http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (3): 333-339, 2009
ISSN 1812-5638

A Secure Remote Mutual Authentication and Key Agreement without Smart Cards

[1,2]Han-Cheng Hsiang and [1]Wei-Kuan Shih
[1]Department of Computer Science, National Tsing Hua University, Taiwan, Republic of China
[2]Department of Information Management, Vanung University, Taiwan, Republic of China

Abstract: This study proposes a new and secure scheme for remote mutual authentication without using the smart cards. The scheme may satisfy all of the essential security requirements. In the last couple of decades the Internet technology has advanced so rapidly. It leads to the spreading and penetration of the technology to the network services and applications. Remote user authentication is a very effective means to check the legality of a user. Among many schemes, password authentication has been commonly used. Also in many schemes proposed for the remote user authentication, smart card has been intensively used to store the secret information for authentication. However, the smart card and its reader are not always available everywhere and in anytime. With this scheme, the user can login to the remote server from anywhere and in anytime to access the secure service. This may be more practical and easy-to-use.

Key words: Password authentication, security attacks, information security, communications

INTRODUCTION

Remote user authentication scheme allows a server to check the legality of a remote user through open network. In addition, a smart card based remote mutual authentication scheme is very efficient to authenticate remote users (Juang, 2006; Wang *et al*., 2007). Lamport (1981) proposed the first well-known password based remote user authentication scheme rooted on verifier table, but this scheme was vulnerable to stolen verifier attack. Hwang *et al*. (1990) initially proposed a non-interactive password authentication scheme using smart cards, without storing verifier table in the server. Since then Tan and Zhu (1999), Yang and Shieh (1999), Hwang and Li (2000) and Sun (2000) have proposed new schemes to improve the efficiency and the security of remote authentication. Chien *et al*. (2002) proposed an efficient remote mutual authentication scheme using smart card allowing server and user to authenticate each other. However, Chien *et al*. (2002) scheme was vulnerable to the parallel session attack (Hsu, 2004). Later, Juang (2004) proposed another scheme preserving all the merits of Chien *et al*. (2002) scheme. Juang (2004) scheme is nonce based authentication and key agreement scheme. No synchronized clocks are required in the scheme. Besides, Juang (2004) scheme generates a session key for the user and server in their following communication.

Recently, Shieh and Wang (2006) pointed out the security flaws of Juang (2004) scheme and then proposed an improvement to remedy the security flaws. However, most of the user authentication schemes were designed using smart cards. In practice, card readers are not available everywhere. In particular, it is difficult to produce tamper-resistant smart cards. Research shows that the secret information kept in smart cards can be cracked by analyzing the dripped messages or the power consumption (Messerges *et al*., 2002). This study proposes a secure and efficient password authentication protocol without the use of smart cards. The proposed scheme fulfills the following requirements, which are regarded as important criteria for password authentication (Sun, 2000; Juang, 2004; Lee *et al*., 2008).

Requirement 1: Freely chosen password: The user can freely choose his password for easy memorization. This requirement is essential to make the password authentication protocol more user-friendly.

Requirement 2: Security: Security is the most important issue of any password authentication protocol. The password authentication scheme must be able to resist various kinds of attacks such that it can be applied in the real world.

Requirement 3: Session key agreement: The legal user and the server should be able to negotiate a session key to protect the transmitting messages after successful authorization.

Corresponding Author: Han-Cheng Hsiang, Department of Computer Science, National Tsing Hua University, 101, Kuang Fu Road, Sec. 2, Hsing Chu, 300, Taiwan, Republic of China

Requirement 4: Mutual authentication: The authentication scheme allows the users and the remote server to authenticate each other. It is necessary to protect not only the server but also the legal user from malicious attacks.

This study presents a secure and efficient mutual authentication scheme without smart cards. The proposed scheme providing perfect forward secrecy, can withstand various malicious attacks.

