Kuwait J. Sci. Eng. **38** (2A) pp. 55-68, 2011

Further improvement of Juang *et al.*'s password-authenticated key agreement scheme using smart cards

DEBIAO HE, JIANHUA CHEN AND JIN HU

School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China
Email: hedebiao@163.com

ABSTRACT

Very recently, Sun *et al.* proposed an improved password authenticated key agreement scheme based on Juang *et al.*'s scheme. However, after reviewing their scheme and analyzing its security, we find their scheme is vulnerable to two kinds of attacks, i.e., the offline password guessing attack, and the Denial-of-Service (DoS) attack. The analysis shows that Sun's scheme is insecure for practical application. Then, we propose a further improved scheme to eliminate the security vulnerability. Compared with Juang *et al.*'s scheme and Sun *et al.*'s scheme, our scheme is more secure and more suitable for real-life applications.

Keywords: Authentication; Security; Cryptanalysis; Smart card; Attacks.

INTRODUCTION

Remote authentication is a method to authenticate remote users over insecure communication channels. Password-based authentication schemes have been widely deployed to verify the legitimacy of remote users. Recently, many password authentication schemes using smart cards have been proposed by some researchers (Chen *et al.,* 2009; Chung *et al.,* 2009; Hölbl *et al.,* 2009; Hsiang *et al.,* 2009; Hwang *et al.,* 2009; Juang *et al.,* 2009; Li *et al.,* 2009; Sun *et al.,* 2009; Tsai *et al.,* 2006; Wu *et al.,* 2008; and Xu *et al.,* 2009). In these schemes, the smart-card-oriented remote login authentication scheme is used to authenticate a legitimate user. The smart card contains a microprocessor, which can perform arithmetic operations quickly, an I/O port, a RAM, and a ROM in which some messages are stored. Therefore, there is no need to store a password table or verification table in the server.

Juang *et al.* (2008) proposed a password-authenticated key agreement scheme using smart cards. They broke new ground by pointing out the threat of the smart-card loss. The major contributions of Juang *et al.*'s scheme are to address the threat of the smart-card loss and the use of the elliptic-curve algorithm for reducing the implementation costs. In fact, most of the previous schemes are

insecure under the smart-card-loss assumption. Although the "Juang *et al.*'s" scheme has many benefits, Sun *et al.* (2009) found that it suffers from three weaknesses: 1) the ineffectiveness of the password-changing operation; 2) the session-key problem; and 3) the inefficiency of the double secret keys. Sun *et al.* also proposed an improved scheme to enhance the security. They claimed their scheme could withstand various attacks. However, after reviewing and analyzing its security, we find their scheme is vulnerable to two kinds of attacks, i.e., the offline password guessing attack, and the Denial-of-Service (DoS) attack. Therefore, we propose a further improved scheme to eliminate the security vulnerability. Compared with Juang *et al.*'s scheme and Sun *et al.*'s scheme, our scheme is more secure and suitable for real-life applications.

REVIEW OF SUN'S SCHEME

In order to facilitate future references, frequently used notations are listed below with their descriptions.

- U : a user.
- S : a remote server.
- ID : U's identifier.
- PW : U's password
- K_S : S's long secret key.
- K_{SU} : session key shared between S and U.
- $h(\cdot)$: secure hash function.
- $\oplus$: bitwise XOR operation.
- $||$: concatenation operation

Sun *et al.*'s scheme consists of four phases: a parameter-generation phase, registration phase, authentication phase and password change phase. We describe them as follows.

Parameter-Generation Phase

In this phase, S generates the parameter of the system.

1 - S chooses an elliptic curve E over a finite field F_p. Let $E(F_p)$denote the set of all the point on E.

2 - S chooses a point $G \in E(F_p)$, such that the subgroup generated by G has a large order n.

3 - *S*chooses three hash functions $h(\cdot), h_1(\cdot), h_2(\cdot)$.

4 - *S*publishes the parameter $(p, E, G, n, h(\cdot), h_1(\cdot), h_2(\cdot))$.

Registration phase

In this phase, everyone who wants to register at the server should obtain a smart card. The user U begins his registration at the server S as follows:

1 - U freely chooses his sub-identifier ID_U and sends it to S through a secure channel.

