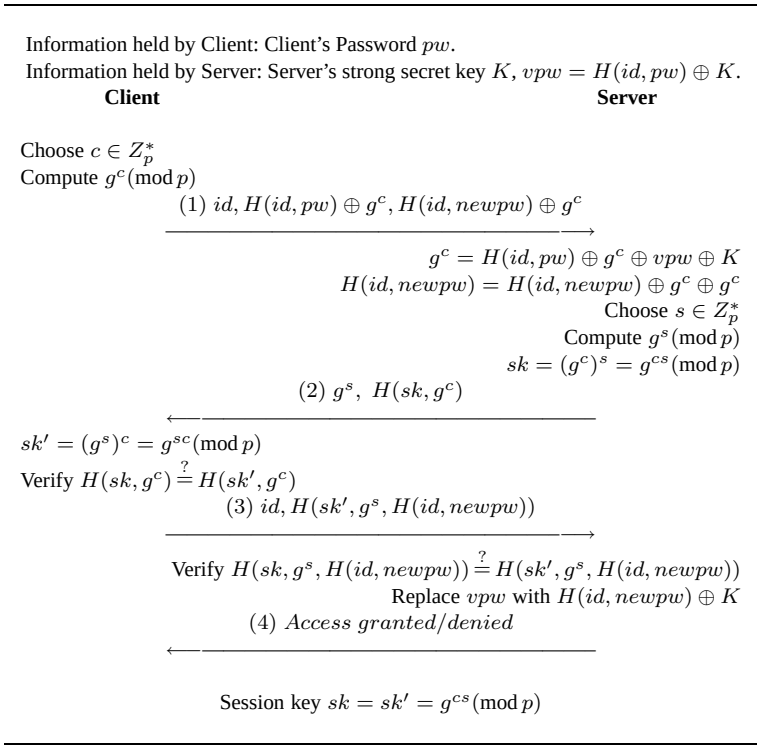

Fig. 2. Proposed protected password changing scheme.

compute $H(id, pw) \oplus g^c$ and $H(id, newpw) \oplus g^c$. The client sends its with the id as a login request to the server.

Step (2) Server $\rightarrow$ Client: $g^s,\ H(sk, g^c)$

The server retrieves g^c from $H(id, pw) \oplus g^c$ by computing $H(id, pw) \oplus g^c \oplus vpw \oplus K$. Then, the server uses the recovered g^c to obtain $H(id, newpw)$ from $H(id, newpw) \oplus g^c$ by computing $H(id, newpw) \oplus g^c \oplus g^c$. Then, the server chooses a random number s and computes g^s and $sk = (g^c)^s = g^{cs}$. Then, the server uses its own sk and the recovered g^c to compute $H(sk, g^c)$. The server sends its as the server's authentication token to the client.

Step (3) Client $\rightarrow$ Server: $id,\ H(sk', g^s, H(id, newpw))$

The client computes $sk' = (g^s)^c = g^{sc}$ and $H(sk', g^c)$, then verifies the consistency between the retrieved $H(sk', g^c)$ and the received $H(sk, g^c)$. If the result is positive, the client computes $H(sk', g^s, H(id, newpw))$ and sends this client authentication token with the id to the server.

Step (4) Server $\rightarrow$ Client: $Access\ granted\ /\ denied$

The server computes the hash value $H(sk, g^s, H(id, newpw))$ using its own copies of sk, g^s and the recovered $H(id, newpw)$ in the Step (1) and checks whether $H(sk, g^s, H(id, newpw))$ = $H(sk', g^s, H(id, newpw))$ holds or not. If it holds, the server can ensure the client is legal and replaces vpw with $H(id, newpw) \oplus K$.

After mutual authentication between the client and the server, $sk = sk' = g^{cs}$ is used as the session key, respectively.

4. Security Analysis

In this section, we will provide the proof of correctness of proposed password changing 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. 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.

DEFINITION 3. The discrete logarithm problem (DLP) is the following: given a prime p, a generator g of Z_p^*, and an element $\beta \in Z_p^*$, find the integer α, $0 \leqslant \alpha \leqslant p-2$, such that $g^\alpha \equiv \beta (\mathrm{mod}\, p)$.

DEFINITION 4. The Diffie-Hellman problem (DHP) is the following: given a prime p, a generator g of Z_p^*, and an element $g^a (\mathrm{mod}\, p)$ and $g^b (\mathrm{mod}\, p)$, find $g^{ab} (\mathrm{mod}\, p)$.

