INFORMATICA, 2010, Vol. 21, No. 3, 349–359

An Improved Securer and Efficient Nonce-Based Authentication Scheme with Token-Update

Chiu-Hsiung LIAO[1], Ching-Te WANG[2], Hon-Chan CHEN[2]
[1]*General Education Center, National Chin-Yi University of Technology Taichung, Taiwan 411, ROC*
[2]*Department of Information Management, National Chin-Yi University of Technology Taichung, Taiwan 411, ROC*
e-mail: {cliao,ctwang,chenhc}@ncut.edu.tw

Received: November 2008; accepted: May 2010

Abstract. In this paper, we propose a mutual authentication scheme using nonce variable instead of Mac address and accompanying with token updates to improve the functionality. Lee *et al.* (2005a) and Shi *et al.* (2006) proposed the site authentication schemes by using the generating random numbers. The site authentication can identify a personal computer using LAN card's Mac address, but the Mac address is easily detected through Address Resolution Protocol in the Open Systems Interconnection model. Therefore, we propose an improved securer and efficient nonce-based authentication scheme providing mutual authentication to resist the replay attack, man-in-the-middle attack and Mac address attack.
Keywords: mutual authentication, token-update, Mac address.

1. Introduction

The information and communication technologies are ascendant in the recent years. People not only communicate each other, also do business or engage financial activities through network. They rely increasingly on the use of networks. When a client of a network intends to mine the data and exploit the rich resources from the internet, he needs to connect with a server to request for service. In the login instance, the client wants to make sure that he has connected to the right server, while the service provider offers service only for his genuine client. In the scenario, mutual authentication between the client and the service provider is inevitable. Therefore, an efficient, secure authentication scheme is necessary and urgent.

The encryption methods and one-way hash functions are important components in the authentication schemes, the password and nonce variables are usually used and designed in the techniques. The encryption methods and hash functions are exquisite and the passwords can be easily chosen by a network user. Thus, the password-based authentication schemes are widely used in the authentication schemes. Some of them are carried out with other mechanics to enhance the functioality and to strengthen the reliability of the scheme. For instance, Yang *et al.* (1999) proposed two password authentication schemes

employing smart cards; one of his schemes was incorporated with timestamps and the other with nonces. More else which are also carried out along with smart card are also proposed (Halevi *et al.*, 1998; Juang, 2004; Lee *et al.*, 2005a; Shieh *et al.*, 2006; Yoon *et al.*, 2005). The scheme involving timestamps derives unavoidably the synchronization concerns, because the synchronized clocks are required; and the password-based schemes are vulnerable to a malicious attack, because the selected password could be too short, based on dictionary words, or related to personal information. Aside the potential possibility, there are couple of problems need to take care of: writing the password down, retaining the same password for a long period and using the same password on multiple systems (Furnell, 2007). Some improved password-based schemes are introduced (Goyal *et al.*, 2006; Jiang *et al.*, 2004; Lin *et al.*, 2001; Phan, 2006; Yang *et al.*, 2005).

2005; Chien *et al.*, 2003; Hwang *et al.*, 2002; Lee *et al.*, 2005a; Liao *et al.*, 2009; Liaw *et al.*, 2006; Shi *et al.*, 2006; Shieh *et al.*, 2006; Yang *et al.*, 2005), the advantage is that the nonces are always fresh and can be generated easily, conveniently. Especially, it is much more easily and conveniently for a remote server to generate the nonces. Therefore, the nonce-based authentication methods are also frequently discussed. Juang (2004) proposed a nonce-based authentication method by using the nonce variable, which was updated at every login phase. In the scheme, they apply a symmetric encryption algorithm to authenticate the user and server. The implementation operations are cost for the exponential computations. Simultaneously, Chen *et al.* (2004) also proposed a nonce-based scheme accompanying the use of hash function. However, the computation cost is still expensive and the user bent with an evaluation burden on generating two nonces in the login instance. Recently, Chen and Yeh (2005) proposed an efficient method to improve Chen *et al.* (2004) scheme. However, both the service provider and the user are required to compute the nonce in each side. Obviously, it is difficult and inconvenient for the user to generate a random number.