BACKGROUND

Perfect forward secrecy: For evaluating a strong protocol, perfect forward secrecy is considered to be an important security issue. A protocol providing perfect forward secrecy means that even if one entity's long-term secret key is compromised, it will never reveal any old short-term keys used before (Menezes *et al*., 1997; Sun and Yeh, 2006). For example, the well-known Diffie-Hellman key agreement scheme (Diffie and Hellman, 1976) can provide perfect forward secrecy.

Diffie-Hellman problem: Given a prime p, a generator g and two numbers g^d mod p and g^e mod p, one tries to find g^{de} mod p, it is believed infeasible to solve in polynomial time (Diffie and Hellman, 1976).

RELATED WORKS

Here, briefly reviews Juang's and Shieh-Wang's scheme. The notations are used throughout here.

U	:	The user
S	:	The server
ID	:	The identification of U
PW	:	The password of U
h(·)	:	Secure one-way hash function
x	:	The secret key maintained by the server
$E_k(m)$	:	The encryption function of the message m with the encryption key k
$D_k(m)$	:	The decryption function of the message m with the decryption key k
SK	:	The session key shared between U_i and S for this protocol run
⊕	:	Exclusive-or operation
‖	:	String concatenation operation
→	:	A common channel

Review of Juang's scheme: Juang's (2004) scheme is based on the symmetric encryption. The scheme consists of two phases: the registration phase and the login and session key agreement phase.

Registration phase: Assume a user U_i submits his identity ID_i and password PW_i to the server over a secure channel for registration. The server computes $V_i = h(ID_i, x)$, $W_i = V_i \oplus PW_i$ and issues U_i a smart card containing W_i, ID_i and h(·).

Login and session key agreement phase: When U_i wants to login to the server, he (including she) inserts his smart card into a card reader and inputs his identity ID_i and password PW_i.

Step 1: $U_i \rightarrow S$: $ID_i, N_i, E_{PWi}(g^a \bmod p)$.
The smart card computes $V_i = W_i \oplus PW_i$, then sends the message $\{N_u, ID_i, E_{Vi}(ru_j, C_i)\}$ to the server S, where, $C_i = h(ID_i \| N_u)$, E_{Vi} denotes a symmetric encryption algorithm using V_i as the secret key, N_u is a nonce and ru_j is a random value chosen by the smart card to generate the session key SK.

Step 2: $S \rightarrow U_i$: $E_{Vi}(rs_j, N_u+1, N_s)$.
After receiving the message, S computes $V_i = h(ID_i, x)$ and $(ru_j, C_i) = D_{Vi}(E_{Vi}(ru_j, C_i))$, where, $D_{Vi}(\cdot)$ denotes the corresponding symmetric decryption algorithm of E_{Vi} using V_i as the secret key. After decryption, if C_i is not equal to $h(ID_i \| N_u)$, or N_u is not fresh, the server rejects U_i's request. Otherwise, the server sends the message $E_{Vi}(rs_j, N_u+1, N_s)$ to U_i, where, N_s is a nonce and rs_j is a random value chosen by the server to generate the session key SK.

Step 3: $U_i \rightarrow S$: $E_{SK}(N_s+1)$.
When U_i receives the message, the smart card decrypts and checks whether N_u+1 is in it. If yes, the smart card computes the session key $SK = h(rs_j, ru_j, V_i)$ and sends the message $E_{SK}(N_s+1)$ back to S.

Step 4: On receiving the last message, S computes $D_{SK}(E_{KJ}(N_s+1))$ to check whether, N_s+1 is in it. If N_s+1 is found, S and U_i have achieved mutual authentication and session key agreement.

Review of Shieh-Wang's scheme: Shieh and Wang's (2006) scheme consists of three phases: the registration phase, the login phase and the authentication and key agreement phase. The scheme works as follows:

Registration phase: Assume a user U_i submits his identity ID_i and password PW_i to the server over a secure channel for registration. If the request is accepted, the

server computes $R_i = h(ID_i \oplus x) \rightarrow PW_i$ and issues U_i a smart card containing R_i and $h(\cdot)$.