2 - Upon receiving ID_U, *S*checks the validity of ID_U. If ID_U is not valid, *S*rejects the registration. Otherwise, S selects a sub-identifier ID_S and generates the identifier $ID = ID_U||ID_S$ for U. Then S generates a random number r and computes $V = h(ID||K_S) \oplus h(PW)$,$IM = E_{K_S}(ID||r)$, where PW is the initial password select by S.

3 - S then issues the password $PW$and the smart card which contains IM and V to U through a secure channel.

Authentication phase

In this phase, the user U sends a login request message to the server S whenever U wants to access some resources upon*S*. Then the server S verifies the authenticity of the login message requested by the user U.

1 - *U*inserts his smart card into a smart card reader and then inputs his password PW.

2 - U's smart card generates a random number $r_C \in [1, n-1]$, and computes $G_C = r_C \times G$. Then U's smart card sends the message $M_1 = \{IM, G_C\}$ to S.

3 - Upon receiving the message M_1, *S*decrypts the parameter IM by the master key K_S and obtains $ID||r$. Then S checks the validity of ID. If ID is not valid, S aborts the current session. Otherwise, S generates a random number $r_S \in [1, n]$, and computes $G_S = r_S \times G$ and $Q_S = r_S \times G_C$. Then *S*computes $K_{SU} = h_1(h(ID||K_S)||Q_S)$, $M_S = h_2(K_{SU}||G_C||G_S)$ and sends $M_2 = \{M_S, G_S\}$ to the smart card.

4 - Upon receiving the messageM_2, U's smart card computes $V' = V \oplus h(PW)$, $Q_C = r_C \times G_S$ and$K'_{SU} = h_1(V'||Q_C)$. Then U's smart card checks whether the value M_S equals $h_2(K'_{SU}||G_C||G_S)$. If not, the smart card terminates the session. Otherwise, the smart card computes $M_U = h_2(K'_{SU}||G_S)$, then sends the message $M_3 = \{M_U\}$ to S.

5 - Upon receiving the message M_3, S checks whether M_U equals $h_2(K_{SU}||G_S)$.

If not, S stops the session. Otherwise U and S successfully authenticate each other and establish the session key K_{SU}.

Password change phase

1 - U inserts his smart card into the smart-card reader of a terminal, enters the old password PW, and requests to change the password. Next, U enters the new password PW^*.

2 - U's smart card computes $V^* = V \oplus h(PW) \oplus h(PW^*)$, which yields $V^* = h(ID||K_S) \oplus h(PW^*)$, and then replaces V with V^*.

Cryptanalysis of Sun's scheme

Password guessing attack

In password-based authentication schemes, in which the user is allowed to choose his password, the user tends to choose a password that can be easily remembered for his convenience. However, these easy-to-remember passwords are potentially vulnerable to password guessing attacks, in which an adversary can try to guess the user's password and then verify his guess. In general, the password guessing attack can be classified into the on-line password guessing attack and the off-line password guessing attack. On-line password guessing attacks can be easily thwarted by limiting the number of continuous login attempts within a short period, while in the off-line password guessing attack; the server cannot easily notice the attack, since there is no need for the server to participate in the verification.

Although Sun *et al.* (2009) claim that their scheme is secure even when the user's smart card is lost and the parameters in the card are derived, an off-line password guessing attack method will be given here as a counter example.

In practice, several attacks on smart cards were demonstrated. Kocher *et al.* (1999) stated that existing smart cards are vulnerable to attack where the power consumption is monitored (side-channel attack), and in this way the secret keys stored in the smart card can be extracted. Later, Messerges *et al.* (2002) demonstrated another attack, where the secrets stored in a smart card can be acquired by analyzing the leaked information. Further details of these attacks can be found in the appropriate references. Suppose the user's smart card is lost; an attacker$A$can read all the data, including IM and V from the smart card via physical access to the storage medium (Kocher *et al.,*1999; Messerges *et al.,* 2002). Then$A$ can carry out the password guessing attack as follows:

Phase 1

1 - A generates a random number r_A and computes $G_A = r_A \times G$. Then A impersonates U and sends $M_1 = \{IM, G_A\}$ to the server$S$.