Lee *et al.* (2005a) proposed a site authentication scheme with token updates trying to strengthen the functionality. Being short of mutual authentication between the user and the remote server, the scheme is not secure to resist attacks. Shi and Yoo (2006) in 2006 proposed another site authentication scheme to improve the security. However, there are some latent weaknesses, when the physical address of media access control (Mac) is used. In fact, the authentication scheme uses Mac address is insecure. In other words, the address resolution protocol (ARP) in the Ethernet can find the Mac address from the corresponding IP address.

Tseng proposed an identity-based key exchange protocol in 2007. The proposed protocol provides implicit key authentication and is based on the difficulty of computing a discrete logarithm problem. Because the protocol is based on the difficulty of computing a discrete logarithm problem, there are a lot of exponentiation computations to execute, thus the computing burden is concerned. Also, in 2008, Tseng *et al.* proposed another identity-based remote user authentication scheme using bilinear pairings. Based on the computational Diffie–Hellman assumption, the protocol is built up. The computation cost and the limited computing ability of smart card on the client end must be put into consideration. In 2009, Yoon *et al.* proposed a protocol which uses the one-way hash function

to protect the required tokens and proceeds authentication between a client's smart card and a pre-set set-top box, which is connected with a digital television broadcasting service provider. Both of them generate random numbers to compute specific tokens. The protocol based on one-way hash function and Diffie–Hellman key exchange algorithm, and the protocol was demonstrated able to provide perfect forward secrecy.

Putting aside the computing burden of Diffie–Hellman algorithm and isolating the fraud of the site authentication, we, therefore, propose a nonce-based authentication scheme using nonces instead of Mac address and accompanying with token updates to improve the functionality.

The rest of the paper is organized as follows: the related schemes are reviewed in Section 2 and we propose an improved scheme in Section 3. In Section 4, the security and performance of our scheme are discussed. Finally, the conclusions are given in the last section.

2. Reviews of Related Schemes

In this section, we review Lee *et al.*'s (2005a) and Shi *et al.*'s schemes.

2.1. *Review of Lee, Jang and Yoo's Scheme*

In 2005, Lee, Jang and Yoo added the site authentication by using nonce-based scheme (Lee *et al.*, 2005a). The scheme includes the registration phase, login and authentication phase.

In registration phase, a client uses his identity (id), password (pw), and media address (ma), to construct $token1 = (id \parallel pw \parallel ma)$ and $h(token1)$, where "$\parallel$" denotes the concatenation operation and h is a hashing function. Then, the user transmits $h(token1)$ and encrypted $E_{ks}(token1)$ to server. After receiving the message from user, the server generates a random nonce, rs, and computes $h(rs)$. Then, the server encrypts $h(rs)$ and sends back user the encrypted $E_{ks}(h(rs))$. At the same time, the server computes $token2 = token1 \parallel h(rs)$ and stores it in the database. The user decrypts $E_{ks}(h(rs))$ to obtain $h(rs)$ and construct a new token.

In login and authentication phase, the user constructs a new token, $token1 \parallel h(rs)$ and sends $h(token1 \parallel h(rs))$ to login the server. The data includes $h(rs)$ to ensure it is fresh. On receiving the login request from the user, the server checks whether $h(rs)$ is in the message to ensure the client's legitimacy and verifies if $h(token2)$ and received $h(token1 \parallel h(rs))$ are equal. If it is true, the user's login is accepted. At the stage, the server generates another nonce, rs^*, to update $token2^* = token1 \parallel h(rs^*)$ and sends $E_{ks}(h(rs^*))$ to user. Next time, the user will use the new nonce $h(rs^*)$ to login the server.

In the login and authentication phase of Lee, J. *et al.*'s scheme, there is no information to inform the user that the datum, $h(rs^*)$, is fresh and is transmitted from the server. The authentication between the legitimate user and the remote server is never implemented. Therefore, an attacker can apply replay attack to personate a legal client. The remote server will respond the attacker with a new nonce and update the token simultaneously.