Login phase: When the user U_i wants to login to the server, he first inserts his smart card into a card reader then inputs his identity ID_i and password PW_i. The smart card then performs the following steps to begin an access session:

Step 1: Compute $a_i = R_i \oplus Pw_i$.

Step 2: Acquire current time stamp T_u, store T_u temporarily until the end of the session and compute $MAC_u = h(T_u||a_i)$.

Step 3: $U_i \rightarrow S$: ID_i, T_u, MAC_u.
Send the message $\{ID_i, T_u, MAC_u\}$ to the server and wait for response from the server. If no response is received in time or the response is incorrect, report login failure to the user and stop the session.

Authentication and key agreement phase: After receiving the message $\{ID_i, T_u, MAC_u\}$ from U_i, the server performs the following steps to assure the integrity of the message:

Step 1: Check the freshness of T_u. If T_u has already appeared in a current executing session of user U_i, reject U_i's login request and stop the session. Otherwise, T_u is fresh.

Step 2: Compute $a_i' = h(ID_i \oplus x)$, $MAC_u' = h(T_u||a_i')$ and check whether, MAC_u' is equal to the received MAC_u. If it is not, reject U_i's login and stop the session.

Step 3: $S \rightarrow U_i$: T_u, T_s, MAC_s.
Acquire the current time stamp T_s. Store temporarily paired time stamps (T_u, T_s) and ID_i for freshness checking until the end of the session. Compute $MAC_s = h(T_u||T_s||a_i')$ and session key $SK = h((T_u||T_s) \oplus a_i')$. Then, send the message $\{T_u, T_s, MAC_s\}$ back to U_i and wait for response from U_i. If no response is received in time or the response is incorrect, reject U_i's login and stop the session.

Step 4: On receiving the message $\{T_u, T_s, MAC_s\}$ from the server, the smart card checks if the received T_u is equal to the stored T_u to assure the freshness of the received message. If it is not, report login failure to the user and stop the session.

Step 5: Compute $MAC_s' = h(T_u||T_s||a_i)$ and check whether it is equal to the received MAC_s. If not, report login failure to the user and stop. Otherwise, conclude that the responding party is the real server.

Step 6: $U_i \rightarrow S$: T_s, MAC_u''.
Compute $MAC_u'' = h(T_s||(a_i + 1))$ and session key $SK = h((T_u||T_s) \oplus a_i)$, then send the message $\{T_s, MAC_u''\}$ back to the server. Note that, in the message $\{T_s, MAC_u''\}$, T_s is a response to the server.

Step 7: When the message $\{T_s, MAC_u''\}$ from U_i is received, the server checks if the received Ts is equal to the stored T_s. If it fails, reject U_i's login request and stop the session.

Step 8: Compute $MAC_u''' = h(T_s||(a_i'+1))$ and check whether it is equal to MAC_u''. If it is not, reject U_i's login request and stop the session. Otherwise, conclude that U_i is a legal user and permit the user U_i's login.

At this moment, mutual authentication and session key agreement between U_i and the server are achieved. From now on, the user U_i and the server can use the session key SK in their further secure communication until the end of the access session.

THE PROPOSED SCHEME

Here, we propose a new remote mutual authentication and key agreement scheme without smart cards. The flowchart of the new protocol is shown in Fig. 1. Before describing the details of present protocol, we first list the notations as follows.