2 - Upon receiving the message M_1, Sdecrypts the parameter IM by the master keyK_S and obtains $ID||r$. Then S checks the validity of ID (It is obvious that ID can pass the check of the server). S generates a random number $r_S \in [1, n]$, computes $G_S = r_S \times G$ and $Q_S = r_S \times G_A$. Then Scomputes$K_{SU} = h_1(h(ID||K_S)||Q_S)$, $M_S = h_2(K_{SU}||G_A||G_S)$ and sends $M_2 = \{M_S, G_S\}$ to A.

3 - Upon receiving the message $M_2 = \{M_S, G_S\}$, A stops the session.

Phase 2

1 - A computes $Q'_S = r_A \times G_S = r_S \times G_A = Q_S$.

2 - A selects a password PW' from a uniformly distributed dictionary D.

3 - A computes $h(ID||K_S)' = V \oplus h(PW')$ and $K'_{SU} = h_1(h(ID||K_S)'||Q'_S)$.

4 - A computes $M'_S = h_2(K'_{SU}||G_A||G_S)$ and check if M'_S equals M_S. If M'_S equals M_S, then A find the correct passwords. Otherwise, A repeats steps 1, 2, 3 and 4 until the correct password is found.

Denial-of-service attack on password changing

In password authentication, DoS attacks can cause permanent errors on authentication by introducing unexpected data during the procedures of authentication. The most vulnerable procedure is the password changing phase, since it usually refreshes the data on storage. If an attacker can modify the password, or tamper the message containing password with valid data format, the updated password or its related verification data will then be different from what the user expects. The user can thereby never pass the subsequent authentication.

In Sun *et al.*'s scheme, the password changing phase is performed on the user's terminal with smart cards, i.e., the user can change his password without communicating with the server (Sun *et al.,* 2009). This enhances the security of password changing as no sensitive message need be transmitted over the insecure network. Meanwhile, it relieves the overhead of a server.

However, due to the drawbacks of design, it is still possible to load a DoS attack on password-changing in their scheme. Suppose an attacker temporarily gets access to the user's smart card. He then inserts the card in a terminal device and performs the following operations: he randomly selects two different

passwords PW' and PW'' as the old and the new password, respectively. Then he sends a changing password request to the smart card. As described in the previous section, the smart card will then compute $V^* = V \oplus h(PW') \oplus h(PW'')$, then it replace V with V^*. From then on, Ucan never pass the authentication of the server. This is because in the login phase, U cannot be verified by the server in the third step of the authentication phase.

Sun claimed the users should accept this trouble just as someone who loses his key for the door of his house can get another new lock and key for the door. The attack will not be included if the user can register whenever he wants. But the truth is that it is sometimes impossible for the user to register immediately after the attack through the remote authentication protocol if he is on business and dealing with extremely important issues. In the latest work on these issues, on-line password change protocols (Hwang *et al.,* 2010) and biometrics-based password change protocols (Li *et al.,* 2009) are proposed to withstand the DoS attack. So we think the password change phase of Sun *et al.*'s scheme is not reasonable and the attack must be considered.

The improved scheme

In Sun *et al.*'s scheme, the user U can change passwords freely without the help of the server. But the character makes Sun *et al.*'s scheme vulnerable to the DoS attack. In order to make our scheme withstand the DoS attack and the user able to change the password freely, we apply biometric keys in our scheme, as Li *et al.* (2010) did. Our scheme consists of the parameter-generation phase, the registration phase, the authentication phase, and the password-change phase.

Parameter-generation phase

In this phase, S generates parameter of the system.

1 - S chooses an elliptic curve E over a finite field F_p. Let $E(F_p)$denote the set of all the point on E.

2 - S chooses a point $G \in E(F_p)$, such that the subgroup generated by G has a large order n.

3 - S chooses one hash function $h(\cdot)$.

4 - S publishes the parameter $(p, E, G, n, h(\cdot))$.

Registration Phase

As shown in Fig. 1, in this phase, everyone who wants to register at the server should obtain a smart card. The user U begins his registration at the server S as follows:

1 - *U*inputs his/her personal biometrics B on the specific device and chooses his sub-identifier ID_U and sends them to S through a secure channel.