The scenario will cause the legal client unable to login the server again, the connection between the user and the remote server is collapsed. There is another weakness as well. The server directly stores $h(token2)$ for verifying the user's legitimacy next time. Once an attacker intrudes the server, he can steal the datum and login the server. Moreover, if the datum is changed, the legal user can not login the server again. Thus, the way of protecting datum, just using hash function and saving it directly, is not secure enough.

2.2. *Review of Shi and Yoo's Scheme*

Shi and Yoo (2006) proposed another site-authenticated scheme. For security consideration, the scheme uses exclusive-OR operations and hash functions. The scheme is composed of three phases: the registration phase, the login phase and the authentication phase.

In registration phase, a user uses his identity (id), password (pw) and Mac address (ma) to construct $token1 = id \parallel pw \parallel ma$. Then the user encrypts $token1$ with server's public key, ks, and sends $M1 = h(token1) \oplus token1$ and encrypted $M2 = E_{ks}(token1)$ to server for registration request. On receiving the message from user, the server decrypts $M2$ to get $token1$ first, then computes $h(token1)$ and checks if the decrypted $token1$ is equal to $M1 \oplus h(token1)$. The server accepts the user's registration request if they are equal. Next, the server generates a random number, rs, to construct $M3 = h(token1) \oplus rs$ and $h(rs)$ and transmits them to the user. At this stage, the server stores $h(token1) \oplus MA$ and $rs \oplus MA$ to protect data, where MA is the Mac address of server.

In the login phase, the user extracts $rs = M3 \oplus h(token1)$ and constructs $M4 = h^2(token1) \oplus rs$. Then he sends his id and $M4$ to login the server.

In the authentication phase, the server uses rs to verify the user's legitimacy and generates a new random number, rs^*. The server also computes $token2 = token1 \parallel (rs - 1)$ and $h(token2)$ to construct $M5 = h(token2) \oplus rs^*$ and $M6 = h(token1) \oplus rs^*$. Then the server sends a message including $M5$ and $M6$ to user. On receiving $M5$ and $M6$, the user computes $h(token2) \oplus M5$ and $h(token1) \oplus M6$ to see if they are equal to authenticate server. After authenticating the server, the user constructs $token3 = token2 \oplus rs^*$ and computes $h(token3)$ to build $M7 = h(token3) \oplus (rs - 2)$. Then $M7$ is sent to the server to verify the user. The user stores $h(token1) \oplus rs^*$ and $h(token1) \oplus h(token3)$ for next login.

In Shi and Yoo's scheme, the server applies exclusive-OR operation on $h(token1)$ and rs respectively with Mac address to protect data before the data were stored. Actually, it is not secure either. Each LAN card at a network node possesses an exclusive Mac address identifier. When a data packet is transmitted through network, the LAN card identifies the destination according the destination address in the transmitting packet. For instance, in the Open Systems Interconnection (OSI) model, the network layer in a transmitted packet has to use its address (probable IP address) to verify the device at destination. Also, the data-link layer uses its address (for example, the Ethernet Mac address) to verify the destination device. When a network layer is packed into a data-link layer, the system needs to obtain the destination's Mac address. Taking IP as an example,

the system consigns the task to Address Resolution Protocol (ARP). In other words, once the IP address is known, the corresponding Ethernet Mac address can be obtained through ARP. Therefore, an attacker can easily obtain the server's Mac address by checking the IP address of destination in the ARP's table if he intercepts the packet in the connection. So, trying to use the network physical address, Mac address, to protect token is not secure.

3. Our Proposed Scheme

For the purpose of security and improving the weakness of the reviewed schemes, we propose a securer nonce-based authentication scheme with token-update. The proposed scheme is composed of three phases: the registration phase, the login phase and the authentication phase.

3.1. *The Registration Phase*

When a user, U_i, intends to create and register an account at a server, S, he executes the following three steps.

Step 1. The user generates a nonce, computes and transmits message, $M_1 = \{V_1, V_2\}$, to the server, that is, $U_i \longrightarrow S : M_1 = \{V_1, V_2\}$.

