Stolen-Verifier 공격과 Impersonation 공격에 안전한 개선된 OSPA 프로토콜

곽 진[†] · 오 수 현[††] · 양 형 규[†††] · 원 동 호[††††]

요 약

인터넷에서의 사용자 인증은 안전한 통신을 위해 가장 중요한 서비스 중의 하나이다. 비록 패스워드 기반 메커니즘이 네트워크 상에서의 사용자 인증을 위해 가장 많이 쓰이는 방법이기는 하나, 사용자들이 기억하기 쉬운 패스워드(easy-to-remember)를 사용하므로, 사전공격(dictionary attack)에 취약한 것과 같은 근본적인 문제점들을 가지고 있다. 이러한 사전공격을 방지하기 위한 방법들의 경우에는 높은 계산량을 필요로 한다. 본 논문에서는 이러한 문제를 해결하기 위한 최근에 발표된 OSPA 프로토콜에 대하여 설명하고, OSPA 프로토콜이 stolen-verifier 공격과 impersonation 공격에 취약함을 보인다. 그리고 이러한 공격들에 안전한 개선된 OSPA 프로토콜을 제안한다. 제안하는 프로토콜은 스마트 카드에 탑재된 co-processor를 통해 암호학적 연산이 수행되므로 사용자에게 낮은 계산량을 제공한다.

An Improved Optimal Strong-Password Authentication (I-OSPA) Protocol Secure Against Stolen-Verifier Attack and Impersonation Attack

Jin Kwak[†] · Soohyun Oh[††] · Hyungkyu Yang[†††] · Dongho Won[††††]

ABSTRACT

In the Internet, user authentication is the most important service in secure communications. Although password-based mechanism is the most widely used method of the user authentication in the network, people are used to choose easy-to-remember passwords, and thus suffers from some innate weaknesses. Therefore, using a memorable password is vulnerable to the dictionary attacks. The techniques used to prevent dictionary attacks bring about a heavy computational workload. In this paper, we describe a recent solution, the Optimal Strong-Password Authentication (OSPA) protocol, and that it is vulnerable to the stolen-verifier attack and an impersonation attack. Then, we propose an Improved Optimal Strong-Password Authentication (I-OSPA) protocol, which is secure against stolen-verifier attack and impersonation attack. Also, since the cryptographic operations are computed by the processor in the smart card, the proposed I-OSPA needs relatively low computational workload and communicational workload for user.

키워드 : 인증(Authentication), 패스워드(Password), 해쉬 함수(Hash Function), Stolen-Verifier 공격(Stolen-Verifier Attack), Impersonation 공격(Impersonation Attack)

1. Introduction

The Internet communications have been increasing considerably recently and many users use them to send private documents. However, there is demerit that such documents can be tapped. Therefore, it is necessary to authenticate users for secure communications. That is to say, above communications need for authentication over the remote sever and the user has become very important [1]. The authentication technique can be guaranteed by the use of a one-way hash function with which it is easy to compute $f(x)$ from x and difficult to compute x such that $y=f(x)$. Usually the password is hashed and stored in the server to prevent stealing by attackers [2-4].

A password-based authentication mechanism is the most widely used method for the user authentication in the network. Existing password-based authentication schemes can be regarded as two types, one uses the weak-password(easy-to-remember) type, and the other use the strong-password(well-chosen) type [5-10]. Although the strong-

† 준 회 원 : 성균관대학교 대학원 정보통신공학부
†† 정 회 원 : 호서대학교 컴퓨터공학부 교수
††† 정 회 원 : 강남대학교 컴퓨터미디어공학부 교수
†††† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2004년 2월 23일, 심사완료 : 2004년 6월 12일

password type needs additional tamper-resistant storage devices to store passwords, it provides advantages such as lower computational workload, simple design and so on.

Recently, in [11], Sandirigama *et al.* proposed the SAS (Simple And Secure strong-password authentication) protocol that eliminates the man-in-the-middle attack and reduces storage, processing, and communication workload. The SAS protocol intended to be superior to previously well-known protocols, such as the Lamport method [12, 13], the CINON(chained one-way data verification method) [14], and the PERM(Privacy Enhanced information Reading and writing Management method) [15].

