International Journal of Innovative Computing, Information and Control
Volume **9**, Number **1**, January **2013**
ICIC International ©2013 ISSN 1349-4198
pp. 33–45

DYNAMIC PASSWORD BASED REMOTE USER AUTHENTICATION WITHOUT TIME STAMPING

Mohamed Hamdy Eldefrawy[1,*], Muhammad Khurram Khan[1]
and Khaled Alghathbar[1,2]

[1]Center of Excellence in Information Assurance
[2]Information Systems Department
College of Computer and Information Sciences
King Saud University
P.O. Box 92144, Riyadh 11653, Saudi Arabia
[*]Corresponding author: meldefrawy@ksu.edu.sa; { mkhurram; kalghathbar }@ksu.edu.sa

Received August 2011; revised December 2011

Abstract. *The rapid growth of electronic commerce and Internet-based applications demands strong privacy protection and robust system security, which are essential requirements for an authentication scheme or access control mechanism. The issue of remote user authentication has received much attention recently. Password authentication is one of the most convenient authentication mechanisms. Most of the presented remote user authentication schemes utilize time stamping in the authentication process, which requires accurate time synchronization. Because it would be very beneficial to overcome the insufficiencies of such schemes regarding the need for accurate time synchronization, in this paper, we present a new algorithm that uses two different types of hash functions, which come with a nested hashing chain and utilize the discreet logarithm problem in message exchange. We first review the existing solutions and emphasize on their shortcomings and drawbacks. We then propose a secure and reliable remote user authentication scheme without the need of time stamping which covers the vulnerabilities of existing solution. A detailed security analysis was also performed that covered types of attacks that could influence our scheme. The proposed scheme establishes a common session key that provides message confidentiality.*

Keywords: Dynamic password, Nested hashing, Remote authentication, Smart card

1. Introduction.

Since Lamport [1] proposed the first password-based remote authentication scheme to identify a legal user within an insecure communication environment, a series of relevant studies and authentication mechanisms has been investigated. However, most of the previously published schemes cannot achieve computation efficiency and system security in parallel. In password-based authentication schemes with smart cards, remote users are authenticated using their smart card as an identification token; the smart card takes an input password from a user, calculates unique parameters from the user-given password, creates a login message using these parameters, and then sends a login message to the server, which then checks the validity of the login request before allowing access to any services or resources. In this way, the administrative overhead of the server is greatly reduced and a remote user is only required to remember his password to log on. In addition to just creating and sending login messages, smart cards support mutual authentication, where a challenge-response communication between the card and server takes place so that each can verify the other's identity. Mutual authentication is a vital necessity in most real-world applications, where one's private information should not be released to anyone until mutual confidence is established. Thus, at this stage, we are concerned with mutual authentication and secure session generation. From a security point of view, it is better to consider these topics in cooperation rather than disjointedly. A protocol providing authentication without key exchange is susceptible to an enemy who waits until the authentication is complete and then takes over one end of the communication line. Such an attack is not precluded by a key exchange that is independent of authentication. Key exchange should be linked to mutual authentication so that a party has assurances that an exchanged key (which might be used to facilitate privacy or integrity and thus keep authenticity alive) is in fact shared with the authenticated party, and not an impostor. For these reasons, it is essential to keep key exchange mutual authentication in mind in the design and analysis of authentication protocols.

In this paper, we propose a remote user authentication scheme without time-stamping. The proposed scheme uses dynamic password authentication that is valid only one time to provide improved protection. Our scheme provides a practical remote user authentication scheme while retaining all advantages of robust remote user authentication scheme without the need of time synchronization. Our scheme is featured as following: (1) It achieves mutual authentication and session key generation; (2) It prevents from the security attacks due to the disclosure of the secrets stored in the storage device; (3) No verification tables are required; (4) No time stamping is required and consequentially no time synchronization is required; (5) The public key of each communication party is authenticated without the need of key certificate; (6) The private key of each party cannot be disclosed by any other third party.

1.1. **Related work.** In 1981, Lamport [1] proposed the first well-known remote password authentication scheme using smart cards. In Lamport's scheme, the Authentication Server (AS) stores a password table at the server to check the validity of the login request made by the user. However, a high hash overhead and the need to reset passwords decrease the suitability and practical ability of Lamport's scheme. In addition, Lamport's scheme is vulnerable to a small n attack [2]. Since then, many similar schemes [3,4] have been proposed. These all have a common feature: a verification password table should be securely stored in the AS. Actually, this property is a disadvantage from the security judgment. The AS will be partially or totally broken if an adversary steals, removes, or modifies the password table. Hwang and Li [5] pointed out that Lamport's scheme secures from the risk of a modified password table and the cost of protecting and maintaining the password table. Further, they proposed a new remote user authentication scheme using smart cards. Hwang and Li's [5] scheme does not maintain the password table at the server to check the validity of the login request. In addition, it can resist a message-replaying attack [6]. In the year 2000, Chan and Cheng [7] presented the security flaws of Hwang-Li's scheme [5]. Afterwards Shen, Lin and Hwang [8] discussed a different type of attack on Hwang-Li's scheme and also offered an enhanced scheme to solve the security problem of Hwang-Li's scheme. Also, Chang and Hwang [9] clarified the practical problems of Chan-Cheng's attack [7] on Hwang-Li's scheme and Leung, Cheng, Fong, and Chen [10] mentioned that Shen, Lin and Hwang's scheme [8] is still vulnerable to the attack proposed by Chan and Cheng [7]. Awasthi and Lal [11] also mentioned a different type of attack on Hwang-Li's scheme and introduced a remote user authentication scheme. Awasthi and Lal [11] stated that their scheme provides forward secrecy to the AS. In the year 2004, Kumar [12] analyzed the practical drawbacks of Awasthi and Lal's scheme [11]. In the same year, Lee et al. [13] raised a question about the correctness of Awasthi and Lal's scheme [11]. Lee et al. [13] also proved that Awasthi and Lal's scheme [11] is inaccurate and does not grant forward secrecy for the secret key of the authentication server, AS.

In 2004, Kumar [14] presented a new scheme to solve the security problems of Hwang-Li's scheme [5]. Kumar's scheme [14] removes the security vulnerabilities of Hwang-Li's scheme [5], but does not provide a complete solution to solve all of the possible problems and withstand all possible attacks.

1.2. **Organization.** The remainder of this paper is organized as follows. Section 2 gives the notations. Section 3 reviews the other schemes. Section 4 shows some security flaws in previous schemes. Section 5 presents the proposed scheme. Section 6 illustrates the practical implementation of the proposed scheme. The security of the proposed scheme is analyzed in Section 7. Section 8 discusses the proposed scheme's performance analysis and comparison. Finally, the conclusions are presented in Section 9.