2 - Upon receiving B and ID_U, S checks the validity of ID_U. If ID_U is not valid, S rejects the registration. Otherwise, *S*selects a sub-identifier ID_S and generates the identifier $ID = ID_U||ID_S$ for U. Then*S*generates a random number r and computes $V_1 = h(ID||K_S) \oplus h(PW)$, $V_2 = h(h(B)||PW)$, $IM = E_{K_S}(ID||r)$, where PW is the initial password select by S.

3 - S issues the password PW and the smart card which contains IM, V_1 and V_2 to U through a secure channel.

4 - U changes the password after receiving the smart card.

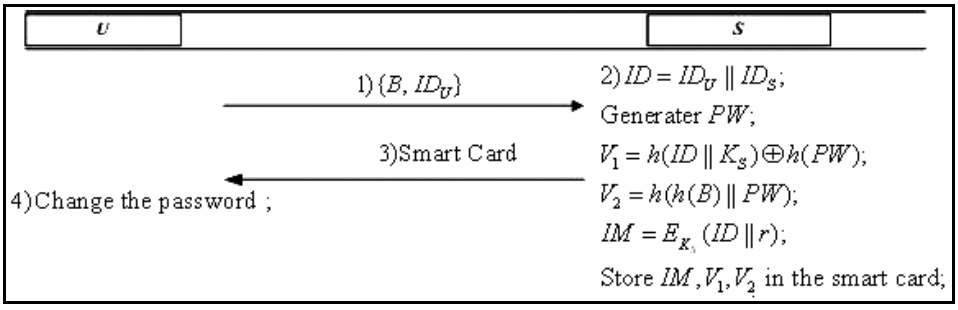

Fig. 1. The registration phase of our scheme.

Authentication phase

As shown in Fig. 2, in this phase, the user U sends a login request message to the server S whenever U wants to access some resources upon S. Then the server S verifies the authenticity of the login message requested by the user U.

1 - U inserts his/her smart card into a smart card reader and inputs his/her password PW and his/her personal biometrics B on the specific device.

2 - U's smart card computes $V_2' = h(h(B)||PW)$ and checks whether V_2' and V_2 are equal. If V_2' and V_2 are not equal, U's smart card reject the request. Otherwise,U's smart card computes $h(ID||K_S)' = V_1 \oplus h(PW)$. Then,$U$'s smart card generates a random number $r_C \in [1, n-1]$, and computes $G_C = r_C \times G$ and $C_1 = G_C \oplus h(h(ID||K_S)')$. U's smart card sends the message$M_1 = \{IM, C_1\}$to$S$.

3 - Upon receiving the message M_1, *S*decrypts the parameter IM by the master key K_S and obtains $ID||r$. Then S checks the validity of ID. If ID is not valid, S aborts the current session. Otherwise, S computes $G_C' = C_1 \oplus h(h(ID||K_S))$, generates a random number $r_S \in [1, n]$, computes

$G_S = r_S \times G, _S = r_S \times G_C$ and $C_2 = G_S \oplus h(h(ID||K_S))$. Then S computes $K_{SC} = h(1||Q_S)$, $C_3 = h(K_{SC}||G'_C||G_S)$ and sends $M_2 = \{C_2, C_3\}$ to the smart card.

4 - Upon receiving the message M_2, U's smart card computes $G'_S = C_2 \oplus h(h(ID||K_S)')$, $Q_C = r_C \cdot G'_S$, $K'_{SC} = h(1||Q_C)$ and $C'_3 = h(K'_{SC}||G_C||G'_S)$. Then U's smart card checks whether the value C'_3 equals C_3. If not, the smart card terminates the session. Otherwise, the server is authenticated. The smart card computes $K_{CS} = h(2||Q_C)$, $C_4 = h(K_{CS}||G_C||G'_S)$ and the session key $K = h(Q_C)$. Last, the smart card sends the message $M_3 = \{C_4\}$ to S.

5 - Upon receiving the message M_3, S computes $K'_{CS} = h(2||Q_S)$, $C'_4 = h(K'_{CS}||G'_C||G_S)$, and checks whether C'_4 equals C_4. If not, S stops the session. Otherwise, U is authenticated. Then S computes the session key $K = h(Q_S)$.