The user generates a nonce, N_u, and concatenates his own identity number (ID_i) and password (PW_i) to compute a token, i.e., $Tkn_1 = (ID_i \parallel PW_i) \oplus N_u$ and a value, $PN = PW_i \oplus N_u$, where "$\parallel$" denotes the concatenation operation and "$\oplus$" denotes the exclusive-OR operation. Next, he constructs $V_1 = h(Tkn_1) \oplus Tkn_1$ and obtains $V_2 = E_{ke}(Tkn_1)$ by encrypting Tkn_1 with server's public key, ke . Then, the user sends the message, $M_1 = \{V_1, V_2\}$, to the server which he is intending to register. The value, PN, is kept secretly and is used to protect the nonce, N_u.

Step 2. The server verifies the user's registration and replies a message, $M_2 = \{V_3, h(N_s)\}$, to the applicant, that is, $S \longrightarrow U_i : M_2 = \{V_3, h(N_s)\}$.

After receiving the message, M_1, the server decrypts V_2 with his private key, kd, to have $Tkn_1 = D_{kd}(V_2)$. Next, the server computes $V_1 \oplus Tkn_1$ to obtain $h(Tkn_1)^*$ and checks whether $h(Tkn_1)$ and $h(Tkn_1)^*$ are equal. When it is true, the user's registration is granted. At this moment, the server generates a nonce, N_s, and computes $V_3 = h(Tkn_1) \oplus N_s$ and $h(N_s)$. Then, the server transmits the message $M_2 = \{V_3, h(N_s)\}$ to U_i and stores $h(Tkn_1) \oplus N_s$ in its own database.

Step 3. The user gets the nonce, N_s, and confirms his registration.

On receiving the message, $M_2 = \{V_3, h(N_s)\}$, from server, the user operates exclusive-OR on $h(Tkn_1)$ and V_3 to get a nonce, N_s', i.e., $N_s' = V_3 \oplus h(Tkn_1)$. After that, he checks whether $h(N_s')$ and $h(N_s)$ are the same. Once they are equal, the user confirms that his registration at the prospective server is successful.

3.2. *The Login Phase*

A user offers a login request, $M_3 = \{ID_i, V_4\}$, to the server, that is, $U_i \longrightarrow S : M_3 = \{ID_i, V_4\}$.

Suppose a user, U_i, intends to login the server, which he has registered before. The user at first retrieves N_u by implementing the exclusive-OR operation on PW_i and PN, that is, $N_u = PN \oplus PW_i$. He next uses N_u and his ID_i, PW_i to construct the token, $Tkn_1 = (ID_i \parallel PW_i) \oplus N_u$, and compute $h(Tkn_1)$. The user executes the exclusive-OR operation on V_3 and $h(Tkn_1)$ to extract $N_s = V_3 \oplus h(Tkn_1)$. Then, $V_4 = h^2(Tkn_1) \oplus N_s$ is computed and the message, $M_3 = \{ID_i, V_4\}$, is transmitted to the server for login request.

3.3. *The Authentication Phase*

In this phase, mutual authentication between the user and the server is inevitable. The following steps are implemented.

Step 1. The server transmits message, $M_4 = \{V_5, V_6\}$, to the user, that is, $S \longrightarrow U_i : M_4 = \{V_5, V_6\}$.

After receiving the login request from user, the server instantly checks whether N_s and $h^2(Tkn_1) \oplus V4$ are the same. If they are equal, the server generates a fresh nonce, N_s^*. It follows that the server computes another token, $Tkn_2 = Tkn_1 \parallel (N_s - 1)$, and $h(Tkn_2)$. Also, the server uses this new nonce, N_s^*, to compute $V_5 = h(Tkn_2) \oplus N_s^*$ and $V_6 = h(Tkn_1) \oplus N_s^*$. After all those computations are done, the server transmits message, $M_4 = \{V_5, V_6\}$, to the user.

Step 2. The user transmits message, $M_5 = \{V_7\}$, to server, that is, $U_i \longrightarrow S : M_5 = \{V_7\}$.