$h(\cdot)$: A collision-resistant one-way hash function
x : The secret key maintained by the server

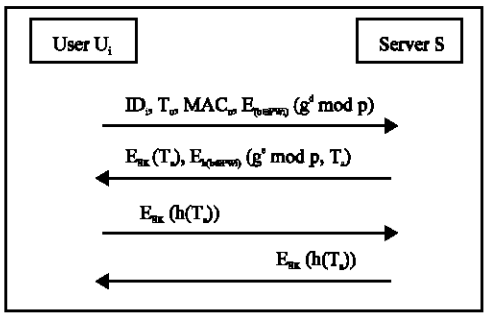

Fig. 1: The flowchart of the proposed scheme

p : A large prime number
g : A primitive element in GF(p)
$E_k(m)$: The encryption function of the message m with the encryption key k
$D_k(m)$: The decryption function of the message m with the decryption key k
SK : The session key shared between U_i and S for this protocol run
$\oplus$: Exclusive-or operation
$\|$: String concatenation operation
$\rightarrow$: A common channel

The proposed scheme consists of three phases: the registration phase, the login phase and the authentication and key agreement phase. The scheme works as follows:

Registration phase: Assume a user U_i selects a random number b, password PW_i and computes h ($b \oplus PW_i$). He submits his identity ID_i and h ($b \oplus PW_i$) to the server over a secure channel for registration. If the request is accepted, the server computes $c_i = h(ID_i \| x)$ and $R_i = c_i \oplus h(b \oplus PW_i)$. Then store R_i in S's database and send an accepted message to U_i through a secure channel.

Login phase: When the user U_i wants to login to the server, he inputs his identity ID_i, password PW_i, b and a large random integer d<p-1. The user then performs the following steps to begin an access session:

Step L1: U_i computes $a_i = h(b \oplus PW_i)$ and E_{ai} (g^d mod p).

Step L2: Acquire current time stamp T_u, store T_u temporarily until the end of the session and compute $MAC_u = h(T_u \| a_i)$.

Step L3: $U_i \rightarrow S$: ID_i, T_u, MAC_u, E_{ai} (g^d mod p).
Send the message {ID_i, T_u, MAC_u, E_{ai} (g^d mod p)} to the server S and wait for response from the server. If no response is received in time or the response is incorrect, report login failure to the user and stop the session.

Authentication and key agreement phase: Upon receiving the message {ID_i, T_u, MAC_u, E_{ai}(g^d mod p)} from U_i, the server S performs the following steps to assure the integrity of the message:

Step V1: Check the freshness of T_u. If T_u has already appeared in a current executing session of user U_i, reject U_i's login request and stop the session. Otherwise, T_u is fresh.

Step V2: Compute $a_i' = R_i \oplus h(ID_i \| x)$, $MAC_u' = h(T_u \| a_i)$ and check whether MAC_u' is equal to the received MAC_u. If it is not, reject U_i's login and stop the session.

Step V3: S chooses a large random integer e<p-1 and computes g^e mod p. S retrieves g^d mod p by computing $D_{ai}'(E_{ai}(g^d \text{ mod } p))$. After all parameters are known, S computes session key $SK = (g^d)^e$ mod p for this scheme to operate.

Step V4: $S \rightarrow U_i$: $E_{SK}(T_s)$, $E_{h(b \oplus PWi)}(g^e \text{mod } p, T_s)$.
Acquire the current time stamp T_s. Store temporarily paired time stamps (T_u, T_s) and ID_i for freshness checking until the end of the session. S computes $E_{SK}(T_s)$. Next, send the message including ID_i, $E_{SK}(T_s)$, $E_{h(b \oplus PWi)}$ (g^e mod p, T_s) to U_i Then, send the message {$E_{SK}(T_s)$, $E_{h(b \oplus PWi)}$ (g^e mod p, T_s)} back to U_i and wait for response from U_i. If no response is received in time or the response is incorrect, reject U_i's login and stop the session.