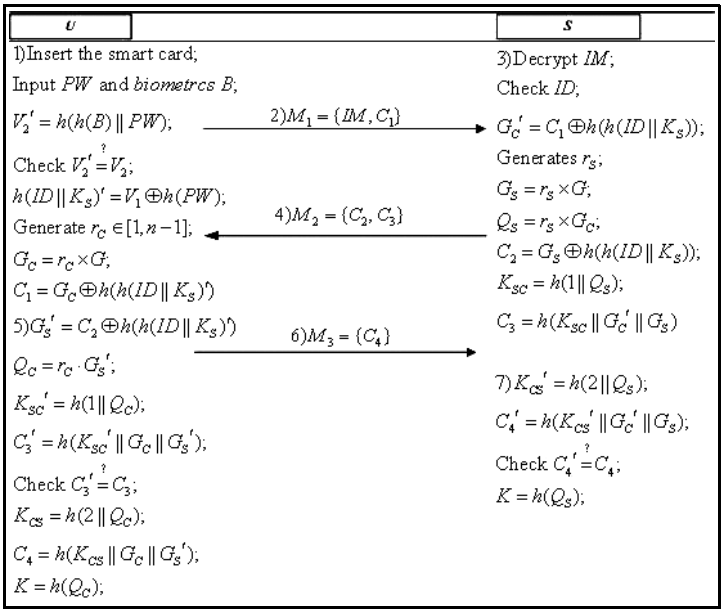

Fig. 2. The authentication phase.

Password change phase

1 - U inserts his/her smart card into a smart card reader and inputs his/her password PW and his/her personal biometrics B on the specific device.

2 - U's smart card computes $V'_2 = h(h(B)||PW)$and checks whether V'_2 equals V_2. If V'_2 doesn't equal V_2, U's smart card reject the request. Otherwise, Uinput the new password PW^*.

3 - U's smart card computes $V_2^* = V_2 \oplus h(PW) \oplus h(PW^*)$, which yields $V_2^* = h(ID||K_S) \oplus h(PW^*)$, and then replaces V_2 with V_2^*.

Security analysis of our scheme

The security of our scheme is based on elliptic curve discrete logarithm problem (*ECDLP*) and the secure one-way hash functions. We will show that our improvement not only can provide mutual authentication, perfect forward and backward secrecy and key freshness, but also can resist the following attacks: replay attack, off-line password guessing attack, insider attack, man-in-the-middle attack, on-line password guessing attack, and DoS attack. We will use Chen's method (Chen *et al.,* 2009) to analyze the security of our scheme.

Mutual authentication

Mutual authentication means that U and S is authenticated to each other within the same protocol (Chen *et al.,* 2009). In our scheme, U and S can authenticate each other by checking the validity of C_3 and C_4 separately. Then the mutual authentication between them is achieved.

Perfect forward and backward secrecy

Perfect forward and backward secrecy means that if an intruder gets the session key, he cannot reconstruct any previous or subsequent session keys (Chen *et al.,* 2009). In our improvements, a compromised password PW cannot be used to reconstruct any previous or subsequent session keys, using the Diffie-Hellman key agreement scheme. If an intruder gets PW in our scheme, he/she may get $G_C = r_C \times G$ and $G_S = r_S \times G$, but he/she cannot deduce $K = h(r_C \times r_S \times G)$ without the knowledge of the two random numbers r_C and r_S. Therefore, our scheme can provide perfect forward and backward secrecy.

Key freshness

Key freshness means that the key used in each session is different from the ones used in other sessions (Chen *et al.,* 2009). Since each party picks his random nonce secretly when computing the session key in our protocol, it can be easily seen that the freshness of the used session keys in our scheme is guaranteed.

Preventing the replay attack

Replay attack means that a legal peer's transmission message is intercepted and replayed by an adversary for fooling another legal peer into regarding him as authentic (Chen *et al.,* 2009). However, the fresh nonces chosen at each protocol run are used to avoid such replay attacks in our improvements.

Preventing the off-line password guessing attack

An off-line password guessing attack means that a passive attacker intercepts the communication line between a legal user and the server, and tries to guess the user's password off line (Chen *et al.,* 2009). The attack A may intercept$M_1 = \{IM, C_1\}$, $M_2 = \{C_2, C_3\}$, $M_3 = \{C_4\}$. A may get IM, as V_1 and V_2 stored the smart card. Then A could guess a password PW'. But A can't verify the correctness of PW', since he/she will face the *ECDLP*.