2. **Notations.**

The notations used throughout this paper are illustrated in the following table:

Notation	Description
ID_i	Denotes an identity of a remote user U_i
PW_i	Denotes the corresponding password for a certain registered identity ID_i
SM_i	Denotes the corresponding smart card of user U_i
T	Current time stamp
ΔT	Valid time interval
AS	Denotes the authentication server
p	Large prime (usually at least 1024 bits)
q	Prime (typically of 160 bits) with $q\| (p-1)$
G	G is a subgroup of $\mathbb{Z}_p^*$ and is often a subgroup of order q
g	Generator of G
$h(\cdot)$	Denotes a cryptographic one way hash function
$h_A(\cdot)$	Represents the first hash function
$h_B(\cdot)$	Represents the second hash function
s_{int}	The OTP initial seed
s_t	The OTP seed number t for the tth authentication (current seed)
OTP_t	The OTP number t for the tth authentication
(x_t, y_t)	The nested hashing progress values for the tth authentication
$h_B^{y_t}(h_A^{x_t}(s_t))$	Hashing the seed number tth (s_t) by $h_A(\cdot)$ for x_t times followed by $h_B(\cdot)$ hashing for y_t times
$\|\|$	Denotes a concatenation operation

3. **Review of Authentication Schemes.**

This section reviews three authentication schemes, which are composed of four parts: the registration phase, login phase, verification phase, and password change phase. This section will only focus on the first three phases, which will be mainly used in their security analyses.

3.1. **Wang-Li's scheme.** If user U_i wants to log in to an AS [15], he or she must insert his or her smart card into a card reader and key in his or her identifier, ID_i, and password, PW_i. Then, the smart card generates a random number, $r \in \mathbb{Z}_q^*$; computes $R_i = h(ID_i||x_s), X_i = R_i \oplus h(ID_i||PW_i)$, where x_s is held by the authentication server. It computes $t = g^r \bmod p$ and $V_i = X_i \oplus h(ID_i||PW_i)$. Then the smart card computes $W_i = h(V_i \oplus T)$, where T is the current time-stamp; computes $s = h(t||W_i)$; and sends a message, $C_1 = \{ID_i, t, s, T\}$, to the AS. Upon receiving the authentication request message, C_1, the remote system and smart card perform the following steps for mutual authentication between the user and remote system. Let T' be the time that the system receives C_1. The AS compares T and T'. If the difference between T and T' is within a valid time interval, ΔT, C_1 is considered to be a valid message. The system computes $V_i' = h\left(ID_i||x_s\right)$, as well as $W_i' = h\left(V_i' \oplus T\right)$. The system compares $h\left(t||W_i'\right)$ with s. If they are equal, then the system validates the login call and proceeds to the next step; otherwise, it rejects the login request. The system picks a random number, $\bar{r} \in \mathbb{Z}_q^*$, and computes the session key, $k = t^{\bar{r}} \bmod p$. The system acquires the current time-stamp, T'', and computes $w = h\left(V_i' \oplus T''\right)$, $u = g^{\bar{r}} \bmod p$, $v = h\left(u||w\right)$. The system sends back the message $C_2 = \{u, v, T''\}$ to U_i. Upon receiving the message $\{u, v, T''\}$, the smart card verifies the validity of the time interval between T'' and the current time-stamp, T''', and then computes $W' = h\left(V_i \oplus T''\right)$. If $v = h\left(u||w'\right)$, the mutual authentication is complete. Then, $k = g^{\bar{r}r} \bmod p$ is used as the session key between the user, U_i, and the remote system. Figure 1 present Wang-Li's Scheme.

3.2. **Wang-Liu-Xiao-Dan's scheme.** In the user registration phase, U_i submits ID_i to AS via a secure channel. Then, AS computes $N_i = h\left(PW_i\right) \oplus h\left(x\right) \oplus ID_i$, where PW_i is chosen by AS, and personalizes a smart card, SM_i stored in the parameters $\{h(.), N_i, y\}$. AS sends SM_i and PW_i to U_i via a secure channel. Figure 2 introduces the login and