Theorem 1. *The proposed scheme can resist the replay attack.*

Proof. The attacker intercepts $\{id, H(id, pw) \oplus g^c, H(id, newpw) \oplus g^c\}$ sent by the client in Step (1) and uses it to impersonate the client when sending the next login

message. However, he/she has no ability to make a correct response $\{id, H(sk', g^s, H(id, newpw)\}$ in Step (3) because the random challenge g^c and g^s separately generated by the client and server are 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 is to the other party. Furthermore, obtaining g^c and g^s is computationally infeasible, as it is a discrete logarithm problem. Therefore, without knowing g^c and g^s, the attacker cannot impersonate the client or the server.

Theorem 2. *The proposed scheme can resist the password guessing attacks.*

Proof. Because the on-line guessing attacks can be prevented by letting the server take appropriate intervals between trials. As we described in DEFINITION 1, weak passwords with low entropy are easily guessed by off-line guessing attacks. To avoid this problem, there must be no verifiable information on passwords in message exchanges. In the improved scheme, the password pw is protected by the client's random integer g^c. As such, no one can reveal the pw from the client's login message $\{id, H(id, pw) \oplus g^c, H(id, newpw) \oplus g^c\}$ without knowing the client's random integer g^c. If the attacker wants to guess the client's password, he/she first guesses a password pw' and then finds $g^c = H(id, pw) \oplus g^c \oplus H(id, pw')$. However, the attacker has to break the discrete logarithm problem and Diffie-Hellman problem to find g^c in Step (1) and g^{cs} in Step (2), respectively. Hence, without knowing g^c and g^{cs}, the attacker cannot verify the correctness of the guessed password by checking $H(id, pw) \oplus g^c = H(id, pw') \oplus g^c$ in Step (1) and $H(sk', g^s, H(id, newpw)) = H(g^{cs}, g^s, H(id, newpw'))$ in Step (3), respectively. For the same reason, the attacker cannot guess session key g^{cs} from server's response message $\{g^s, H(sk, g^c)\}$ in Step (2) and from client's response message $\{id, H(sk', g^s, H(id, newpw))\}$ in Step (3) because $H(\cdot)$ is a secure one-way hash function.

Theorem 3. *The proposed scheme can resist the stolen-verifier attack.*

Proof. Servers are always the target of attacks. An attacker may acquire $vpw = H(id, pw) \oplus K$ stored in the server. However, without knowing the server's strong secret key K, the attacker cannot forge a login request to pass the authentication, as $H(id, pw)$ is hidden in $vpw = H(id, pw) \oplus K$ using the server's strong secret key K, thus the correctness of the guessed password cannot be verified by checking $H(id, pw') \oplus K' = vpw$, where pw' is guessed client's password and K' is guessed server's strong secret key.

Theorem 4. *The proposed scheme can resist the server spoofing attack.*

Proof. The improved scheme uses the client's password $H(id, pw)$ to ensure that only the real server can obtain g^c and $H(id, newpw)$ from the client's login message $\{id, H(id, pw) \oplus g^c, H(id, newpw) \oplus g^c\}$. After verifying the identity of the client, the server sends a correct response $\{g^s, H(sk, g^c)\}$ to the client to achieve mutual authentication in Step (2). Due to the discrete logarithm problem and Diffie-Hellman problem,

an illegal client cannot computes session key g^{cs} from $\{g^s, H(sk, g^c)\}$ and then make a correct response $\{id, H(sk', g^s, H(id, newpw)\}$ in Step (3).

Theorem 5. *The proposed scheme can resist the denial of service attack.*

Proof. In Step (1) of proposed scheme, an attacker can replace new password digest $H(id, newpw) \oplus g^c$ with current client password digest $H(id, pw) \oplus g^c$. In Step (2), after receiving the replaced messages $\{id, H(id, pw) \oplus g^c, H(id, pw) \oplus g^c\}$, the server retrieves g^c from $H(id, pw) \oplus g^c$ by computing $H(id, pw) \oplus g^c \oplus vpw \oplus K$. Then, the server uses the recovered g^c to obtain replaced password verifier $H(id, pw)$ from $H(id, pw) \oplus g^c$ by computing $H(id, pw) \oplus g^c \oplus g^c$. However, in the improved scheme, a check item $H(sk', g^s, H(id, pw))$ for new password is added in Step (3). The server updates replaced password verifier $H(id, pw) \oplus K$ only if the computed hash value $H(sk, g^s, H(id, pw))$ is equivalent to the received $H(sk', g^s, H(id, pw))$. But an attacker cannot computes this session key sk' in hashed value $H(sk', g^s, H(id, pw))$ because the discrete logarithm problem, Diffie-Hellman problem and a secure one-way hash function.