Preventing the insider attack

Insider attack means that a legal user D can impersonate another legal user U to gain the service of server S (Chen *et al.,* 2009). Assume that D wants to impersonate U to login to S. However, without the knowledge of U's password, he/she can not deduce $h(ID||K_S)$, and consequently be authenticated by$S$. Therefore, our scheme can withstand the insider attack.

Preventing man-in-the-middle attack

Man-in-the-middle attack means that an active attacker intercepts the communication line between a legal user and the server and uses some means to successfully masquerade as both the server to the user and the user to the server. Then, the user will believe that he is talking to the intended server, and vice versa (Chen *et al.,* 2009). In our scheme, the attack A cannot generate the valid C_3 and C_4 without the value of K_S. Then if A forge C_3 or C_4, U and S will find the attack through checking the correctness of C_3 or C_4, separately.

Preventing the on-line password guessing attack

Suffering on-line password guessing attacks means that an attacker can successfully guess a legal user's password on line (Chen *et al.,* 2009). Since our scheme has the mutual authentication function, only the user with the right password can pass the authentication of the server. Therefore, any attempt to launch a password guessing attack will be detected by the server. Moreover, we can set both improvements to tolerate some times of wrong password logins, e.g., three times. If the number of wrong login times is reached, the system would reject the login request. Under such a setting, our scheme can resist the on-line password guessing attack.

Preventing smart-card-lost attack

The smart-card-lost attack means an attacker can launch various attacks when he/she gets a legal user's smart card (Chen *et al.,* 2009). In the following, we

discuss two of the most common attacks launched under such a situation: off-line password guessing attack and impersonation attack.

1 - Suppose U's smart card is lost and obtained by A. A can read IM, V_1 and V_2 in U's smart card. Then A could guess a passwordPW'. But A cannot verify the correctness of PW', since he/she will face the $ECDLP$.

2 - If A impersonates U to login in the server, he/she can not construct the valid message C_4, since he/she does not know the value $h(ID||K_S)$. Then the impersonation attack will be found by the server.

Preventing DoS attack after password changing

Suffering a DoS attack means that if an attacker temporarily gets access to the user's smart card and successfully guesses the password, then he can perform the password change phase to replace the old password with his new one. This would result in making the legal user's password invalid, and thereafter the server will deny any service to the legal user (Chen *et al.,* 2009). However, our scheme checks the correctness of personal biometrics B and the old password. That is, even when an attacker can temporarily get access to the user's smart card, he/she can't successfully change the password. Consequently, our method can resist the DoS attack.

Comparison with related works

In this section, we will compare the performance and functionality of our scheme with that of related works.

Table 1 shows the performance comparison results of the authentication phase. Since the other phases just need to be executed once, it is not necessary to compare them. The following notations are used in Table 2. The names of the computation operations have been abbreviated to save space: H denotes the cryptographic hash computation, E denotes the symmetric encryption or decryption computation, and M denotes the scalar multiplication computation over the elliptic curve.

Table 1. Performance comparisons in the authentication phase.

	Juang *et al.*'s scheme	**Sun *et al.*'s scheme**	**Our scheme**
Smart Card	2M + 4H + 1E	2M + 4H	2M + 6H
Server	1M + 4H + 2E	2M + 4H + 1E	2M + 6H + 1E

Then, we compare the functionality of our scheme with Juang's scheme and Sun's scheme. Table 2 shows the functionality comparison results.

Table 2. Functionality comparisons.

	Juang *et al.*'s scheme	**Sun *et al.*'s scheme**	**Our scheme**
The password is changed by the user freely	No	Yes	Yes
provide the explicit key confirmation	No	Yes	Yes
Need of double secret keys	Yes	No	No
Mutual authentication	Yes	Yes	Yes
Password guessing attack resistance	No	No	Yes
DoS attacks resistance	No	No	Yes
No verification table	Yes	Yes	Yes

From the comparison results in Table 1 and Table 2, we know our scheme needs one or two hash function operations or symmetric encryption or decryption operations. But the cost of hash function operation and symmetric encryption or decryption operation may be ignored when compared with the cost of scalar multiplication computation over the elliptic curve. Then our scheme has nearly the same performance with "Juang *et al.'s*" *scheme and Sun et al.*'s scheme. In addition, ours can withstand both of the DoS and the password guessing attacks. In this way, our scheme is more practical.