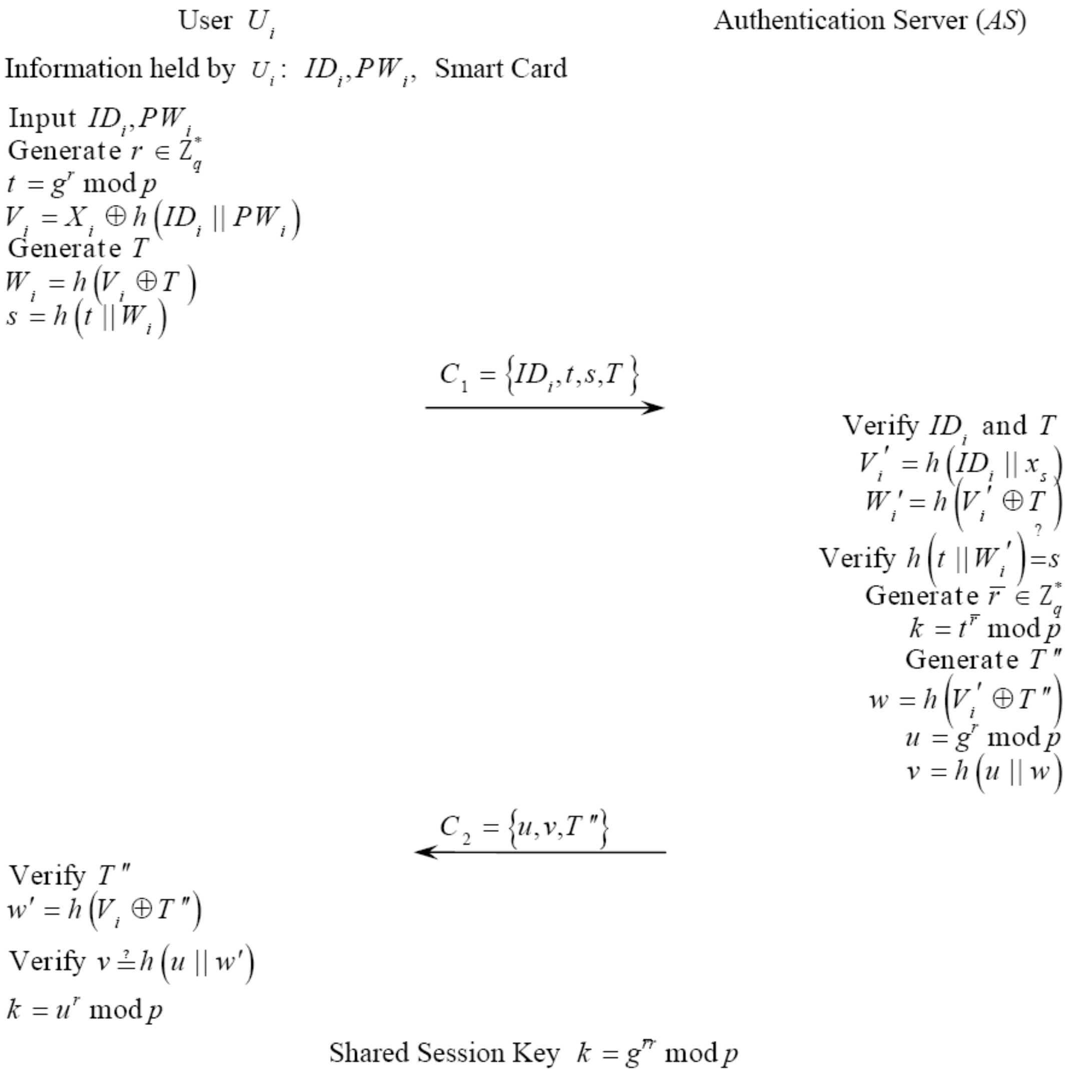

User U_i
Authentication Server (AS)
Information held by U_i: ID_i, PW_i, Smart Card
Input ID_i, PW_i
Generate $r \in Z_q^*$
$t = g^r \bmod p$
$V_i = X_i \oplus h\left(ID_i || PW_i\right)$
Generate T
$W_i = h\left(V_i \oplus T\right)$
$s = h\left(t || W_i\right)$
$C_1 = \{ID_i, t, s, T\}$
Verify ID_i and T
$V_i' = h\left(ID_i || x_s\right)$
$W_i' = h\left(V_i' \oplus T\right)$
Verify $h\left(t || W_i'\right) \overset{?}{=} s$
Generate $\bar{r} \in Z_q^*$
$k = t^{\bar{r}} \bmod p$
Generate T''
$w = h\left(V_i' \oplus T''\right)$
$u = g^r \bmod p$
$v = h\left(u || w\right)$
$C_2 = \{u, v, T''\}$
Verify T''
$w' = h\left(V_i \oplus T''\right)$
Verify $v \overset{?}{=} h\left(u || w'\right)$
$k = u^r \bmod p$
Shared Session Key $k = g^{\bar{r}r} \bmod p$

FIGURE 1. Wang-Li's user authentication scheme

verification phases of Wang-Liu-Xiao-Dan's scheme [16]. Wang et al. stated that their scheme could make the security and secrecy of the password stronger by using a scheme where a password is chosen by the server and transmitted to the user to reduce the possibility of weak-password-selection.

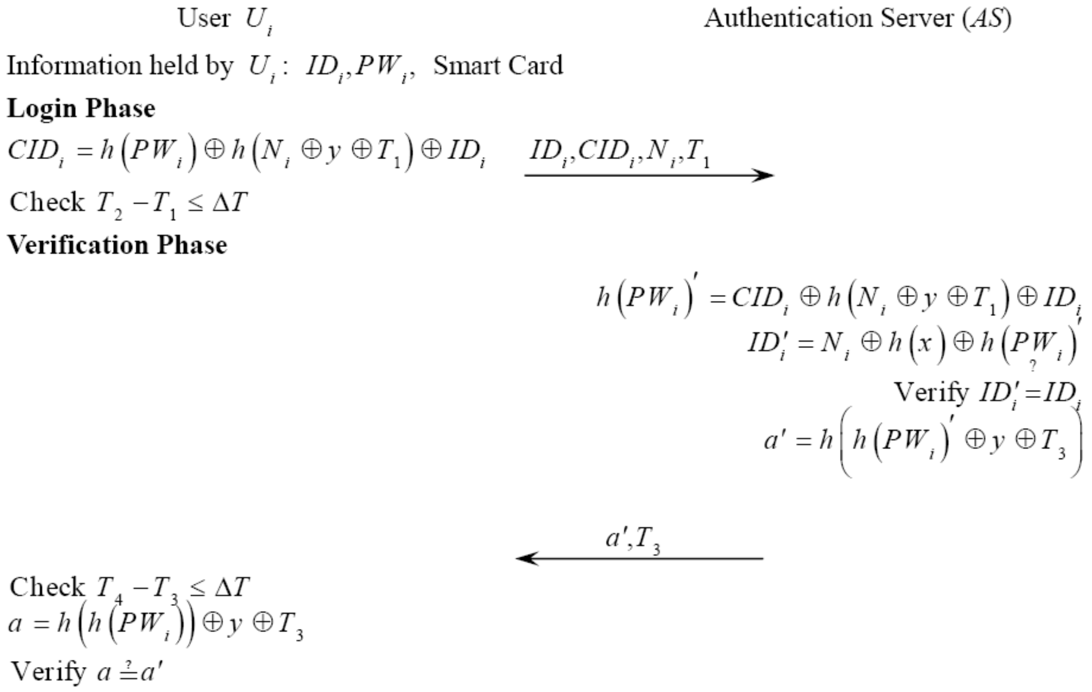

User U_i
Authentication Server (AS)
Information held by U_i: ID_i, PW_i, Smart Card
Login Phase
$CID_i = h(PW_i) \oplus h(N_i \oplus y \oplus T_1) \oplus ID_i$
ID_i, CID_i, N_i, T_1
Check $T_2 - T_1 \le \Delta T$
Verification Phase
$h(PW_i)' = CID_i \oplus h(N_i \oplus y \oplus T_1) \oplus ID_i$
$ID_i' = N_i \oplus h(x) \oplus h(PW_i)$
Verify $ID_i' \overset{?}{=} ID_i$
$a' = h\left(h(PW_i)' \oplus y \oplus T_3\right)$
a', T_3
Check $T_4 - T_3 \le \Delta T$
$a = h(h(PW_i)) \oplus y \oplus T_3$
Verify $a \overset{?}{=} a'$

Figure 2. Wang-Liu-Xiao-Dan's user authentication scheme

3.3. **Kim-Chung's scheme.** Kim-Chung's scheme also has the merit that the server does not need to maintain a verification table [17]. They quarreled that their scheme can offer some useful features, including the early detection of an incorrect password, prevention of password leak, and so on. In the user registration phase, U_i submits ID_i and PW_i to AS via a secure channel. Then, AS computes $K_1 = h(ID_i \oplus x) \oplus N$, where N is a random number unique to user U_i, $K_2 = h(ID_i \oplus x \oplus N) \oplus h(PW_i \oplus h(PW_i))$, and $R = K_1 \oplus h(PW_i)$ and personalizes a smart card, SM_i, stored in the secure parameters $\{K_1, K_2, R, h(\cdot)\}$ AS sends SM_i to U_i via a secure channel. Figure 3 presents the login and verification phases of Kim-Chung's scheme.

4. **Security Flaws in the Previous Authentication Schemes.** In this section, we are going to show that the previously presented schemes are vulnerable to some attacks. For their security analyses, we use the same assumption as Kim-Chung's scheme that all existing smart cards are vulnerable because the secret values stored in a smart card could be extracted by monitoring its power consumption.