Step V5: $U_i \rightarrow S$: $E_{SK}(h(T_s))$.
On receiving the message {$E_{SK}(T_s)$, $E_{h(b \oplus PWi)}$ (g^e mod p, T_s)} from the server, U_i computes $D_{h(b \oplus PWi)}(E_{h(b \oplus PWi)}$ (g^e mod p, T_s) to retrieve g^{bs} mod p and T_s. Then U_i computes the session key $SK = (g^e)^d$ mod p and $T = E_{SK}(T_s)$ and then compare T with the received $E_{SK}(T_s)$. If they are not equal, terminate this session. Otherwise, U_i computes $E_{SK}(h(T_s))$ and sends it to S.

Step V6: $S \rightarrow U_i$: $E_{SK}(T_u)$.
Upon receiving the message from U_i, S computes $D_{SK}(E_{SK}(h(T_s)))$ to retrieve $Q = h(T_s)$. Then, S_j computes $Q' = h(T_s)$ by using T_s generated in Step V4 and compares it with Q. If they are not equal, S terminates this session; otherwise, S computes $E_{SK}(T_u)$ and sends the computation result to U_i.

Step V7: After getting the transmitted message, U_i computes $D_{SK}(E_{SK}(T_u))$ and checks if T_u is in the decryption result for freshness checking. If it holds, the authentication is successful; otherwise, the connection is interrupted. After finishing mutual authentication, the user U_i and the remote server S can use the session key SK to encrypt/decrypt the secret information for the following communication.

SECURITY ANALYSIS

Provide perfect forward secrecy: Assuming the random values d and e are large, it is computationally infeasible for an adversary to find g^{de} mod p due to the Diffie-Hellman problem (Diffie and Hellman, 1976). In the proposed scheme, assume that even both the user U_i's password PW_i and the server S's secret key x are all known by an attacker. Then the attacker still cannot decrypt $E_{h(b\oplus PW_i)}$ (g^e mod p) to obtain g^e mod p and decrypt $E_{h(b\oplus PW_i)}$ (g^e mod p, T_s) to obtain g^e mod p, since the attacker does not know PW_i and b. Even if the attacker has known g^d mod p and g^e mod p, he still cannot calculate g^{de} because the difficulty is similar to solve the Diffie-Hellman problem. So, the attacker does not have any opportunity to get the session key SK. Therefore, the session key is still secure. Hence, present scheme provides perfect forward secrecy with high security (Sun and Yeh, 2006).

Resist privileged insider's attack: In the registration phase, a user U_i selects a random number b, password PW_i and then computes $h(b\oplus PW_i)$. The user submits ID_i and $h(b\oplus PW_i)$ to the remote server S. If the privileged insider of S wants to use U_i's password to impersonate U_i to login the other servers, the action will fail. Since U_i registers to S by presenting $h(b\oplus PW_i)$ instead of PW_i, the privileged insider of S can not directly obtain PW_i. Besides, as b is not revealed to S and $h(\cdot)$ is a collision-resistant one-way hash function, the privileged insider of S can not obtain PW_i by performing an off-line guessing attack on $h(b\oplus PW_i)$. Therefore, the proposed scheme can resist the privileged insider attack (Ku *et al*., 2005; Ku and Chen, 2004).

Resist the masquerade attack: If the adversary Eve has stolen U_i's authentication data R_i which stored in S, Eve cannot masquerade as the legal user U_i to login the remote server S_j, since, Eve cannot obtain $h(b\oplus PW_i)$ without knowing the knowledge of both PW_i and b. Hence, the adversary cannot forge a login message to pass S's authentication, the proposed scheme can resist the masquerade attacks.

Resist the stolen-verifier attack: In the proposed scheme, the user U_i's authentication data stored in S is $R_i = h(ID_i||x) \oplus h(b\oplus PW_i)$. Suppose that an attacker Eve has stolen the R_i, she can obtain $h(b\oplus PW_i)$ only if Eve has the information of $h(ID_i||x)$, which implies she knows S's long-term secret key x. Since, $h(b\oplus PW_i)$ is hidden in R_i and the secret key x is under strict protection as assumed, Eve's obtaining $h(b\oplus PW_i)$ in this way is infeasible. In addition, if Eve is a legal user and has stolen her R_e, since $h(\cdot)$ is a collision-resistant one-way hash function, Eve's retrieing x is still computational infeasible. That is, the proposed scheme can resist the stolen-verifier attack.