However, in [16], Lin *et al.* it pointed out that SAS protocol vulnerable to the reply attack and denial of service attack, and they also proposed OSPA(Optimal Strong-Password Authentication) protocol, which has solved the security problems of the SAS protocol. Later, in [17, 18], Chen-Ku and Tsuji-Shimizu show that the SAS and OSPA protocols are vulnerable to the active attack.

In this paper, first, we describe the OSPA protocol, Stolen-Verifier attack, and Impersonation attack on OSPA protocol, and then propose the I-OSPA(Improved Optimal Strong-Password Authentication) protocol secure against aforementioned attacks.

This paper is organized as follows, in section 2, we review the OSPA protocol. In section 3, we describe vulnerability of the OSPA protocol. Then we propose the I-OSPA protocol which is secure against stolen-verifier attack and impersonation attack in section 4. In section 5, we analyze the security of the I-OSPA protocol. Finally, we make our conclusions in section 6.

2. The OSPA protocol

The OSPA protocol consists of the two phases, the registration phase and the authentication phase. To starts communication between the user and the server, the user need to registration for the successful login. The registration process is done only once and authentication is needed every time the user logs in. The following definitions and notations are used in this paper.

- *User i* : the user that who uses the protocol for authentication
- *Server s* : the server that authenticates the users
- *Eve* : an attacker
- ID_i : the identity of the user i
- P_i : the password of the user i
- Q_i : the random integer chosen by the server s
- h : a cryptographic one-way hash function. $h(x)$ means x is hashed once, and $h^2(x)$ means x is hashed twice
- V : the user-verifier generated by the server
- T, T' : the synchronized time
- SR : the Service Request by the user, it is message of the allow login to the server
- n : the number of nth authentication sessions
- $\|$: the concatenation
- $\oplus$: the bitwise XOR operation

[*Registration Phase*]

(Figure 1) shows the initial registration phase of the OSPA protocol.

User i		Server s
ID_i, P_i computes **(R1)** $h^2(P_i \oplus 1)$	**(R2)** ID_i, $h^2(P_i \oplus 1)$ ----------> (*secure channel*)	stores **(R3)** ID_i, $h^2(P_i \oplus 1)$ sets $n=1$

(Figure 1) Registration phase of the OSPA protocol

R1. The user computes first verifier $h^2(P_i \oplus 1)$ with password P_i.

R2. Then the user sends ID_i and $h^2(P_i \oplus 1)$ to the server through a secure channel.

R3. After receiving the registration message, the server stores ID_i, $h^2(P_i \oplus 1)$, and sets $n=1$ for next authentication.

[*Authentication Phase*]

After the registration phase, when the user wants to login subsequently he executes the nth authentication protocol. (Figure 2) shows the authentication phase of the OSPA protocol.

User i		Server s
ID_i	**(A1)** ID_i, SR ---------->	ID_i, n, $h^2(P_i \oplus n)$
input P_i, n **(A3)** computes $c_1 = h(P \oplus n) \oplus h^2(P \oplus n)$ $c_2 = h^2(P_i \oplus (n+1)) \oplus h(P_i \oplus n)$ $c_3 = h^3(P_i \oplus (n+1))$	<---------- **(A2)** n c_1, c_2, c_3 ---------->	**(A4)** checks $c_1 \neq c_2$ if it does, computes y_1, y_2, $y_1 = c_1 \oplus h^2(P_i \oplus n) = h(P_i \oplus n)$ $y_2 = c_2 \oplus y_1 = h^2(P_i \oplus (n+1))$ **(A5)** if, $h(y_1) =$ stored $h^2(P_i \oplus n)$ $h(y_2) = c_3$ updates $h^2(P_i \oplus (n+1))$ set $n = n+1$

(Figure 2) Authentication phase of the OSPA protocol

A1. The user issues a login SR, and sends it with ID_i to the server.
A2. The server responds to the user with nth sequential number n.
A3. The user computes c_1, c_2, and c_3 using password P_i and received n, then the user sends c_1, c_2, and c_3 to the server through an insecure network such as the Internet.