4.1. **Reflection attack.** We consider the scenario of a reflection attack [18] on Wang-Li's scheme. In the login phase, if attacker U_a has intercepted and blocked a message transmitted, i.e., $C_1 = \{ID_i, t, s, T\}$, he or she can impersonate the remote system and send $C_2 = \{u, v, T''\}$ to U_i of the authentication phase, where $u = t$, $v = s$, and $T'' = T$ is the current timestamp. Upon receiving the first item of the received message, i.e., T'', U_i will compute $w' = h(h(V_i \oplus T''))$. U_i will be fooled into believing that the attacker is the legal remote system. Because U_i cannot actually authenticate the remote system's

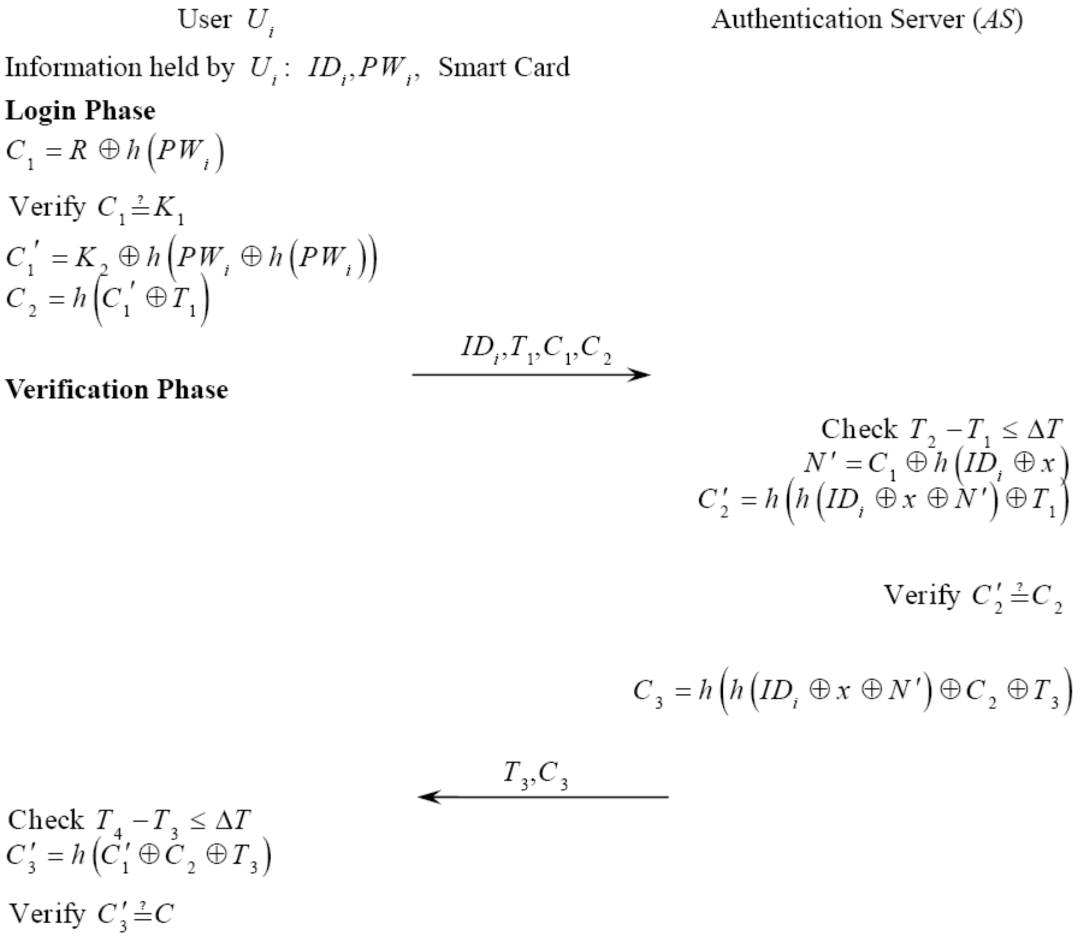

User U_i
Authentication Server (AS)
Information held by U_i: ID_i, PW_i, Smart Card
Login Phase
$C_1 = R \oplus h(PW_i)$
Verify $C_1 \overset{?}{=} K_1$
$C_1' = K_2 \oplus h(PW_i \oplus h(PW_i))$
$C_2 = h(C_1' \oplus T_1)$
ID_i, T_1, C_1, C_2
Verification Phase
Check $T_2 - T_1 \leq \Delta T$
$N' = C_1 \oplus h(ID_i \oplus x)$
$C_2' = h(h(ID_i \oplus x \oplus N') \oplus T_1)$
Verify $C_2' \overset{?}{=} C_2$
$C_3 = h(h(ID_i \oplus x \oplus N') \oplus C_2 \oplus T_3)$
T_3, C_3
Check $T_4 - T_3 \leq \Delta T$
$C_3' = h(C_1' \oplus C_2 \oplus T_3)$
Verify $C_3' \overset{?}{=} C$

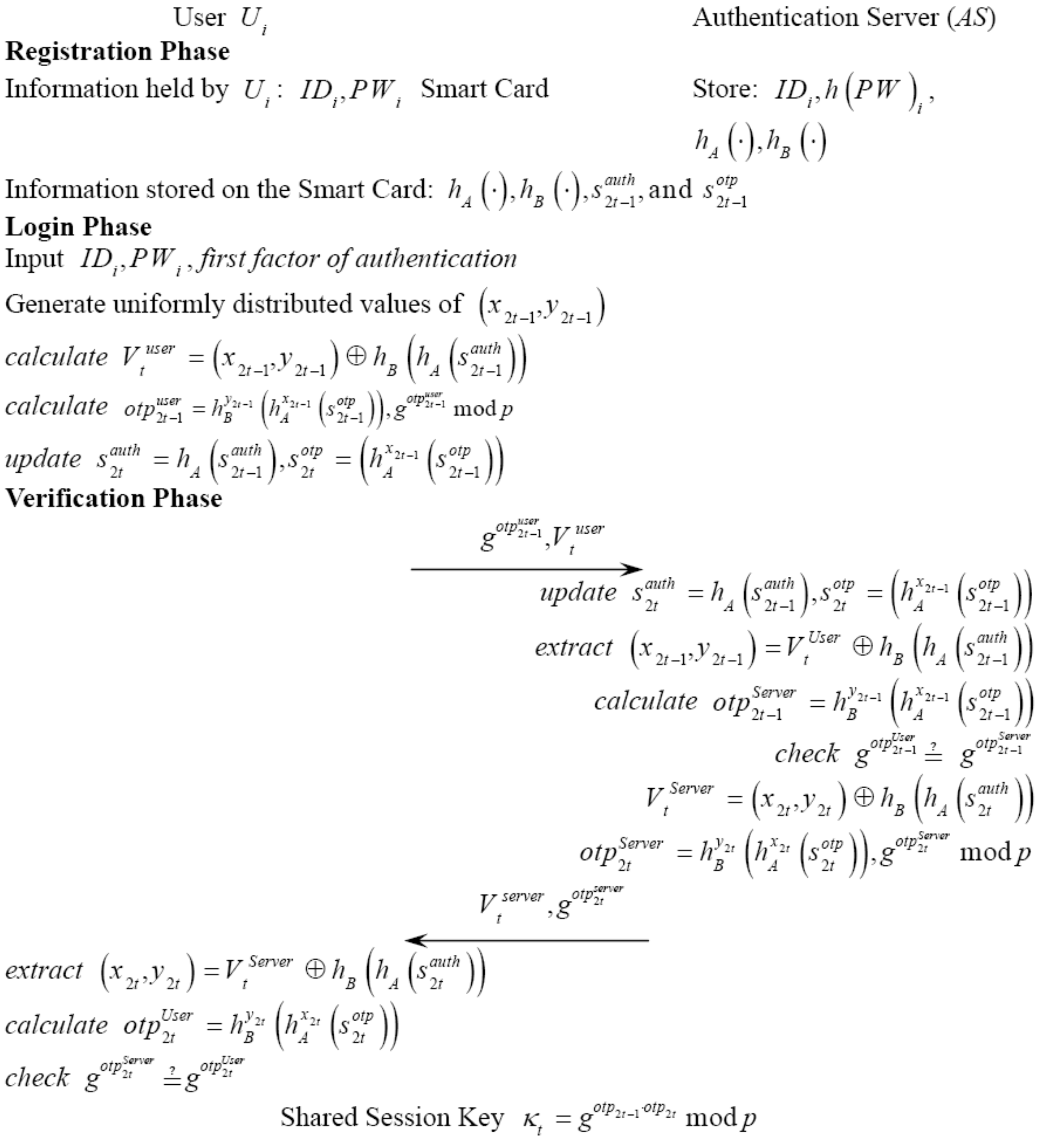
FIGURE 3. Kim-Chung's user authentication scheme

identity, Wang-Li's authentication scheme fails to provide mutual authentication as the authors claim.

