Secure Communication Using Needham-Schroeder Protocol

Mandeep Kumar*
Alok Tuli**
Ruby Tuli***

ABSTRACT

Due to rapid development in the field of communication technologies, security has become a major issue in transformation level data and off-line data. It has further increased the need for online security and authentication for secure information exchange. In an internet environment, such as UNIX, a remote user has to obtain the access right from a server before doing any job. The procedure of obtaining access right is called a user authentication protocol. Password Authentication Protocols (PAP), smart cards, digital signatures, hand impressions are some of the solutions proposed, but the easiest and the most feasible solution on the network are the Password Authentication Protocols. It can be easily implemented and provides us with simple, efficient and a secure authentication mechanism. Several password authentication protocols have been introduced each claiming to withstand to the several attacks, including replay, password file compromise, denial of service, etc. The different password authentication protocols are Needham Schroeder Protocol, Lamport's Protocol, Optimal Strong Password Authentication Protocol (OSPA), and Secure Optimal Strong Password Authentication Protocol. I have mainly focused on removing two possible attacks on Needham-Schroeder protocol by modifying the protocol, and also, it includes studying and comparing the above mentioned protocols.

Keywords- *User authentication protocol, Password authentication protocol, Replay attack, Password file compromise, Denial of service.*

I. A BRIEF OVERVIEW OF PASSWORD

1. Authentication Protocols

Authentication tools [1] provide the ability to determine the identity of a party to an interaction and to ensure that a message came from who it claims to have come from. Authentication is seldom used in isolation.

*,**,*** *Lecturer*, Dept. of Computer Science, S.R.P.A. Adarsh Bhartiya College, Pathankot, India

Authentication is used as the basis for authorization determining whether a privilege will be granted to a particular user or process), privacy (keeping information from becoming known to non-participants), and non-repudiation (not being able to deny having done something that was authorized to be done based on the authentication)[2].

In an internet environment, such as UNIX, a remote user has to obtain the access right from a server before doing any job. The procedure of obtaining access right is called a user authentication protocol. User authentication via user memorable password provides convenience without needing any auxiliary devices, such as smart card. A user authentication protocol via username and password should basically withstand the off-line password guessing attack, the stolen verifier attack, and the DoS attack.

II. POSSIBLE ATTACKS ON PAP SCHEME

- Guessing Attacks
- Replay Attacks/Man-in Middle Attacks
- Stolen verifier Attacks
- Impersonation Attacks
- Denial Service Attack

2.1 Guessing Attacks

Guessing attacks, also known as dictionary attacks, occur when an attacker is able to recover the value of a secret data by searching the entire space of values. Passwords and, more generally, low-entropy secrets are especially vulnerable to guessing attacks, which gives a strong motivation for their study.

2.2 Replay Attacks / Man-in Middle Attacks

A breach of security in which information is stored without authorization and then retransmitted to trick the receiver into unauthorized operations such as false identification or authorization or a duplicate transaction.

2.3 Stolen verifier attacks

In stolen verifier attack the intruder steals the verifier to be used for the next login from the server's database. Next time the intruder can now easily login into the machine and can continue communicating over the network as an authenticated user.

2.4 Impersonation attack

This attack is in sequel with the stolen verifier attack. In this attack after acquiring the verifier for the session, now the intruder impersonates as the authorized user. He can make the malicious changes to the verifier and perform the denial of service attack as a sequence to impersonation attack.

2.5 Server spoofing attack

In a spoofing attack, the attacker creates misleading context in order to trick the victim into making an inappropriate security relevant decision. For example, there have been several incidents in which criminals set up bogus automated-teller machines, typically in the public areas of shopping malls.

III. NEEDHAM SCHROEDER PROTOCOL

The term Needham-Schroeder protocol [9] can refer to one of two communication

protocols intended for use over an insecure network, both proposed by Roger Needham and Michael Schroeder in a paper in 1978. Needham-Schroeder protocol is one of the earliest computer network authentication protocol designed for use on insecure networks (e.g. internet). It allows individuals communicating over a network to prove their identity to each other while also preventing eavesdropping.[16]. These are:

1. The Needham-Schroeder Symmetric Key Protocol is based on a symmetric encryption algorithm. It forms the basis for the Kerberos protocol. This protocol aims to establish a session key between two parties on a network, typically to protect further communication.
2. The Needham-Schroeder Public-Key Protocol, based on public-key cryptography. This is intended to provide mutual authentication between two parties communicating on a network, but in its proposed form it is insecure.