$c_1 = h(P \oplus n) \oplus h^2(P \oplus n)$: for the current authentication session
$c_2 = h^2(P_i \oplus (n+1)) \oplus h(P_i \oplus n)$: for updating the next password verifier
$c_3 = h^3(P_i \oplus (n+1))$: for an integrity check of updating

A4. After receiving c_1, c_2, and c_3, the server first checks whether $c_1 \neq c_2$. If it does, the server uses the stored $h^2(P_i \oplus n)$ to compute y_1 and y_2.

$y_1 = c_1 \oplus h^2(P_i \oplus n) = h(P_i \oplus n)$
$y_2 = c_2 \oplus y_1 = h^2(P_i \oplus (n+1))$

A5. Then, the server passes the authentication only if $c_1 \neq c_2$, $h(y_1)$ is equal to the stored $h^2(P_i \oplus n)$, and $h(y_2)$ is equal to c_3.

$c_1 \neq c_2$
$h(y_1) = h(h(P_i \oplus)) \stackrel{?}{=}$ stored $h^2(P_i \oplus n)$
$h(y_2) = h(h^2(P_i \oplus (n+1))) \stackrel{?}{=} c_3 \ (= h^3(P_i \oplus (n+1)))$

If the authentication is successful, the server updates $h^2(P_i \oplus (n+1))$ and sets $n = n+1$ for the next authentication session.

$h^2(P_i \oplus n) \rightarrow h^2(P_i \oplus (n+1))$
$n \rightarrow n+1$

3. Attacks on the OSPA protocol

3.1 Stolen-Verifier attack

Suppose the Eve has stolen $h^2(P_i \oplus n)$ after the user's $(n-1)$th login. Then, during the user's nth login process, Eve blocks and copies the message transmitted in (A3). Next, Eve derives $h(P_i \oplus n)$ from captured c_1 and stolen $h^2(P_i \oplus n)$. Also, Eve selects a password P_i', and computes c_2' and c_3'.

$$c_2' = h^2(P_i' \oplus (n+1)) \oplus h(P_i \oplus n)$$
$$c_3' = h^3(P_i' \oplus (n+1))$$

Next, Eve can masquerade as the user to send c_1, c_2', and c_3' to the server. In *A4*, the server uses the stored $h^2(P_i \oplus n)$ to compute y_1 and y_2'.

$$y_1 = c_1 \oplus h^2(P_i \oplus n) = h(P_i \oplus n)$$
$$y_2' = c_2' \oplus y_1 = h^2(P_i' \oplus (n+1))$$

Then, the server passes the authentication only if $c_1 \neq c_2'$, $h(y_1)$ equals the stored $h^2(P_i \oplus n)$, and $h(y_2')$ equals c_3'.

$c_1 \neq c_2'$
$h(y_1) = h(h(P_i \oplus n)) \stackrel{?}{=}$ stored $h^2(P_i \oplus n)$
$h(y_2') = h(h^2(P_i' \oplus (n+1))) \stackrel{?}{=} c_3' (= h^3(P_i' \oplus (n+1)))$

If the authentication is successful, the server updates $h^2(P_i' \oplus (n+1))$ and sets $n = n+1$ for the next authentication session.

$$h^2(P_i \oplus n) \rightarrow h^2(P_i' \oplus (n+1))$$
$$n \rightarrow n+1$$

Since $h(y_1)$ equals the stored $h^2(P_i \oplus n)$ and $h(y_2') = c_3'$ holds, the server will update the current verifier for the user's password with $h^2(P_i' \oplus (n+1))$ for the user's next login. Although the user successfully login in this session, he will be rejected in the next login process because he cannot authenticate himself to the server by using the password P_i. On the other hand, Eve can use P_i' to login as the user from now on. Therefore, even though the server has stored the wrong verifier for the next the user's login, the server believes it is correct. The OSPA protocol is vulnerable to the stolen-verifier attack.

3.2 Impersonation attack

We assume that Eve is an attacker who tries to masquerade as a legitimate user in the authentication session. When the user tries to be authenticated by the server on the $(n+1)$th authentication session, it is assumed that Eve has intercepted transmission data from the $(n-1)$ to the $(n+1)$th authentication sessions.