4.2. **Masquerade attack.** We next consider the scenario of a masquerade attack on Wang-Liu-Xiao-Dan's scheme. The adversary can masquerade as any legal user, as follows. He (or she) just selects two random numbers, P and I, as U_i's password and falsified identification, and computes a new $N_i' = h(P) \oplus h(x) \oplus I$ for U_i. After that, the adversary can forge a login request by U_i by computing $CID_i = h(P) \oplus h(N_i' \oplus y \oplus T_1) \oplus I$, where T_1 is the timestamp at the adversary's attack, and sends a message, $\{I, CID, N_i', T_1\}$, to S_j. Then, the message can pass the server's verification phase because the used timestamp, T_1, would successfully pass the check $T_2 - T_1 \leq \Delta T$ and I would be matched with ID_i' computed by S_j, which would be derived by $ID_i' = N_i' \oplus h(x) \oplus h(P)'$. Thereby, S_j will accept the adversary's login request.

4.3. **Password guessing attack.** We consider the scenario of a masquerade attack on Kim-Chung's scheme. In the registration phase, the server, S_j, stores $\{K_1, K_2, R, h(\cdot)\}$ into the smart card, SM_i, of U_i. If an adversary has got the smart card, SM_i, and extracted the secret values from it, he (or she) can easily figure out U_i's password by performing an off-line password guessing attack as follows. The adversary picks a candidate password, P, computes $h(P)$ and $C_1' = K_1 \oplus h(P)$, and checks whether C_1' is equal to K_2. A match indicates a correct guess of the password only if the used hash function is really collision free. However, if it is not, the attacker needs an additional verifier to check the guessed password, which is one value from one of the previous login messages of U_i. However, it is not difficult to get the information because of using an insecure channel. An additional verification could be processed by computing $C_2' = K_2 \oplus h\left(P \oplus h\left(P\right)\right)$ and checking whether C_2' is equal to C_2 in the previous message. A match indicates the success of the offline password guessing. Then, the adversary could impersonate the user, U_i, of the smart card, SM_i.

5. **Proposed Remote User Authentication Algorithm.** In this section, we propose a new remote user authentication scheme that eliminates the security flaws described in the previous sections, as well as dispensing with time stamping. Figure 4 illustrates the proposed remote user authentication scheme. To resist such attacks, the proposed login and authentication phases are performed as follows.



User U_i Authentication Server (AS)

Registration Phase

Information held by U_i: ID_i, PW_i Smart Card Store: $ID_i, h\left(PW\right)_i$, $h_A\left(\cdot\right), h_B\left(\cdot\right)$

Information stored on the Smart Card: $h_A\left(\cdot\right), h_B\left(\cdot\right), s_{2t-1}^{auth}$, and s_{2t-1}^{otp}

Login Phase

Input ID_i, PW_i, *first factor of authentication*

Generate uniformly distributed values of $\left(x_{2t-1}, y_{2t-1}\right)$

calculate $V_t^{user} = \left(x_{2t-1}, y_{2t-1}\right) \oplus h_B\left(h_A\left(s_{2t-1}^{auth}\right)\right)$

calculate $otp_{2t-1}^{user} = h_B^{y_{2t-1}}\left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right), g^{otp_{2t-1}^{user}} \bmod p$

update $s_{2t}^{auth} = h_A\left(s_{2t-1}^{auth}\right), s_{2t}^{otp} = \left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$

Verification Phase

$$\xrightarrow{g^{otp_{2t-1}^{user}}, V_t^{user}}$$

update $s_{2t}^{auth} = h_A\left(s_{2t-1}^{auth}\right), s_{2t}^{otp} = \left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$

extract $\left(x_{2t-1}, y_{2t-1}\right) = V_t^{User} \oplus h_B\left(h_A\left(s_{2t-1}^{auth}\right)\right)$

calculate $otp_{2t-1}^{Server} = h_B^{y_{2t-1}}\left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$

check $g^{otp_{2t-1}^{User}} \overset{?}{=} g^{otp_{2t-1}^{Server}}$

$V_t^{Server} = \left(x_{2t}, y_{2t}\right) \oplus h_B\left(h_A\left(s_{2t}^{auth}\right)\right)$

$otp_{2t}^{Server} = h_B^{y_{2t}}\left(h_A^{x_{2t}}\left(s_{2t}^{otp}\right)\right), g^{otp_{2t}^{Server}} \bmod p$

$$\xleftarrow{V_t^{server}, g^{otp_{2t}^{server}}}$$

extract $\left(x_{2t}, y_{2t}\right) = V_t^{Server} \oplus h_B\left(h_A\left(s_{2t}^{auth}\right)\right)$

calculate $otp_{2t}^{User} = h_B^{y_{2t}}\left(h_A^{x_{2t}}\left(s_{2t}^{otp}\right)\right)$

check $g^{otp_{2t}^{Server}} \overset{?}{=} g^{otp_{2t}^{User}}$

Shared Session Key $\kappa_t = g^{otp_{2t-1} \cdot otp_{2t}} \bmod p$

FIGURE 4. Proposed user authentication scheme

We consider the registration stage to be established manually between U_i and AS. The establishment of the algorithms and seeds requires manual intervention, e.g., U_i should go personally to the AS administrator to establish the system.

5.1. **Registration phase.** The *user* gets the two different hash functions established on his token plus two different seeds, $h_A(\cdot)$, $h_B(\cdot)$, s_{2t-1}^{auth}, and s_{2t-1}^{otp}, which are also installed on his token. Moreover, he chooses p, q and g, where p is a large prime number with bit size 1024, q is a prime divisor of $(p-1)$ with size 160-bit, and g is an element of order q in the Galois field, $GF(p)$. The bit size of the output of $h_A(\cdot)$ and $h_B(\cdot)$ is $|q|$. To ensure that the information is completely shared with the service provider, these two seeds are produced by the shared and unique parameters of the *host* and *user*.