3.1 The symmetric protocol

Here, Alice (A) initiates the communication to Bob (B). Also,

- S is a server trusted by both parties
- K_{AS} is a symmetric key known only to A and S
- K_{BS} is a symmetric key known only to B and S
- N_A and N_B are nonces

The protocol can be specified as follows in security protocol notation:

$$A \rightarrow S: A, B, N_A$$

Alice sends a message to the server identifying herself and Bob, telling the server she wants to communicate with Bob.

$$S \rightarrow A: \{ N_A, K_{AB}, B, \{K_{AB}, A\}_{KBS}\}_{KAS}$$

The server generates K_{AB} and sends back to Alice a copy encrypted under K_{BS} for Alice to forward to Bob and also a copy for Alice. Since Alice may be requesting keys for several different people, the nonce assures Alice that the message is fresh and that the server is replying to that particular message and the inclusion of Bob's name tells Alice who she is to share this key with.

$$A \rightarrow B: \{K_{AB}, A\}_{KBS}$$

Alice forwards the key to Bob who can decrypt it with the key he shares with the server, thus authenticating the data.

$$B \rightarrow A: \{N_B\}_{KAB}$$

Bob sends Alice a nonce encrypted under K_{AB} to show that he has the key.

$$A \rightarrow B: \{N_B\text{-}1\}_{KAB}$$

Alice performs a simple operation on the nonce, re-encrypts it and sends it back verifying that she is still alive and that she holds the key.

The protocol is vulnerable to a replay attack. If an attacker records one run of this protocol, then subsequently learns the value K_{AB} used, she can then replay the message $\{K_{AB}, A\}_{KBS}$ to Bob, who will accept it, being unable to tell that the key is not fresh. This flaw is fixed in the Kerberos protocol by the inclusion of a timestamp.

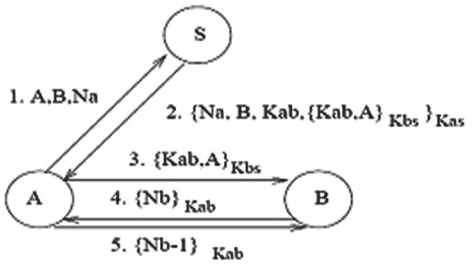

Figure 1: The Symmetric Key Protocol

3.2 The public-key protocol

This assumes the use of a public-key encryption algorithm.

Here, Alice (A) and Bob (B) use a trusted server (S) to distribute public keys on request. These keys are:

- K_{PA} and K_{SA}, respectively public and private halves of an encryption key-pair belonging to A
- K_{PB} and K_{SB}, similar belonging to B
- K_{PS} and K_{SS}, similar belonging to S. (Note this has the property that K_{SS} is used to encrypt and K_{PS} to decrypt).

The protocol runs as follows:

$$A \rightarrow S: A, B$$

A requests B's public keys from S

$$S \rightarrow A: \{K_{PB}, B\}_{KSS}$$

S responds. B's identity is placed alongside K_{PB} for confirmation.

$$A \rightarrow B: \{N_A, A\}_{KPB}$$

A invents N_A and sends it to B.

$$B \rightarrow S: B, A$$

B requests A's public keys.

$$S \rightarrow B: \{K_{PA}, A\}_{KSS}$$

Server responds.

$$B \rightarrow A: \{N_A, N_B\}_{KPA}$$

B invents N_B, and sends it to A along with N_A to prove ability to decrypt with K_{SB}.

$$A \rightarrow B: \{N_B\}_{KPB}$$

A confirms N_B to B, to prove ability to decrypt with K_{SA}

At the end of the protocol, A and B know each other's identities, and know both N_A and N_B. These nonces are not known to eavesdroppers.