- $(n-1)$th authentication data

$$c_{(n-1)1} = h(P_i \oplus (n-1)) \oplus h^2(P_i \oplus (n-1))$$
$$c_{(n-1)2} = h^2(P_i \oplus n) \oplus h(P_i \oplus (n-1))$$
$$c_{(n-1)3} = h^3(P_i \oplus n)$$

- nth authentication data

$$c_{(n)1} = h(P_i \oplus n) \oplus h^2(P_i \oplus n)$$
$$c_{(n)2} = h^2(P_i \oplus (n+1)) \oplus h(P_i \oplus n)$$
$$c_{(n)3} = h^3(P_i \oplus (n+1))$$

- $(n+1)$th authentication data

$$c_{(n+1)1} = h(P_i \oplus (n+1)) \oplus h^2(P_i \oplus (n+1))$$
$$c_{(n+1)2} = h^2(P_i \oplus (n+2)) \oplus h(P_i \oplus (n+1))$$
$$c_{(n+1)3} = h^3(P_i \oplus (n+2))$$

Eve tries to masquerade as a legitimate the user, she has to compute $c'_{(n+1)2}$ and replaces $c'_{(n+1)3}$ with $c_{(n-1)3}$ from intercepted data.

$$c'_{(n+1)2} = c_{(n)1} \oplus c_{(n)2} \oplus c_{(n+1)1}$$
$$= h^2(P_i \oplus n) \oplus h(P_i \oplus (n+1))$$
$$c'_{(n+1)3} = h^3(P_i \oplus n)$$

Next, Eve sends the $c_{(n+1)1}$, $c'_{(n+1)2}$, and $c'_{(n+1)3}$ to the server for the $(n+1)$th authentication session.

$$c_{(n+1)1} = h(P_i \oplus (n+1)) \oplus h^2(P_i \oplus (n+1))$$
$$c'_{(n+1)2} = h^2(P_i \oplus n) \oplus h(P_i \oplus (n+1))$$
$$c'_{(n+1)3} = h^3(P_i \oplus n)$$

After receiving the above data, the server first checks whether $c_{(n+1)1} \neq c'_{(n+1)2}$. If it does, the server uses the stored $h^2(P_i \oplus n)$ to compute y_1' and y_2'.

$$y_1' = c_{(n+1)1} \oplus h^2(P_i \oplus (n+1)) = h(P_i \oplus (n+1))$$
$$y_2' = c_{(n+1)2} \oplus y_1 = h^2(P_i \oplus n)$$

Then the server compares $h(y_1')$ with the stored $h^2(P_i \oplus (n+1))$, and compares $h(y_2)$ with the received $c'_{(n+1)3}$. These are the same, so Eve is authenticated.

$$h(y_1') = h(h(P_i \oplus (n+))) \stackrel{?}{=} \text{stored } h^2(P_i \oplus (n+1))$$
$$h(y_2') = h(h^2(P_i' \oplus n)) \stackrel{?}{=} c'_{(n+1)3} = h^3(P_i \oplus n)$$

If these are correct, the server updates $h^2(P_i \oplus n)$ and sets $n = n+1$ for the next authentication session.

$$h^2(P_i \oplus (n+1)) \rightarrow h^2(P_i \oplus n)$$
$$n \rightarrow n+1$$

In the future, if Eve wants to login, she alternately sends the following two sets. Eve used the first set in the $(n+2k)$th authentication session, where k is a natural number.

$$c_1 = h(P_i \oplus n) \oplus h^2(P_i \oplus n)$$
$$c_2 = h^2(P_i \oplus (n+1)) \oplus h(P_i \oplus n)$$
$$c_3 = h^3(P_i \oplus (n+1))$$

And Eve uses the second set in the $(n+2k+1)$th authentication session, then the verifier is changed by the server to $h^2(P_i \oplus (n+1))$. In this way, Eve can impersonate the legitimate the user whenever she wants to login.

$$c_1 = h(P_i \oplus (n+1)) \oplus h^2(P_i \oplus (n+1))$$
$$c_2 = h^2(P_i \oplus n) \oplus h(P_i \oplus (n+1))$$
$$c_3 = h^3(P_i \oplus n)$$

4. Improved Optimal Strong Password Authentication (I-OSPA) protocol

In this section, we propose I-OSPA protocol using a smart card to withstand stolen-verifier attack and impersonation attack. The I-OSPA protocol also consists of two phases as with the OSPA protocol ; the registration phase and the authentication phase. These two phases are as follows.