5.2. **Login and authentication phase.** For the *first time* of the authentication process, after being prompted for his ID_i and PW_i, the *user* sends the following vector, $V_t^{user} = (x_{2t-1}, y_{2t-1}) \oplus h_B\left(h_A\left(s_{2t-1}^{auth}\right)\right)$ and $otp_{2t-1}^{user} = h_B^{y_{2t-1}}\left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$. The *user updates* $s_{2t}^{auth} = h_A\left(s_{2t-1}^{auth}\right)$, $otp_{2t} = \left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$. Upon the receipt of V_t^{user} and otp_{2t} the AS can extract $(x_{2t-1}, y_{2t-1}) = V_t^{user} \oplus h_B\left(h_A\left(s_{2t-1}^{auth}\right)\right)$ and calculate $otp_{2t-1} = h_B^{y_{2t-1}}\left(h_A^{x_{2t-1}}\left(s_{2t-1}^{otp}\right)\right)$ and after that check that $g^{otp_{2t-1}^{user}} \overset{?}{=} g^{otp_{2t-1}}$. Unless positive results are obtained, the authentication process fails. Otherwise, the AS calculates $V_t^{server} = (x_{2t}, y_{2t}) \oplus h_B\left(h_A\left(s_{2t}^{auth}\right)\right)$ and $otp_{2t}^{server} = h_B^{y_{2t}}\left(h_A^{x_{2t}}\left(s_{2t}^{otp}\right)\right)$ and sends them to U_i, who *checks* $g^{otp_{2t}^{server}} \overset{?}{=} g^{otp_{2t}}$. Upon confirmation, both AS and U_i will calculate the shared session Key, $\kappa = g^{otp_{2t-1} \cdot otp_{2t}} \bmod p$.

6. **Numerical Illustration.** In this section we need to present a practical and convincing model to address the practical use of our theoretical results. In the stage of registration we need to agree on the following:

p *is a prime number of 1024 bit length:*
<113832011549867329696708833974581573235126635938227811759517696593165356147
826919743919486857810042468001180059040935817408568806283436004554602757110816
09762015404559356724047739507598203121048625116151645185280314442750066289691
62501791003556740904129625834467759003448810834088606424601623148591776308831
01>.

q *is a prime number of typically 160 bits with* $q|(p-1)$*:*
<871171773626971368137989833740146907250185681957>

g *is a generator of* $\mathbb{Z}_p^*$, $g = k^{\frac{p-1}{q}} \bmod p$ *at* k *is any integer with* $1 < k < p-1$, $k = 2$*:*
<105161432990682464142076827632386108899592081258027731338299554880702187057
535655776472902490671889440540477652651935062683544385981671991470227655578881
43780252040986721650385303945690459668713147760616894172084908589525225286593
28694454267715029406684300936133819509165804814298978035734395833863598990535
96>.

$h_A(\cdot)$ and $h_B(\cdot)$ are presented by SHA-1 and MD5 in order, also s^{auth} and s^{otp} take the following initial values <123456789> and <987654321>.

In the step of login and authentication, in the earliest authentication round $t = 1$, the remote user generates (x_1, y_1), *randomly*, to be <243> and <521>, also calculates $V_1^{user} = (x_1, y_1) \oplus h_B\left(h_A\left(s_1^{auth}\right)\right)$ = <210058283814880835179777629747971242422>, <210058283814880835179777629747971242828>. Also, he calculates $otp_1^{user} = h_B^{521}(h_A^{243}(s_1^{otp}))$ = <225521356074620219945945908103201857766>, as well as $g^{otp_1^{User}} \bmod p$ <2772
368388327286650905226934603796267707712402175048178182223399455129298272742257
5394292336630159942754375841689108975869557244057430458570948977496067707784

11428127778748756281064767340018724945679911649597354954165041895416284559773 6465793230985623610191429056370738770080599545117666377599715016534013795>.

Seeds updating is going as the following; $s_2^{auth} = h_A\left(s_1^{auth}\right)$ = <1257740151873880076561077972214116076863512829942>, $s_2^{otp} = \left(h_A^{243}\left(s_1^{otp}\right)\right)$ = <72708630625729893597492935264584363010338762O287>.

In the verification phase, the user transfers $g^{otp_1^{user}}$, V_1^{user}, *calculated above*, to his remote server to let it calculate the challenges values (x_1, y_1) to obtain otp_1^{Serevr} and consequently $g^{otp_1^{Serevr}} \bmod p$. Upon the validity check of $g^{otp_1^{user}} \overset{?}{=} g^{otp_1^{Server}}$, the authentication server starts generating two uniformly distributed random values (x_2, y_2) = <359> and <434>, and then calculates $V_1^{server} = (x_2, y_2) \oplus h_B\left(h_A\left(s_2^{auth}\right)\right)$ = <127823686367185312175428045796710836843>, <127823686367185312175428045796710836926> and $otp_2^{Server} = h_B^{434}(h_A^{359}(s_2^{otp}))$ = <26479610273285708151578045056215244366O>, and consequently $g^{otp_2^{Server}} \bmod p$ = <613538641832168849153238326656386078983912171573716904315833194195811419632865958100624860156571862054771075863382013934993257516097688065985292686609691137626366710579983610584604327569816957863894434187400289576068760007794857799034626089054547865882640112629254652020196738766050147275231222721357824 56623>. Now, the authentication server responds to remote user with V_1^{Server} and $g^{otp_2^{Server}}$. This allows the user to extract (x_2, y_2) to reach otp_2^{Serevr} and consequently $g^{otp_2^{Serevr}} \bmod p$. Upon the validity check of $g^{otp_2^{User}} \overset{?}{=} g^{otp_2^{Server}}$, both user and server calculate the shared session key of $\kappa_1 = g^{otp_1 \cdot otp_2} \bmod p$ = <682460211571144730746333956011686287542359119275656295863287884238886350387485259469429225639081162903219386372460735856942191446849274011333943615148742082028907847209572383096281134898147573027675925689125306804820221503189643112151611392278877023517949600572253646241458047769359144780944198042338191743 73>.

The second hash function h_B allowed us to go in the forward direction by protecting the h_A produced chain. Also, it is not admissible for (x_{2t-1}, y_{2t-1}) to be equal to zero.

7. Security Analysis.

Naturally, the proposed scheme can resist an off-line guessing attack because it uses strong passwords with strong hash functions. Moreover, the gaining of unauthorized access by replaying reusable passwords is restricted by encoding passwords, which are used once. In this section, we will briefly give a security assessment of our proposed scheme.

7.1. Pre-play attack.

Unless the challenge is protected, a type of "suppress-replay attack" (known as a "pre-play attack") becomes possible [2]. Consider that an intruder, E, who is able to predict the next challenge, wishes to impersonate AS to U_i. E takes the AS role, by impersonating it to U_i. The challenge sent by U_i is memorized by E. Then, at some future time, E can impersonate U_i to AS, using this memorized response. Our proposal allows the challenges to be unpredictable uniformly distributed values of x_t and y_t. If we suppose that x_t and y_t can take one value of forward m values, the probability of successfully guessing a challenge will be the joint probability of x_t and y_t, which is equal to $1/m^2$. We can refer to this property as the ability to *resist predictable attacks.* The restriction of transferring password information in just one direction, from *user* to *host*, also increases the robustness against this type of attack. The two exchanged vectors between *user* and *host* are transferred in a cipher format.

7.2. Non-repudiation attack.

To prevent a non-repudiation attack, the *user* and AS have to update s^{auth} and s^{otp} twice per session. Hence, the *user* updating will be used as the *host's* next verifier, and vice versa, so that any unauthorized modification of the exchanged vectors will be detected by the authentication partners.