CONCLUSION

In this paper, we pointed out that Sun *et al.*'s scheme is vulnerable to two kinds of attacks, i.e., the offline password guessing attack, and the Denial-of-Service (DoS) attack. In order to overcome the weaknesses, we propose an improved scheme. The analysis and comparison show our scheme is more secure than Juang *et al.*'s scheme and Sun *et al.*'s scheme.

ACKNOWLEDGEMENT

The authors thank the anonymous reviewers and Prof. Fawzia Al-Ruwaih for their valuable comments. This research was supported by the Fundamental Research Funds for the Central Universities under Grants 201275786 and the Open Funds of State Key Laboratory of Information Security.

REFERENCES

Chen, Y., Chou, J. & Huang, C. 2009. Improvements on two password-based authentication protocols. http://eprint.iacr.org/2009/561.pdf.

Chung, H. R., Ku, W.C. & Tsaur, M. J. 2009. Weaknesses and improvement of Wang *et al.*'s remote user password authentication scheme for resource-limited environments. Computer Standards & Interfaces **31**(4):863-868.

Hölbl, M. & Welzer, T. 2009. Two improved two-party identity-based authenticated key agreement protocols. Computer Standards & Interfaces **31**(6):1056-1060.

Hsiang, H. C. & Shih, W. K. 2009. Weaknesses and improvements of the Yoon--Ryu--Yoo remote user authentication scheme using smart cards. Computer Communications **32**(4):649-652.

Hwang, M. S., Chong, S. K. & Chen, T. Y. 2009. DoS-resistant ID-based password authentication scheme using smart cards. Journal of Systems and Software **83**(1): 163-172.

Juang, W. S., Chen, S. T. & Liaw, H. T. 2008. Robust and efficient password authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics **55**(6): 2551--2556.

Kocher, P., Jaffe, J. & Jun, B. 1999. Differential power analysis. Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, 388-397.

Li, C. T. & Hwang, M. S. 2010. An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications **33**(1):1-5.

Messerges, T.S., Dabbish, E.A. & Sloan, R.H. 2002. Examining smart card security under the threat of power analysis attacks. IEEE Transactions on Computers **51**(5):541--552.

Sun, D.Z., Huai, J.-P. & Sun, J.Z. 2009. Improvements of Juang *et al.*'s Password-Authenticated Key Agreement Scheme Using Smart Cards. IEEE Transactions on Industrial Electronics **56**(6):2284-2291.

Tsai, C., Lee, C. & Hwang, M. 2006. Password authentication schemes: Current status and key issues. International Journal of Network Security **3**(2):101-115.

Wu, S. & Zhu, Y. 2008. Proof of forward security for password-based authenticated key exchange. International Journal of Network Security **7**(3):335-341.

Xu, J., Zhu, W. T. & Feng, D. G. 2009. An improved smart card-based password authentication scheme with provable security. Computer Standards & Interfaces **31**(4):723-728.

Submitted : *4/2/2010*
Revised : *1/11/2010*
Accepted : *5/11/2010*

التطوير المتزايد لمخطط اتفاقية لمفتاح مصادق بكلمة السر لجيانغ وآخرون

هي ديبيو ، تشن جيان هوا و هو جين

كلية الرياضيات والإحصاء – جامعة وهان – وهوبي 430072 – الصين

خلاصة

حديثاً اقترح الباحث سون وآخرون مخطط اتفاقية لمفتاح مصادق بكلمة السر بناءًا على مخطط جيانغ وغيره. ومع ذلك، وبعد استعراض مخططهم وتحليل أمنه، نجد أن مخططهم معرضاً لنوعين من الهجومات، أولاً هجومية تخمين كلمة السر خارج الشبكة، ثانياً هجوم لرفض الخدمة (DOS). يظهر التحليل أن مخطط سون وآخرون ليس آمناً في التطبيقات العملية. لذلك في هذا البحث نقترح مخططاً مطوراً للقضاء على ضعف الأمن. بالمقارنة مع مخطط الباحث جيانغ وآخرون ومخطط الباحث سون وآخرون، يكون مخططنا أكثر أمناً. نعتقد أن مخططنا المطور هو أكثر ملاءمة للتطبيقات في الحياة العملية.