On receiving the message from server, the user computes $Tkn_2 = Tkn_1 \parallel (N_s - 1)$ and $h(Tkn_2)$ immediately. Then, he check whether $h(Tkn_2) \oplus V_5$ and $h(Tkn_1) \oplus V_6$ are equal or not. The user, U_i, authenticates the server, if they are definitely the same. Next, the user constructs a new token, $Tkn_3 = Tkn_2 \parallel N_s^*$. With the new token, the user computes $h(Tkn_3)$ and $V_7 = h(Tkn_3) \oplus (N_s - 2)$. Then, the user transmits message, $M_5 = \{V_7\}$, to server. At this moment, the user stores the data, $h(Tkn_1) \oplus N_s^*$ and $h(Tkn_1) \oplus h(Tkn_3)$. Those data will be used to construct a new token for next login. Actually, the user will transmit the message, $\{ID_i, h^2(Tkn_3) \oplus N_s^*\}$, to the server for next login request. The login token, $h^2(Tkn_1) \oplus N_s$, will be updated to $h^2(Tkn_3) \oplus N_s^*$.

Step 3. The server authenticates the user.

On receiving the message, M_5, the server computes $Tkn_3 = Tkn_2 \parallel N_s^*$ and $V_7^* = h(Tkn_3) \oplus (N_s - 2)$ to check whether the received V_7 is equal to V_7^*. If they are equal, the server authenticates the user.

The proposed nonce-based authentication scheme is depicted in Fig. 1.

4. The Security and the Performance of Our Scheme

We discard the use of timestamp, thus, there is no time synchronization problem in our scheme. We mostly adopt hashing functions and exclusive-OR operations. Not only is the computation cost negligible, also the performance is very easy and efficient. The hashing

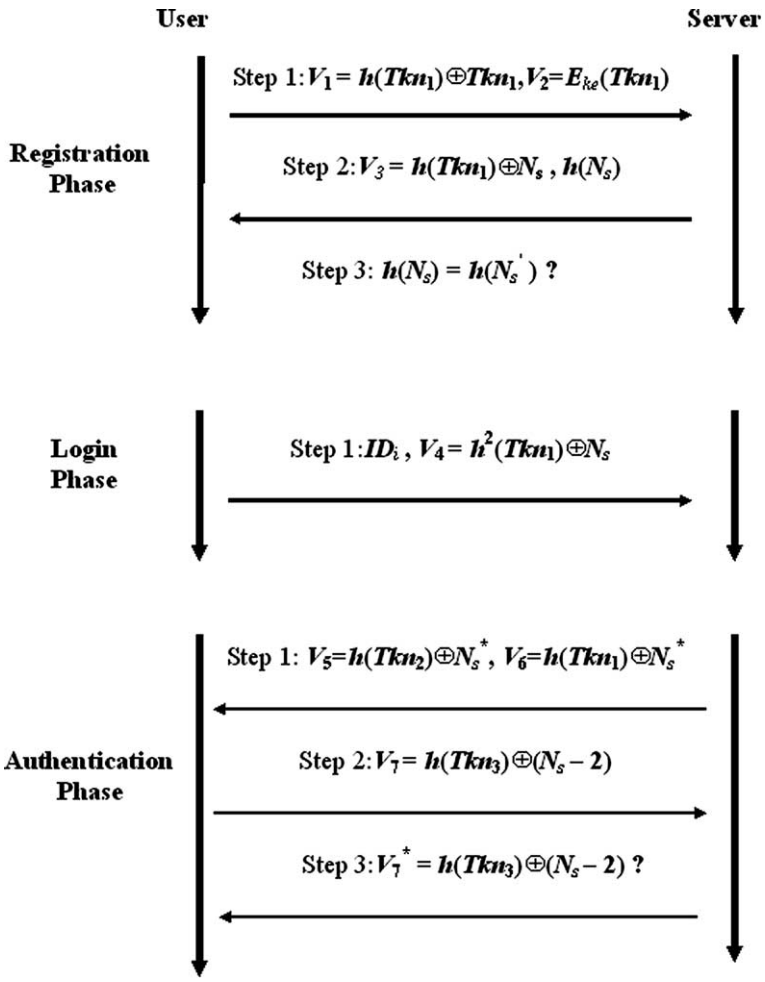

Fig. 1. The proposed nonce-based authentication scheme.

function and exclusive-OR operation also provide the necessary and reliable security and functionality of protection.