7.3. **Forgery attack.** To mount a forgery attack on the proposed scheme, an adversary must generate an authentication message corresponding to the given $g^{otp^{user}_{2t-1}}$, V^{user}_t *vectors*. Because the adversary does not know s^{auth} and s^{otp}, he cannot correctly update the seeds nor correctly produce session *OTPs* that will be accepted by the communicating parties. Hence, the proposed scheme can resist the forgery attack.

7.4. **Insider attack.** If an *AS* insider tries to impersonate the *user* to access other *host*s using the shared Session Key, $\kappa = g^{otp_{2t-1} \cdot otp_{2t}} \bmod p$ between them, s/he will not be able to do so because the cooperation of the *OTPs'* seed fabrication between this *user* and the different *hosts* is strong. Furthermore, as the *OTP* production, using two different types of strong hashes, $h_A(\cdot)$ and $h_B(\cdot)$, is strong, the *host* insider cannot derive those *OTPs* by performing an off-line guessing attack on what he has received.

7.5. **Small challenge attack.** Attacks based on sending small challenges by intruders who impersonate the communication *host* only affect the backward hash chains' *OTPs*. Our scheme uses forward hashing techniques, which eliminates this type of attack completely.

7.6. **Mutual authentication.** The *AS* is authenticated to U_i by checking the equality of $g^{otp^{user}_{2t-1}} \overset{?}{=} g^{otp_{2t-1}}$ and the user is also authenticated to the *host* by verifying $g^{otp^{server}_{2t}} \overset{?}{=} g^{otp_{2t}}$. This confirms the occurrence of the two-way authentication.

7.7. **Known key security.** The protocol provides known-key security. Hence, each run of the protocol between two principal U_i and *AS* should produce a unique session key, which depends on otp_{2t} and otp_{2t-1}. Even if an adversary has learned some other session keys, he cannot predict a new otp_{2t} nor a new otp_{2t-1}. Therefore, the protocol still achieves its goal in the face of the adversary.

7.8. **Small subgroup attack.** One way to avoid a small subgroup attack [19] is to make G a $\mathbb{Z}^*_p$ subgroup of a prime order, q. This is done by choosing g, a subgroup generator, to have a prime order q. In this case, the only proper subgroup of G consists of a single identity element. Moreover, the provided key confirmation property has the responsibility to notify the principals of this type of attack. Hence, the shared secret will be equal to $\kappa^w \neq \kappa$, at $w = (p-1)/r$, where r is a small factor of $(p-1)$.

7.9. **Implicit key authentication.** An authentication protocol is said to provide implicit key authentication (of U_i to *AS*) if the *AS* is assured that no other entity aside from a specifically identified U_i can learn the value of a particular secret key. Any modification of the exchanged messaged cannot be recovered by an intruder. Hence, the random challenges values are only known to the challenger, and s^{auth} and s^{otp} are only known by the communicating parties.

7.10. **Key confirmation.** The assurance of the rightful participants in a key-establishment protocol is that the intended recipient of the shared key actually possesses the shared key. The two checks of $g^{otp^{server}_{2t}} \overset{?}{=} g^{otp_{2t}}$ and $g^{otp^{user}_{2t-1}} \overset{?}{=} g^{otp_{2t-1}}$ have the responsibility of providing this key confirmation.

7.11. **Explicit key authentication.** If both implicit key authentication and key confirmation (of U_i to *AS*) are provided, the key establishment is said to provide *explicit key authentication* (*EKA*) (of U_i to *AS*). A key agreement that provides explicit key authentication to both principals is called *authenticated key agreement with key confirmation* (*AKC*) *protocol*.

7.12. **Key control.** Neither of the principles must be able to force the key to be any chosen value; otherwise one party could force the use of an old key. One potential benefit is that each principle does not have to rely on any other party to generate appropriate keys. As long as neither party is malicious, it can often be guaranteed that the session key is sufficiently random inputs by the utilization of random and uniformly distributed values (x, y). A related benefit is that the principals can often be sure that the session key is fresh by ensuring that their own input is fresh.

7.13. **Key freshness.** The derived session key is fresh, as opposed to the reuse of old keying material. This is to be done by maintaining the randomization of the generated *otp* and consequently the generated session key $\kappa = g^{otp_{2t-1} \cdot otp_{2t}} \bmod p$.

8. **Performance Analysis.** To compare the performances of our scheme and the other schemes we should consider the computational cost for each one, by counting the number of Hash and Exponential operations for the tth login with respect to the chain algorithm from the *user* side, U_i. The hash computation costs of the proposed scheme in the registration, login, and authentication phases are summarized in Table 1.

Table 1. Comparison of our scheme and its related schemes in terms of computational costs

	Proposed Scheme		Kim-Chung's Scheme	
	U_i	AS	U_i	AS
Login Phase	$\begin{pmatrix}2+x_{2t-1}\\+y_{2t-1}\end{pmatrix}$ Hash +1Xor + 1Exp	–	4Xor +3Hash	–
Verification Phase	$\begin{pmatrix}2+x_{2t}\\+y_{2t}\end{pmatrix}$ Hash +1Xor +1Exp	$\begin{pmatrix}4+x_{2t-1}\\+x_{2t}+\\y_{2t-1}+y_{2t}\end{pmatrix}$ Hash +2Xor +2Exp	3Xor +1Hash	9Xor +3Hash
Communication Costs	$\begin{pmatrix}160+\\1024\end{pmatrix}$ bits	$\begin{pmatrix}160+\\1024\end{pmatrix}$ bits	–	–
Time Stamping	NOT-required		Required	
	Wang-Liu-Dan's Scheme		Wang-Li's Scheme	
	U_i	AS	U_i	AS
Login Phase	4Xor + 2Hash	–	1Exp + 3Hash +1Xor	–
Verification Phase	2Xor + 2Hash	8Xor + 3Hash	1Exp +2Hash	2Exp +5Hash
Communication Costs	–	–	$\begin{pmatrix}160+\\1024\end{pmatrix}$ bits	$\begin{pmatrix}160+\\1024\end{pmatrix}$ bits
Time Stamping	Required		Required	

Note that Exp denotes Exponentiation operations, $(\oplus)$ stands for an operation of exclusive-or, and Hash means Cryptographic hash operations. It is clear that our scheme's computation cost Hashes of the login and verification phases are increased by the random and uniformly distributed values (x, y). This is the payment for removing time stamping and time synchronization. An authentication mechanism that uses time-stamping [14] requires time synchronization capability for a certain reference between the U_i and AS, which is not easy to apply in many applications. All of the algorithms that have previously been discussed need accurate time synchronization to achieve a reliable time-stamping in the login and the authentication process. However, our proposed algorithm does not need any time synchronization to prevent the clock un-synchronization problem. Although our proposed protocol additionally needs (x, y) hash function operations to complete the whole authentication procedure, our scheme is efficient. This is because only lightweight operation modules such as a one-way hash function and exclusive-or operation are required. In Borst et al., [20] the authors showed that there are ways to implement a sufficiently fast hash function on a smart card to perform authentication and other access control mechanisms.