Resist the server spoofing attack: Any attempt of an attack to impersonate as any remote server S to cheat U_i is infeasible, since he cannot construct the session key SK without the knowledge of PW_i and b. Thus, the attacker cannot decrypt the transmitted messages from some legal user. After communicating with the masqueraded remote server, the legal user can detect immediately and terminates the session. Hence, the proposed scheme can protect the user from being cheated by the masqueraded remote server.

Resist the replay attack: In the proposed protocol, if an attacker retransmits either the login message $\{ID_i, T_u, MAC_u, E_{h(b\oplus PW_i)}(g^d \text{ mod } p)\}$ in Step L3 of Login Phase or the response message $\{E_{SK}(T_s), E_{h(b\oplus PW_i)}(g^e \text{ mod } p, T_s)\}$ in Step V4 of Authentication and key agreement phase, the attacker cannot pass authentication successfully. Since the legality of the messages will be checked with the challenges d, e and T_s. On the contrary, only the legal partners know PW and b to bind the corresponding random numbers and nonce in the encrypted messages. Therefore, the proposed scheme can resist the replay attack.

Resist the man-in-the-middle attack: In the proposed scheme, the transmitted message is protected by $h(b\oplus PW_i)$ during each authentication. Attempt of the attack to produce the correct transmitted message without knowing $h(b\oplus PW_i)$ is impossible. Since S and U çan check whether the transmitted messages during Steps L3 and V1 to V6 are forged or replaced, no attacker can change the transmitted messages. If any malicious attacker wants to mount this attack, either S or U_i will detect it. In addition, because the messages contain T_s and T_u, the transmitted messages for different sessions are different. Consequently, the attacker cannot compute the correct messages or modify the messages which cannot be detected by user or server even if the attacker has collected all the messages in other sessions.

DISCUSSION

R1: Freely choose password: In registration phase of the proposed scheme, a new user can freely choose his password for registering to the remote server. Later, the user can use this pair of (ID_i, $h(b\oplus PW_i)$) to pass the server's authentication procedure.

Table 1: Comparisons of various security attributes

Security attributes	Our	Shieh-Wang	Jung
Use smart card and reader	No	Yes	Yes
Insider's attack	Yes	No	No
Mutual authentication	Yes	Yes	Yes
Securely change password	Yes	No	No
Session key agreement	Yes	Yes	Yes
No time synchronization	Yes	Yes	Yes
Server spoofing resistance	Yes	Yes	No
Perfect forward secrecy	Yes	No	No

R2: Security: It is described in security analysis has some attack resistances, including the privileged insider's attack, the masquerade attack, the stolen-verifier attack, the server spoofing attack, the replay attack, the man-in-the-middle attack and the perfect forward secrecy to confirm the security of the proposed scheme. With the security analysis described earlier, it is ensured that the proposed scheme is robust and secure.

R3: Mutual authentication: In this scheme, allows the user and the server to authenticate each other. As shown in the proposed scheme, S and U_i can authenticate each other in steps V6 and V7 of authentication and key agreement phase, respectively. Accordingly, mutual authentication is ensured in the proposed scheme.

R4: Session key agreement: As shown in authentication and key agreement phase, after finishing mutual authentication, the user U_i and the remote server S can compute the session key $SK = g^{de} \bmod p$. Later, they can use SK to protect the transmitting messages in the following communication.

We will here show several functionality comparisons between the proposed scheme and related schemes in Table 1.