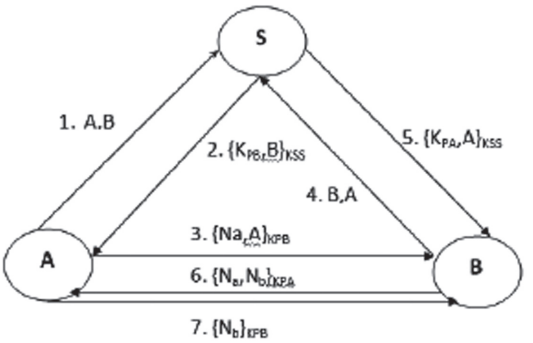

Figure 2: The Public Key Protocol

IV. ATTACKS ON THE NEEDHAM-SCHROEDER PROTOCOL

Unfortunately, this protocol is vulnerable to a man-in-the-middle attack.[11] If an impostor I can persuade A to initiate a session with him, he can relay the messages to B and convince B that he is communicating with A.

Ignoring the traffic to and from S, which is unchanged, the attack runs as follows:

$$A \rightarrow I: \{N_A, A\}_{KPI}$$

A sends N_A to I, who decrypts the message with K_{SI}

$$I \rightarrow B: \{N_A, A\}_{KPB}$$

I relay the message to B, pretending that A is communicating

$$B \rightarrow I: \{N_A, N_B\}_{KPA}$$

B sends N_B

$$I \rightarrow A: \{N_A, N_B\}_{KPA}$$

I relay it to A

$$A \rightarrow I: \{N_B\}_{KPI}$$

A decrypts N_B and confirms it to I, who learns it

$$I \rightarrow B: \{N_B\}_{KPB}$$

I re-encrypts N_B, and convinces B that he's decrypted it

At the end of the attack, B falsely believes that A is communicating with him, and that N_A and N_B are known only to A and B.

The attack was first described in a 1995 paper by Gavin Lowe [10]. The paper also describes a fixed version of the scheme, referred to as the Needham-Schroeder-Lowe protocol. The fix involves the modification of message six:

$$B \rightarrow A: \{N_A, N_B\}_{KPA}$$

With the fixed version:

$$B \rightarrow A: \{N_A, N_B, B\}_{KPA}$$

V. ANALYZING THE PROBLEM

In remotely accessed computer systems, a user identifies himself to the system by sending a secret password. There are three ways an intruder could learn the user's secret password and then impersonate him when interacting with the system:

1. By gaining access to the information stored inside the system, e.g., reading the system's password file.
2. By intercepting the user's communication with the system, e.g., eavesdropping on the line connecting the user's terminal with the system, or observing the execution of the password checking program.
3. By the user's inadvertent disclosure of his password, e.g., choosing an easily guessed password.

Compromise in Public Key Systems

Needham and Schroeder described a means for authenticating signatures using public key encryption. User A sends user B a message which has been doubly encrypted, first with A's secret key and then with B's public key. Using Needham and Schroeder's notation, this process is represented by

$$A \rightarrow B: \{\{\text{text-block}\}^{SKA}\}^{PKB}.$$

The receiver B can read the message by applying his secret key first and then A's public key, thus decrypting the text. B can convince an arbiter of the authenticity of the message and of A's authorship simply by allowing the arbiter to apply A's public key to the message after it has been decrypted by B's secret key. In a world of permanent and uncompromised keys this technique provides a foolproof authentication mechanism. [9]

VI. THE ALGORITHMS USED

6.1 RSA Algorithm

The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who

invented it in 1977 [11]. The basic technique was first discovered in 1973 by Clifford Cocks of CESG (part of the British GCHQ) but this was a secret until 1997. The patent taken out by RSA Labs has expired.

The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.[26]

Key Generation Algorithm

1. Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
2. Compute n = pq and (ö) phi = (p-1) (q-1).
3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.
4. Compute the secret exponent d, 1 < d < phi, such that ed a" 1 (mod phi). The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

Where,

- n is known as the modulus.
- e is known as the public exponent or encryption exponent or just the exponent.
- d is known as the secret exponent or decryption exponent.