4.1 Registration phase

(Figure 3) shows the initial registration phase of the I-OSPA protocol.

User i		Server s
ID_i, P_i **(IR1)** computes $h^2(P_i \oplus 1)$	**(IR2)** ID_i, P_i, $h^2(P_i \oplus 1)$ ----------> *(secure channel)* Smart Card <---------- *(secure channel)*	**(IR3)** stores ID_i, $h^2(P_i \oplus 1)$ issue Smart Card with n =1

(Figure 3) Registration phase of the I-OSPA protocol

IR1. The user computes first verifier $h^2(P_i \oplus 1)$ with the password P_i.
IR2. Then the user sends a message ID_i and $h^2(P_i \oplus 1)$ to the server through a secure channel.
IR3. After receiving the registration message, the server stores ID_i and $h^2(P_i \oplus 1)$. Then, the server sets $n = 1$ and then it is stored into the user i's smart card. Then the server sends the smart card to the user through a secure channel.

4.2 Authentication phase

After the registration phase, when the user wants to login subsequently, he executes the nth authentication protocol. The user i inserts his smart card into the input device(e.g. card reader), and types in ID_i and P_i. (Figure 4) shows the authentication phase of the I-OSPA protocol.

User i		Server s
ID_i	**(IA1)** $SR = (ID_i \| n)$ ---------->	ID_i, $h^2(P_i \oplus n)$ **(IA2)** computes V $V = h(ID_i \| n \| Q_i \| T)$
insert Smart Card inputs P_i, stored n, **(IA3)** computes $c_1 = V \oplus h(P_i \oplus n) \oplus h(T')$ $c_2 = h(h^2(P_i \oplus n) \| T') \oplus h(P_i \oplus (n+1)) \oplus h(T')$ updates $n \rightarrow n+1$	<---------- V ----------> c_1, c_2	**(IA4)** checks $c_1 \neq c_2$ if it does, computes y_1, y_2, y_3, y_4 $y_1 = c_1 \oplus V = h(P_i \oplus n) \oplus h(T')$ $y_2 = y_1 \oplus h(T) = h(P_i \oplus n)$ $h(y_2)$?= stored $h^2(P_i \oplus n)$ **(IA5)** $y_3 = c_2 \oplus h(h(y_2) \| T)$ $= h(P_i \oplus (n+1)) \oplus h(T')$ $y_4 = y_3 \oplus h(T) = h(P_i \oplus (n+1))$ updates $h^2(P_i \oplus n) \rightarrow h(y_4) = h^2(P_i \oplus (n+1))$

(Figure 4) Authentication phase of the I-OSPA protocol

IA1. Smart card issues a login $SR = (ID \| n)$, and sends it to the server.
IA2. The server issue a service request(SR) received time T, and computes verifier $V = h(ID_i \| n \| Q_i \| T)$. Then sends it to the user. Where Q_i is a random integer chosen by the server.
IA3. After receiving V from the server, the smart card issues a received time T', then he computes c_1, c_2, and updates $n = n+1$.

$$c_1 = V \oplus h(P_i \oplus n) \oplus h(T')$$
$$c_2 = h(h^2(P_i \oplus n \| T') \oplus h(P_i(n+1)) \oplus h(T')$$
$$n \rightarrow n+1$$

next, the user sends c_1 and c_2 to the server through a public network such as the Internet.
IA4. After receiving c_1 and c_2, the server first checks whether $c_1 \neq c_2$. If it does, the server computes y_1, y_2, y_3, and y_4.

$$y_1 = c_1 \oplus V = h(P_i \oplus n) \oplus h(T')$$
$$y_2 = y_1 \oplus h(T) = h(P_i \oplus n)$$

Then, the server passes the authentication only if $h(y_2)$ equals the stored.

$$h^2(P_i \oplus n)$$
$$h(y_2) \stackrel{?}{=} \text{stored } h^2(P_i \oplus n)$$

IA5. If the authentication is successful, the server computes an update of the information. And then the server updates $h^2(P_i \oplus (n+1))$.

$$y_3 = c_2 \oplus h(h(y_2)) \| T) = h(P_i \oplus (n+1)) \oplus h(T)$$
$$y_4 = y_3 \oplus h(T) = h(P_i \oplus (n+1))$$
$$h^2(P_i \oplus n) \rightarrow h(y_4) = h^2(P_i \oplus (n+1))$$

5. Analysis of the Proposed Protocol

In this section, we will show that the proposed I-OSPA protocol can withstand the stolen-verifier attack, impersonation attack, and replay attack. Due to the fact that the I-OSPA protocol includes random integer Q_i and a synchronized time, it provides implicit the user authentication in the authentication phase.