9. **Conclusions.** Identifying remote users as legal or illegal is a key issue in network security. This study proposed a remote user authentication scheme using two-nested-one-way hash functions and the discrete logarithm problem. The proposed authentication scheme is different from other such schemes in avoiding the need for time stamping and consequently time synchronization. Because our proposed algorithm mainly uses the one-way hash function in the login and authentication processes, thus the problems associated with the cost of computation can be avoided. A detailed security analysis was also performed that covered many types of attacks that could influence our scheme. The proposed scheme not only provides mutual authentication between the user and server, but also establishes a common session key to provide message confidentiality. Future research will apply the proposed scheme to more complex user authentication environments with smart cards such as a multi-server environment with a single user in which clients can be remotely authenticated once and access multiple servers at different locations as many times as they want.

REFERENCES

[1] L. Lamport, Password authentication with insecure communication, *Comm. ACM*, vol.24, no.11, pp.770-772, 1981.

[2] C. J. Mitchell and L. Chen, Comments on the S/KEY user authentication scheme, *ACM Operating System Review*, vol.30, no.4, pp.12-16, 1996.

[3] R. Lennon, S. Matyas and C. Mayer, Cryptographic authentication of time-variant quantities, *IEEE Trans. on Commun.*, vol.29, no.6, pp.773-777, 1981.

[4] T. C. Wu, Remote login authentication scheme based on a geometric approach, *Computer Communication*, vol.18, no.12, pp.959-963, 1995.

[5] M. S. Hwang and L. H. Li, A new remote user authentication scheme using smart cards, *IEEE Trans. Consumer Electronic*, vol.46, no.1, pp.28-30, 2000.

[6] S. M. Yen and K. H. Liao, Shared authentication token secure against replay and weak key attack, *Information Processing Letters*, vol.62, no.2, pp.77-80, 1997.

[7] C. K. Chan and L. M. Cheng, Cryptanalysis of a re-mote user authentication scheme using smart cards, *IEEE Trans. on Consumer Electronic*, vol.46, pp.992-993, 2000.

[8] J. J. Shen, C. W. Lin and M. S. Hwang, A modified remote user authentication scheme using smart cards, *IEEE Trans. on Consumer Electronic*, vol.49, no.2, pp.414-416, 2003.

[9] C. C. Chang and K. F. Hwang, Some forgery attack on a remote user authentication scheme using smart cards, *Informatics*, vol.14, no.3, pp.189-294, 2003.

[10] K. C. Leung, L. M. Cheng, A. S. Fong and C. K. Chen, Cryptanalysis of a remote user authentication scheme using smart cards, *IEEE Trans. on Consumer Electronic*, vol.49, no.3, pp.1243-1245, 2003.

[11] A. K. Awasthi and S. Lal, A remote user authentication scheme using smarts cards with forward secrecy, *IEEE Trans. on Consumer Electronic*, vol.49, no.4, pp.1246-1248, 2003.

[12] K. Manoj, Some remarks on a remote user authentication scheme using smart cards with forward secrecy, *IEEE Trans. on Consumer Electronic*, vol.50, no.2, pp.615-618, 2004.

[13] S. W. Lee, H. S. Kim and K. Y. Yoo, Comment on a remote user authentication scheme using smart cards with forward secrecy, *IEEE Trans. on Consumer Electronic*, vol.50, no.2, pp.576-577, 2004.

[14] K. Manoj, New remote user authentication scheme with smart cards, *IEEE Trans. on Consumer Electronic*, vol.50, no.2, pp.597-600, 2004.

[15] B. Wang and Z. Q. Li, A forward-secure user authentication scheme with smart cards, *International Journal of Network Security*, vol.3, no.2, 2006.

[16] Y. Y. Wang, J. Y. Liu, F. X. Xiao and J. Dan, A more efficient and secure dynamic ID-based remote user authentication scheme, *Computer Communications*, vol.32, pp.583-585, 2009.

[17] S. K. Kim and M. G. Chung, More secure remote user authentication scheme, *Computer Communications*, vol.32, pp.1018-1021, 2009.

[18] E. J. Yoon and K. Y. Yoo, More efficient and secure remote user authentication scheme using smart cards, *Proc. of the 11th International Conference on Parallel and Distributed Systems*, vol.2, pp.73-77, 2005.

[19] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, An efficient protocol for authenticated key agreement, *Technical Report CORR 98-05*, University of Waterloo, Canada, 1998.

[20] J. Borst, B. Preneel and V. Rijmen, Cryptography on smart cards, *Computer Networks*, vol.36, no.4, pp.423-435, 2001.

[21] C.-T. Li, C.-C. Lee, C.-J. Liu and C.-W. Lee, A robust remote user authentication scheme against smart card security breach, *LNCS*, vol.6818, pp.231-238, 2011.

[22] E.-J. Yoon and K.-Y. Yoo, Enhanced forward-secure user authentication scheme with smart cards, *Public Key Infrastructure, LNCS*, vol.4043, pp.197-206, 2006.

[23] E.-J. Yoon and K.-Y. Yoo, Efficient mutual authentication scheme with smart card, *Agent Computing and Multi-Agent Systems, LNCS*, pp.813-818, 2006.

[24] K. Manoj, An enhanced remote user authentication scheme with smart card, *I. J. Network Security*, vol.10, no.3, pp.175-184, 2010.

[25] H.-S. Kim and S.-W. Lee, Robust remote user authentication scheme using smart cards, *Journal of Security Engineering*, pp.495-501, 2010.

[26] Y.-P. Liao and S.-S. Wang, A secure dynamic ID based remote authentication scheme for multi-server environment, *Computer Standards and Interfaces*, vol.31, pp.24-29, 2009.

[27] M. K. Khan and J. Zhang, Improving the security of a flexible biometrics remote user authentication scheme, *Computer Standards and Interfaces*, vol.29, no.1, pp.82-85, 2007.

[28] X.-M. Wang, W.-F. Zhang, J.-S. Zhang and M. K. Khan, Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards, *Computer Standards and Interfaces*, vol.29, no.5, pp.507-512, 2007.

[29] I.-S. Jeon, H.-S. Kim and M.-S. Kim, Enhanced biometrics-based remote user authentication scheme using smart cards, *Journal of Security Engineering*, vol.8, no.2, pp.237-254, 2011.

[30] S.-W. Lee, H.-S. Kim and K.-Y. Yoo, Improvement of Chien et al.'s remote user authentication scheme using smart cards, *Computer Standards and Interfaces*, vol.27, no.2, pp.181-183, 2005.

[31] E.-J. Yoon, E.-K. Ryu and K.-Y. Yoo, An improvement of Hwang-Lee-Tang's simple remote user authentication scheme, *Computers and Security*, vol.24, no.1, pp.50-56, 2005.

[32] Y. Lee and D. Won, Security vulnerabilities of a remote user authentication scheme using smart cards suited for a multi-server environment, *ICCSA*, no.2, pp.164-172, 2009.

[33] M. H. Eldefrawy, M. K. Khan, K. Alghathbar, T. H. Kim and H. Elkamchouchi, Mobile-one-time-password: Two-factor authentication using mobile phones, *Security and Communication Networks, John Wiley & Sons*, vol.5, no.5, pp.508-516, 2012.