6.2 Data Encryption Standard

The Data Encryption Standard (DES) [12] is a cipher (a method for encrypting information) selected as an official Federal Information Processing Standard (FIPS) for the United States in 1976, and which has subsequently enjoyed widespread use internationally. The algorithm was initially controversial, with classified design elements, a relatively short key length, and suspicions about a National Security Agency (NSA) backdoor. DES consequently came under intense academic scrutiny, and motivated the modern understanding of block ciphers and their cryptanalysis.

DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small; in January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes (see chronology). There are also some analytical results which demonstrate theoretical weaknesses in the cipher, although they are infeasible to mount in practice. The algorithm is believed to be practically secure in the form of Triple DES, although there are theoretical attacks. In recent years, the cipher has been superseded by the Advanced Encryption Standard (AES).

DES is the archetypal block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter

discarded. Hence the effective key length is 56 bits, and it is usually quoted as such.

VII. THE PROPOSED PROTOCOL BY USING NEEDHAM-SCHROEDER PROTOCOL

Alice (A) and Bob (B) use a trusted server (S) to distribute public keys on request. These keys are:

- K_{PA} and K_{SA}, respectively public and private halves of an encryption key-pair belonging to A
- K_{PB} and K_{SB}, similar belonging to B
- K_{PS} and K_{SS}, similar belonging to S. (Note this has the property that K_{SS} is used to encrypt and K_{PS} to decrypt).
- Ka and Kb are encryption keys belonging to respectively Alice (A) and Bob (B).

The protocol runs as follows:

$$A \rightarrow S: \{A, B, Ka\}_{KAS}$$

A requests B's public keys from S

$$S \rightarrow A: \{K_{PB}, B\}_{Ka}$$

S responds. B's identity is placed alongside K_{PB} for confirmation.

$$A \rightarrow B: \{N_A, A\}_{KPB}$$

A invents N_A and sends it to B.

$$B \rightarrow S: \{B, A, Kb\}_{KBS}$$

B requests A's public keys.

$$S \rightarrow B: \{K_{PA}, A\}_{KB}$$

Server responds.

$$B \rightarrow A: \{N_A, N_B\}_{KPA}$$

B invents N_B, and sends it to A along with N_A to prove ability to decrypt with K_{SB}.

$$A \rightarrow B: \{N_B\}_{KPB}$$

A confirms N_B to B, to prove ability to decrypt with K_{SA}

At the end of the protocol, A and B know each other's identities, and know both N_A and N_B. These nonces are not known to eavesdroppers.

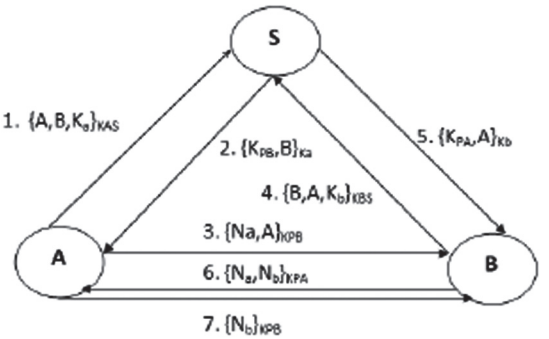

Figure 3: The Proposed Protocol

With this proposed scheme, we can able to remove two possible attacks, are Man-in middle attack and Stolen-verifier attack on Needham-Schroeder protocol. This proposed protocol significantly reduces the communication and computation cost. In this scheme, we propose a new password authentication protocol by using Needham-Schroeder protocol. Compared with Needham's scheme, our scheme is claiming to withstand to the several attacks, including replay, and impersonation attack, and also our scheme is efficient in terms of communication and computation cost.

7.1 The Comparison of various protocols

It shows the comparison of four different password authentication protocols used for secure authentication in an insecure network (internet).