[*Stolen-verifier attack*]

In the network applications, the server stores the password-verifier of the users instead of the plain passwords. The stolen-verifier attack means that Eve who steals the password-verifier from the server, so that she can then use it to masquerade as a legitimate the user in the authentication phase.

In the proposed I-OSPA protocol, we assume that Eve has stolen the password verifier $h^2(P_i \oplus n)$ and intercepted the user's authentication request message c_1 and c_2 over the public network. She cannot derive synchronized time T and T' from the stolen password-verifier $h^2(P_i \oplus n)$, since password P_i is kept a secret, and h is a strong one-way hash function. That is to say, only users who know time T' can calculate the correct authentication request message c_1, can thereby pass the authentication. Therefore, the propose I-OSPA protocol can resist the stolen-verifier attack.

[*Impersonation attack*]

Impersonation attack means that an attacker replays and forges authentication request message from the previous authentication session. In the proposed I-OSPA protocol, when the user tries authentication on the $(n+1)$th authentication session, we assume that Eve has intercepted authentication request messages from the $(n-1)$th to the $(n+1)$th authentication sessions. An Eve may impersonate legitimate users by forging an authentication request message, c_1' and c_2', and sends it to the server. Then the server computes y_1' and y_2' and checks whether these are correct. However, c_1' and c_2' cannot pass the checks because Eve does not know the valid time T', so Eve cannot derive the correct valid value of c_1' and c_2', Therefore, Eve cannot perform the impersonation attack.

[*Replay attack*]

A replay attack is an offensive action in which an Eve impersonates or deceives another legitimate participant through the reuse of information obtained in the protocol.

In the proposed I-OSPA protocol, the next verifier never appeared in the previous session such that c_1 and c_2 are an implicit verifier for the next authentication session. Therefore, Eve has no chance to devise an effective updating message from previous messages.

[*Performance Considerations*]

As in described <Table 1>, the propose I-OSPA protocol required low hash overhead and same transmission passes than previous OSPA protocol. Furthermore, our proposed I-OSPA protocol that is secure against stolen-verifier, impersonation, and reply attacks. As the proposed protocol using smart card that is performed cryptographic operations, therefore it can be reduce the computation overhead for user. Also the proposed protocol can be used in several applications like remote login and electronics payment.

〈Table 1〉 Performance evaluations of OSPA and IOSPA

	user hash iterations	server hash iteration	user-host transmissions	cryptographic module	remarks
OSPA	4	3	4	×	-
I-OSPA	3	3	4	○	· low hash overhead · improved security - stolen-verifier - impersonation - replay · using smart card

6. Conclusion

In this paper, we describe a recently proposed protocol, the OSPA protocol, and also describe that it suffers from vulnerability to the stolen-verifier attack and impersonation attack. And then we propose an Improved

Optimal Strong Password Authentication protocol, I-OSPA protocol, which is secure against stolen-verifier attack, impersonation attack, and replay attack. Also, since the cryptographic operations are computed by the processor in the smart card, the proposed I-OSPA needs relatively low computational workload and communicational workload.

References

[1] T. Arakawa, and T. Kamada, "The Internet home electronics and the information network revolution," IEICE Technical report, OFS 96-1, 1996.

[2] R. Rivest, "The MD5 message-digest algorithm," Internet Request For Comments 1321, April, 2002.

[3] National Institute of Standards and Technology(NIST), "Secure hash standard," FIPS Publication 180-1, April, 2001.

[4] W. Stallings, "Secure hash algorithm," In Cryptography and Network Security : Principles and Practice Second Edition, pp.193-197, 1999.