At the registration phase of our proposed scheme, to construct a token, Tkn_1, we at first concatenate the identity, ID_i, and the password, PW_i, then exploit the exclusive-OR operation on the concatenated string, $ID_i \parallel PW_i$, and a nonce, N_u. The construction using nonce instead of Mac address can prevent the personal information, which is included in the token, from being intercepted or eavesdropped on transmitting. No one can construct the token without this given nonce as well. In this phase, the applicant also dispatches the encrypted token, $E_{ke}(Tkn_1)$, to the prospective server. An interceptor can not procure the token, either, because he does not possess the server's private key. In the scenario, the security of this token can be guaranteed. In our scheme, the server verifies the applicant before the registration is granted and an authenticating nonce, N_s,

is dispatched. Both the server and the user indirectly store the sensitive data. The data are stored under the protection through the exclusive-OR operation. The user stores the nonce, N_u, with PN, since $PN = PW_i \oplus N_u$; and the server keeps secretly the nonce, N_s, by the form $h(Tkn_1) \oplus N_s$. Hence, they are secure and not able to be intercepted or eavesdropped.

On the other hand, the extent of a password concerns the internet security. For the sake of security of internet, the most of websites instruct users with guidance in relation to selecting passwords and the extent of coverage. The extent of password is recommended at least 5–6 characters to 10–16 characters (Furnell, 2007). According to the instructions, the extent of password in our scheme is supposed to be at least 5–6 characters or even more. Thus the extent of string, $ID_i \parallel PW_i$, will be much more. To an attacker, the string will be much harder to guess. Aside hard to guess the string, the nonce, N_u, is randomly generated when it is required on the user's end; it is always fresh, indeed. Therefore, the token, $Tkn_1 = (ID_i \parallel PW_i) \oplus N_u$ is definitely secure and can resist the password-guess attack.

When the user intends to login server, he retrieves the typical nonce to construct Tkn_1 and computes the login token, $V_4 = h^2(Tkn_1) \oplus N_s$. Those data are never exposed directly. No one can masquerade the user for trying to login the remote server.

On receiving the login request from the user, the server verifies the user's legitimacy using nonce, N_s, which is dispatched by the server. Only the legitimate user can obtain this nonce. The nonce is protected with $h(Tkn_1)$ thru exclusive-OR operation. Without $h(Tkn_1)$, which can only be constructed by the user himself, no one can retrieve the nonce, N_s. Thus, no one can personate the user. In other words, our scheme can resist the replay attack. Once authenticating the user, the server generates a new nonce, N_s^*. To protect this new nonce, the server constructs new data and sends them to the user. The user computes new verifying information and transmits it to the server. The information will be used to authenticate the user. The mutual authentication between the server and the user can prevent the communication from the man-in-the-middle attack.

In the schemes of Lee *et al.* (2005a) and Shi *et al.* (2006), they apply the user's identity, id, password, pw, and Mac address, ma, to generate a token, $token1 = (id \parallel pw \parallel ma)$. In our scheme, we construct the token with ID_i, PW_i and N_u. At first, our scheme concatenates the identity, ID_i, and password, PW_i, then applies exclusive-OR with a nonce, N_u. However, the physical address of Mac can easily be detected in the Ethernet from the corresponding IP address through the address resolution protocol, ARP, and is vulnerable to malicious attacks. So, we adopt a nonce, N_u, instead of media access address, ma, in our proposed scheme and incorporate with token updates. Hence, our scheme does improve the functionality and should be invulnerable. In Shi and Yoo's scheme (Shi *et al.*, 2006), the random number, rs, may be revealed from the protected value, $rs \oplus MA$, where MA is the Mac address of the server, once the parameter, MA, is detected. Then the password, pw, and the token, $token1$, will be evaluated immediately. Therefore these schemes suffer from the Mac address attack. They are not secure. That is why we propose an improved securer and efficient scheme based on the nonce variable instead of using Mac address. Our proposed scheme resists against the Mac address