Table 1

The Comparison of various password authentication protocols

	Needham	*Lamport*	*OSPA*	*Secure OSPA*
Guessing Attack	Difficult	Difficult	No	No
Replay Attack	Yes	No	No	No
Impersonation Attack	Yes	Yes	Yes	No
Stolen Verifier Attack	Yes	Yes	Yes	No
Denial of Service Attack	Yes	Yes	Yes	No

This table shows the comparison of three different password authentication protocols on hash computation, here this table

	Needham	*Lamport*	*OSPA*	*Secure OSPA*
Guessing Attack	Difficult	Difficult	No	No
Replay Attack	No	No	No	No
Impersonation Attack	No	Yes	Yes	No
Stolen Verifier Attack	Yes	Yes	Yes	No
Denial of Service Attack	Yes	Yes	Yes	No

not showing the comparison of Needham-Schroeder protocol because it has no hash computation.

Table 2

The Comparison of various password authentication protocols on hash computation

	Lamport's	*OSPA*	*Secure OSPA*
Registration	MT(Hc)	2T(Hc)	2T(Hc) T(Hs)
Authentication	(m-i)T(Hc) T(Hs)	5T(Hc) T(Hs)	12T(Hc) 9T(Hs)

Where Hc – Hashing at the Client

Hs – Hashing at the Server.

This table explains the results obtain by comparing the password authentication protocols and the proposed protocol. The Replay Attack and Impersonation Attack are removed from the Needham-Schroeder protocol.

Table 3

The comparison of various password authentication protocols with proposed one

	Needham	*Lamport*	*OSPA*	*Secure OSPA*
Guessing Attack	Difficult	Difficult	No	No
Replay Attack	Yes	No	No	No
Impersonation Attack	Yes	Yes	Yes	No
Stolen Verifier Attack	Yes	Yes	Yes	No
Denial of Service Attack	Yes	Yes	Yes	No

7.2 Some Example of Sample Runs

The proposed protocol is implemented in Java Platform; mainly in java applets are used to design client, server, and communication windows.[14][19] It is a Stand alone application to deploy the proposed protocol

for secure message transmission. With this application any user in the network can be able to send and receive the messages securely as per the proposed protocol. Socket programming is used to develop the communication between machines in a network.

It uses RSA algorithm to generate private and public keys. It uses the public key for encryption and private key is for decryption of the message. The symmetric key algorithm (DES) used to generate a symmetric key between the server and the client, which is known to only the server and the client.

In every login the user has to generate a random key and it is sent to the server, then the server will use that key as an encryption key while sending the public key of requested user. Who ever gets the key can only be able to decrypt that message.

VIII. CONCLUSIONS AND FUTURE SCOPE

8.1 The Theoretical Assessment

- In this scheme, a new password authentication protocol by using Needham-Schroeder protocol is proposed. It compared with Needham's scheme, this scheme is claiming to withstand to the several attacks, including replay, and impersonation attack, and also this scheme is efficient in terms of communication and computation cost.
- The scheme used Java platform, because it has much less debugging headaches: no pointer problems, exceptions. Stealing has never been easier: the net, portability, reusability. The Excellent documentation and Large and growing body of libraries to help: utilities, media, GUI, networking, threads, databases, cryptography, and also Flip side: versions, large libraries.
- Rivest, Adi Shamir and Leonard Adleman (RSA) Supports Encryption and Digital Signatures. It is most widely used public key algorithm gets its security from integer factorization problem. Relatively it is easy to understand and implement, and it is Patent free (since 2000)[11].
- This scheme is vulnerable to a server spoofing attack and stolen-verifier attacks.

8.2 Work desired to be done in future

- The desired work to be done in future is an attempt to remove the other two possible attacks, server spoofing attack and stolen-verifier attack on the protocol.

XI. REFERENCES

[1] Lamport, "Password authentication with insecure communication," Communications of the ACM, VOL. 24, No. 11, pp. 770-772, November 1981.

[2] Mitchell, C.J., and L. Chen, "Comments on the S/KEY user authentication scheme," ACM Operating Systems Review, VOL. 30, No. 4, pp. 12–16, 1996.

[3] Neil Haller, "The S/KEY one-time password system," In Proceedings of the ISOC Symposium on Network and Distributed System Security, San Diego, CA, pp. 151–157, February 1994.

[4] Shimizu, A, "A dynamic password authentication method by one-way function," IEICE Transactions, VOL. J73-D-I, No. 7, pp. 630–636, 1990.