[5] S. Bellovin and M. Merritt, "Encrypted key exchange : Password-based protocols secure against dictionary attacks," IEEE Symposium on Research in Security and Privacy, pp.72-84, 1992.

[6] S. Bellovin and M. Merritt, "Augmented encrypted key exchange : A password-based protocol secure against dictionary attacks and password-file compromise," ACM Conference on Computer and Communications Security, pp.244-250, 1993.

[7] D. Jablon, "Strong password-only authenticated key exchange," ACM Computer Communication Review, Vol.26, No.5, pp.5-26, 1996.

[8] T. Wu, "The secure remote password protocol," Internet Society Symposium on Network and Distributed System Security, 1998.

[9] T. Kwon, "Ultimate solution to authentication via memorable password," A proposal for IEEE P1363a : Password Authentication, May, 2000.

[10] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password authentication key exchange using Diffie-Hellman," Eurocrypt 2000, LNCS 1807, May, 2000.

[11] M. Sandrigama, A. Shimizu, and M. Noda, "Simple and secure password authentication protocol(SAS)," IEICE Transactions on Communication, Vol.E83-B, No.6, pp.1363-1365, June, 2000.

[12] L. Lamport, "Password authentication with insecure communication," Communication ACM, Vol.24, No.11, pp.770-772, 1981.

[13] N. Haller, "The S/KEY(TM) one-time password system," Proc. of Internet Society symposium on Network and Distributed System Security, pp.151-158, 1994.

[14] A. Shimizu, "A dynamic password authentication method by one-way function," IEICE Transactions, Vol.J37-D-1, No.7, pp.630-636, July, 1990.

[15] A. Shimizu, T. Horioka, and H. Inagaki, "A password authentication method for contents communication on the Internet," IEICE Transactions and Communications, Vol. E81-B, No.8, pp.1666-1763, August, 1998.

[16] C. L. Lin, H. M. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication," IEICE Transactions on Communication, Vol.E84-B, No.9, pp.2622-2627, September, 2001.

[17] C. M. Chen and W. C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols," IEICE Transactions on Communication, Vol.E85-B, No.11, November, 2002.

[18] T. Tsuji and A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA," IEICE Transactions on Communication, Vol.86-B, No.7, July, 2003.

곽 진

e-mail : jkwak@dosan.skku.ac.kr

2000년 성균관대학교 생물기전공학과 (공학사)

2003년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2003년~현재 성균관대학교 대학원 정보통신공학부 박사과정

관심분야 : 암호 프로토콜, 유비쿼터스 보안 등

오 수 현

e-mail : shoh@office.hoseo.ac.kr

1998년 성균관대학교 정보공학과(공학사)

2001년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2003년 성균관대학교 대학원 전기전자 및 컴퓨터공학과(공학석사)

2004년~현재 호서대학교 컴퓨터공학부 정보보호전공 전임강사

관심분야 : 암호 키 분배 프로토콜, 유비쿼터스 보안 등

양 형 규

e-mail : hkyang@kangnam.ac.kr

1983년 성균관대학교 전자공학과(공학사)

1985년 성균관대학교 대학원 전자공학과 (공학석사)

1995년 성균관대학교 대학원 정보공학과 (공학박사)

1984년~1991년 삼성전자 컴퓨터부문 선임연구원

1995년~현재 강남대학교 컴퓨터미디어 공학부 부교수

관심분야 : 암호 프로토콜, 네트워크 보안 등

원 동 호

e-mail : dhwon@dosan.skku.ac.kr

성균관대학교 전자공학과(학사, 석사, 박사)

1978년~1980년 한국전자통신연구원 전임 연구원

1985년~1986년 일본 동경공업대 객원 연구원

1988년~1999년 성균관대학교 교학처장, 전기전자 및 컴퓨터 공학부장, 정보통신대학원장, 정보통신기술연구소장

1996년~1998년 국무총리실 정보화추진위원회 자문위원

2002년~2003년 한국정보보호학회장

2003년~2004년 성균관대학교 연구처장

1982년~현재 성균관대학교 정보통신공학부 교수,

2000년~현재 정보통신부 지정 정보보호인증기술연구센터장

관심분야 : 암호 프로토콜, 정보보안 등