Table 1
Comparison of security on resisting attacks

	Replay	Masquerading as user	Impersonating as server	Man-in-the-midle
Lee *et al.* (2005a)	No	No	No	No
Shi *et al.* (2006)	Yes	Yes	Yes	Yes
Our scheme	Yes	Yes	Yes	Yes

Table 2
Comparison on performance

	Encryption	Verification table	Use Mac address	Mutual authentication
Lee *et al.* (2005a)	Yes	No	Yes	No
Shi *et al.* (2006)	Yes	Yes	Yes	Yes
Our scheme	Yes	Yes	No	Yes

attack. The comparison with the related reviewed schemes on security properties of our scheme is summarized in Table 1 and that on performance is listed in Table 2.

5. Conclusions

In this paper, we propose an improved securer and efficient nonce-based authentication scheme, in which hashing functions and exclusive-OR operations are mostly exploited. The advantage of using hashing functions and exclusive-OR operations is ease, convenience, rapidity and low-cost. According to the analyses on the security and the performance, our scheme is better than the schemes of Lee *et al.* (2005a) and Shi *et al.* (2006), because we discard the use of Mac address, which can be read through Address Resolution Protocol. Furthermore, the proposed scheme can prevent the replay attack, man-in-the-middle attack and Mac address attack. Our scheme not only reduces the computation cost also keeps our most concerns on the security.

References

Chen, T.-H., Lee, W.-B., Horng, G. (2004). Secure SAS-like password authentication schemes. *Computer Standards & Interfaces*, 27, 25–31.

Chen, Y.-C., Yeh, L.-Y. (2005). An efficient nonced-based authentication scheme with key agreement. *Applied Mathematics and Computation*, 169, 982–994.

Chien, H.-Y., Chan, J.-K. (2003). Robust and simple authentication protocol. *Computer Journal*, 46, 193–201.

Furnell, S. (2007). An assessment of website password practices. *Computers & Security*, 26, 445–451.

Goyal, V., Kumar, V., Singh, M., Abraham, A., Sanyal, S. (2006). A new protocol to counter online dictionary attacks. *Computers & Security*, 25(2), 114–120.

Halevi, S., Krawczyk, H. (1998). Public-key cryptography and password protocols. In: *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San-Francisco, CA, pp. 122–131.
Hwang, M.-S., Lee, C.-C., Tang, Y.-L. (2002). A simple remote user authentication scheme. *Mathematical and Computer Modeling*, 36, 102–107.
Jiang, R., Pan, L., Li, J.-H. (2004). Further analysis of password authentication schemes based on authentication tests. *Computers & Security*, 23, 469–477.
Juang, W.-S. (2004). Efficient password authenticated key agreement using smart card. *Computer & Security*, 23(2), 167–173.
Lee, J., Jang, I., Yoo, H.-S. (2005a). Modified token-update scheme for site authentication. *LNCS*, Vol. 3481, 111–116.
Lee, S.-W., Kim, H.-S., Yoo, K.-Y. (2005b). Efficient nonce-based remote user authentication scheme using smart cards. *Applied Mathematics and Computation*, 167, 355–361.
Liao, C.-H., Wang, C.-T., Chen, H.-C. (2009). An exquisite mutual authentication scheme with key agreement using smart card. *Informatica*, 33, 117–124.
Liaw, H.-T., Lin, J.-F., Wu, W.-C. (2006). An efficient and complete remote user authentication scheme using smart card. *Mathematical and Computer Modelling*, 44, 223–228.
Lin, C.-L., Sun, H.-M., Hwang, T. (2001). Attacks and solutions on strong-password authentication. *IEICE Trans. Commun.*, E84-B, 9, 2622–2627.
Phan, Raphael C.-W. (2006). Cryptanalysis of two password-based authentication schemes using smart cards. *Computers & Security*, 25(1), 52–54.
Shi, W., Yoo, H.S. (2006). Efficient nonce-based authentication scheme using token-update. In: *ICCSA* 2006, *LNCS*, Vol. 3982, 213–221.
Shieh, W.-G., Wang, J.-M. (2006). Efficient remote mutual authentication and key agreement. *Computers & Security*, 25(1), 72–77.
Tseng, Y.-M. (2007). An efficient two-party identity-based key exchange protocol. *Informatica*, 18(1), 125–136.
Tseng, Y.-M., Wu, T.-Y., Wu, J.-D. (2008). A pairing-based user authentication scheme for wireless clients with smart cards. *Informatica*, 19(2), 285–302.
Yang, C.-C., Wang, R.-C., Chang, T.-Y. (2005). An improvement of the Yang–Shieh password authentication schemes. *Applied Mathematics and Computation*, 162, 1391–1396.
Yang, W.-H., Shieh, S.-P. (1999). Password authentication schemes with smart cards. *Computers & Security*, 18(8), 727–733.
Yoon, E.-J., Ryu, E.-K., Yoo, K.-Y. (2005). An improvement of Hwang–Lee–Tang's simple remote authentication scheme. *Computers & Security*, 24(1), 50–56.
Yoon, E.-J., Yoo, K.-Y. (2009). Robust key exchange protocol between set-top box and smartcard in dtv broadcasting. *Informatica*, 20(1), 139–150.