[5] Sandirigama, A. Shimizu, and M. T. Noda, "Simple and secure password authentication protocol (SAS)," IEICE Transactions on Communications, VOL. E83-B, pp. 1363-1365, June 2000.

[6] C. M. Chen, and W. C. Ku, "Stolen-verifier Attack on two New Strong-password Authentication Protocol," IEICE Transactions on Communications, Vol. E85-B, No. 11, pp. 2519-2521, November 2002.

[7] Ku, C. M. Chen, and H. L. Lee, "Cryptanalysis of a variant of Peyravian-Zunic's password authentication scheme," IEICE Transactions on Communications, VOL. E86-B, No. 5, pp. 1682-1684, May 2003.

[8] Lin, H. M. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication," IEICE Transactions on Communications, VOL. E84-B, No. 9, pp. 2622-2627, September 2001.

[9] Roger Needham and Michael Schroeder, "Using encryption for authentication in large networks of computers," Communications of the ACM, Journal, VOL. 21, No. 12, December 1978.

[10] G. Lowe, "An attack on the Needham-Schroeder public key authentication protocol," Information processing letters, VOL. 56, No. 3, pp. 131–136, November 1995.

[11] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, VOL.21, No. 2, pp. 120-126, February 1977.

[12] Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), American National Standards Institute, 1998.

[13] Neil Haller, "The S/KEY one-time password system," RFC 1760, Network Working Group, February 1995.

[14] Dana Nourie, "Core Java," Fundamentals (8th Edition), VOL. 1, January 2008.

[15] William Stallings, "Cryptography and network security design and principles".

[16] John Rushby, "The Needham-Schroeder Protocol in SAL," CSL

Technical Note, October 2003 (Updated June 2005).

[17] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," In STOC'98, pp. 419-428.

[18. M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," In EUROCRYPT , VOL. 1807/2000, pp. 139-155, 2000.

[19] Robert Eckstein, "Java SE Application Design With MVC," March 2007.

[20] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm," IEEE Transactions on Information Theory, VOL. 31, pp. 469–472, 1985.

[21] J. Guttman, F. J. T. Fabrega, and L. Zuck, "The faithfulness of abstract protocol analysis: Message authentication," In Proceedings of the 8th ACM Conference on Computer and Communications Security, pp. 186–195, 2001.

[22] Lin, J. J. Shen, and M. S. Hwang, "Security enhancement for optimal strong password authentication protocol," ACM Operating Systems Review, VOL. 37, No. 2, pp. 7-12, April 2003.

[23] S. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password-file compromise," in Proceedings of the ACM Conference on Computer and Communications Security, pp. 244-2.

[24] Gavin Lowe, "An attack on the Needham-Schroeder public key authentication protocol," Information Processing Letters, VOL. 56, No. 3, pp. 131-136, November 1995.

[25]. Brian Goetz, Robert Eckstein, "An Introduction to Real-Time Java Technology," July 2008.

[26] J. Hastad, "On using RSA with low exponent in a public key network," in Lecture Notes in Computer Science: Advances in Cryptology-CRYPTO'85 Proc., pp. 403–408, 1985.

[27] DOLEV, D., AND YAO, A, "On the security of public key protocols," IEEE Transactions on Information Theory IT-29, VOL. 2, pp. 198-208, March 1983.

[28] Sandirigama, M., A. Shimizu and M.T. Noda, "Simple and secure password authentication protocol (SAS)," IEICE Transactions on Communications, VOL. E83-B, No. 6, pp. 1363–1365, 2000.

[29] Lin, C.W., J.J. Shen and M.S. Hwang, "Security enhancement for optimal strong password authentication protocol," ACM Operating Systems Review, VOL. 37, No. 2, pp. 7–12, 2003.

[30] Tsuji and A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA," IEICE Transactions on Communications, VOL. E86-B, No. 7, pp. 2182-2185, July 2003.

[31] Shimizu, A., T. Horioka and H. Inagaki, "A password authentication methods for contents communication on the Internet," IEICE Transactions on Communications, VOL. E81-B, No. 8, pp. 1666–1673, 1998.

[32] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, VOL. 29, No. 2, pp. 198–208, March 1983.