In Table 1, it shows that present schemes can achieve the essential requirements for a secure and efficient remote mutual authentication and key agreement scheme which is described in introduction. It is known that plenty of secret information can be stored in the smart card in advance to reduce the computation costs. Accordingly, the proposed scheme which does not adopt the smart cards to authenticate the user should spend slightly more computation cost than that of Juang's and Shieh-Wang's schemes (Juang, 2004; Shieh and Wang, 2006) for completing the remote authentication procedure. Nonetheless, because the card readers are not obtainable to all and everywhere, authentication protocols adopting the smart cards to authenticate the legitimacy of the user are not applicable to all.

CONCLUSIONS

Over the last decade, most of the proposed password authentication schemes are designed using smart cards, but these schemes do not suffice for users' requirements. Because of the reader of smart cards is not always obtainable to everyone and everywhere. This study proposes a secure and efficient remote mutual authentication scheme without using the smart cards. We have demonstrated that the proposed scheme can satisfy all of the essential security requirements. After the analysis, it is concluded that the new method can be easily practical to the real world.

REFERENCES

Chien, H.Y., J.K. Jan and Y.M. Tseng, 2002. An efficient and practical solution to remote authentication: Smart card. Comput. Secur., 21: 372-375.

Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE Tran. Inform. Theor., 22: 644-654.

Hsu, C.L., 2004. Security of Chien *et al.*'s remote user authentication scheme using smart cards. Comput. Stand. Interfac., 26: 167-169.

Hwang, M.S. and L.H. Li, 2000. A new remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron., 46: 28-30.

Hwang, T., Y. Chen and C.S. Laih, 1990. Non-interactive password authentication without password tables. IEEE Region 10 Conference on Computer and Communication Systems (TENCON'90), Sept. 1990, Hong Kong, pp: 429-431.

Juang, W.S., 2004. Efficient password authenticated key agreement using smart cards. Comput. Secur., 23: 167-173.

Juang, N.S., 2006. Efficient user authentication and key agreement in ubiquitous computing. Proceedings of the 2006 International Conference on Computational Science and its Applications (ICCSA'06), LNCS., 3983, May 8-11, Springer-Verlag Press, German, pp: 396-405.

Ku, W. and S. Chen, 2004. Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron., 50: 204-207.

Ku, W.C., H.M. Chuang and M.J. Tsaur, 2005. Vulnerabilities of Wu-Chieu's improved password authentication scheme using smart cards. IEICE Trans. Fundamentals, E88-A: 3241-3243.

Lamport, L., 1981. Password authentication with insecure communication. Commun. ACM., 24: 770-772.

Lee, J.S., Y.F. Chang and C.C. Chang, 2008. A novel authentication protocol for multi-server architecture without smart cards. Int. J. Innovative Comput. Inf. Control, 4: 1357-1364.

Menezes, A.J., P.C. Oorschot and S.A. Vanstone, 1997. Handbook of Applied Cryptograph. 1st Edn., CRC Press, New York, ISBN: 0-8493-8523-7.

Messerges, T.S., E.A. Dabbish and R.H. Sloan, 2002. Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Comput., 51: 541-552.

Shieh, W.G. and J.M. Wang, 2006. Efficient remote mutual authentication and key agreement. Comput. Secur., 25: 72-77.

Sun, H.M., 2000. An efficient remote use authentication scheme using smart cards. IEEE Trans. Consum. Electron., 46: 958-961.

Sun, H.M. and H.T. Yeh, 2006. Password-based authentication and key distribution protocols with perfect forward secrecy. J. Comput. Sys. Sci., 72: 1002-1011.

Tan, K. and H. Zhu, 1999. Remote password authentication scheme based on cross-product. Comput. Commun., 18: 390-393.

Wang, X.M., W.F. Zhang, J.S. Zhang and M.K. Khan, 2007. Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. Comput. Stand. Interfac., 29: 507-512.

Yang, W.H. and S.P. Shieh, 1999. Password authentication schemes with smart card. Comput. Secur., 18: 727-733.