C.-H. Liao received his BS degree in mathematics at National Kao-Hsiung Normal University in 1971 and the MS degree in mathematics at Pittsburg State University in Kansas, United States, in 1997. He has joined the faculty of National Chin-Yi University of Technology, Taichung, Taiwan, since 1981. Currently, he is a senior instructor of General Education Center at National Chin-Yi University of Technology. Recently, his research interests include information security and cryptography.

C.-T. Wang was born in Taichung, Taiwan, Republic of China, on October 28, 1956. He received his BS degree in mathematics from Tunghai University in 1980, his MS degree in applied mathematics from National Chung Hsing University in 1987, and PhD degree in computer science and information engineering from National Chung Cheng University in 1999. During the academic years 1987–2000, he was on the Faculty of General Education Center at National Chin-Yi University of Technology, Taichung, Taiwan. From August 2000 to July 2002, he was also the director of Computer Center of the college.

Currently, he is a professor of Department of Information Management at National Chin-Yi University of Technology. His research interests include computer security, cryptography and computer algorithms.

H.-C. Chen received the BS, MS, and PhD degrees in information management from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1992, 1994, and 1998, respectively. After the two year military service, he was an assistant professor in National Kaohsiung Institute of Marine Technology, Kaohsiung, Taiwan, from 2000 to 2002. Since 2002, he has been an assistant professor in the Department of Information Management, National Chin-Yi University of Technology, Taichung, Taiwan. His research interests include algorithms, parallel processing, and graph theory.

Saugesnė ir efektyvi laikinos autentifikacijos su žymės atnaujinimu schema

Chiu-Hsiung LIAO, Ching-Te WANG, Hon-Chan CHEN

Straipsnyje pasiūlyta tarpusavio autentifikavimo schema, kurioje vietoje Mac adreso naudojamas laikinas kintamasis, o funkcionalumas pagerinamas vykdant žymės (token) atnaujinimą. Lee ir kt. (2005) ir Shi ir kt. (2006) pasiūlė vietos autentifikavimo schemoje naudoti atsitiktinius skaičius. Ši schema gali identifikuoti personalinį kompiuterį naudodama LAN kortos Mac adresą, tačiau Mac adresas yra lengvai nustatomas iš atviros sistemos jungties modelio adresų protokolo. Pasiūlyta saugesnė bei efektyvi autentifikavimo schema, atliekanti tarpusavio autentifikavimą, kuri yra atspari atsako (reply), piktavalis viduryje (man-in-the-middle) ir Mac adreso atakoms.