Theorem 6. *The proposed scheme provides the mutual authentication.*

Proof. The improved scheme uses the Diffie-Hellman key exchange algorithm (Diffie and Hellman, 1976) to provide mutual authentication, then the key is explicitly authenticated by a mutual confirmation session key, $sk = g^{cs}$.

Theorem 7. *The proposed scheme provides the forward secrecy.*

Proof. In the improved scheme, since the Diffie-Hellman key exchange algorithm is used to generate a session key g^{cs}, forward secrecy is ensured, as an attacker with a compromised client's password pw is only able to obtain the g^c and g^s from an earlier session. In addition, it is also computationally infeasible to obtain the session key g^{cs} from g^c and g^s, as it is a discrete logarithm problem and Diffie-Hellman problem.

5. Conclusions

The current paper demonstrated that Yang *et al.*'s protected password change scheme is vulnerable to stolen-verifier attack and denial of service attack and presented an improved protected password change scheme to isolate such problems. In contrast to Yang *et al.*'s protected password changing scheme and the existing password change schemes using server's public key, the proposed scheme can securely update user passwords without a complicated process and server's public key. Therefore the proposed scheme is more secure.

Acknowledgements

We would like to thanks the anonymous reviewers for their helpful comments. This work was supported by the Brain Korea 21 Project in 2004.

References

Diffie, W., and M. Hellman (1976). New directions in cryptography. *IEEE Transaction on Information Theory*, **IT-22**(6), 644–654.

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 (2003). Cryptoanalysis of variant of Peyravian-Zunic's password authentication scheme. *IEICE Transactions on Communications*, **E86-B**(5), 1682–1684.

Lin, C.L., and T. Hwang (2003). A password authentication scheme with secure password updating. *Computer & Security*, **22**(1), 68–72.

Menezes, A.J., P.C. Oorschot and S.A. Vanstone (1997). *Handbook of Applied Cryptograph*. CRC Press, New York.

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

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.

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.

E.-J. Yoon received his MS degree in computer engineering from Kyung Il University in 2002, South Korea. He is now working toward the PhD degree in Kyungpook National University. His research interests include cryptography and network security.

E.-K. Ryu received her MS degree in information and communication engineering from Keimyung University in 1999, South Korea. She is now working toward the PhD degree in Kyungpook National University. Her research interests include cryptographic protocols for network security.

K.-Y. Yoo received his BS degree in education of mathematics from Kyungpook National University in 1976; the MS degree in computer engineering from Korea Advanced Institute of Science and Technology in 1978 and the PhD degree in computer science from Rensselaer Polytechnic Institute, New York, USA, in 1992. He is now a professor at Department of Computer Engineering, Kyungpook National University. His current research interests are wireless security and cryptography.

Yang ir bendraautorių apsaugoto slaptažodžio keitimo schemos atakos ir sprendimai

Eun-Jun YOON, Eun-Kyung RYU, Kee-Young YOO

Neseniai Yang su bendraautoriais pasiūlė pagerintą Tseng su bendraautoriais apsaugoto slaptažodžio keitimo schemą, kuri gali atsilaikyti prieš serviso paneigimo ataką. Tačiau ši pagerinta schema yra jautri pavogto-tikrintojo ir serviso paneigimo atakoms. Šis straipsnis demonstruoja Yang ir bendraautorių schemos pažeidžiamumą dvejomis paprastomis atakomis ir šių problemų sprendimui pristato pagerintą apsaugoto slaptažodžio keitimo schemą. Priešingai Yang ir bendraautorių apsaugoto slaptažodžio keitimo schemai ir egzistuojančioms serverio viešą raktą naudojančioms slaptažodžio keitimo schemoms, pasiūlyta schema gali saugiai atnaujinti slaptažodžius be sudėtingo proceso ir be serverio viešo rakto.