Breaking a Robust Remote User Authentication Scheme using Smart Cards

Ding Wang[1, 2] , Chun-guang Ma[1], Sen-dong Zhao[1], Chang-li Zhou[1]

[1] College of Computer Science and Technology, Harbin Engineering University
145 Nantong Street, Harbin City 150001, China
wangdingg@mail.nankai.edu.cn
[2] Automobile Management Institute of PLA, Bengbu City 233011, China

Abstract. Understanding security failures of cryptographic protocols is the key to both patching existing protocols and designing future schemes. Recently, Yeh et al. showed that Hsiang and Shih's password-based remote user authentication scheme is vulnerable to various attacks if the smart card is non-tamper resistant, and proposed an improved version which was claimed to be efficient and secure. In this study, however, we find that, although Yeh et al.'s scheme possesses many attractive features, it still cannot achieve the claimed security goals, and we report its following flaws: (1) It cannot withstand offline password guessing attack and key-compromise impersonation attack under their non-tamper resistance assumption of the smart card; (2) It fails to provide user anonymity and forward secrecy; (3) It has some other minor defects. The proposed cryptanalysis discourages any use of the scheme under investigation in practice. Remarkably, rationales for the security analysis of password-based authentication schemes using smart cards are discussed in detail.

Keywords: Cryptanalysis, Authentication protocol, Offline password guessing attack, Smart card, Forward secrecy.

1 Introduction

With the development of distributed computer networks, it is easy for user terminals to share information and computing power with hosts [1,2]. The distributed locations of service providers make it efficient and convenient for subscribers to access the resources, and it is of great concern to protect the systems and the users' privacy and security from malicious adversaries. Accordingly, user authentication becomes an essential security mechanism for remote systems to assure one communicating party of the legitimacy of the corresponding party by acquisition of corroborative evidence. Among numerous methods for user authentication, password based authentication with smart cards is one of the most promising techniques and has been widely adopted over insecure networks to validate the legitimacy of users.

In 1981, Lamport [3] introduced the first password authentication scheme to authenticate a remote user over an insecure channel. This seminal scheme was later refined and used in a number of applications, notably Haller's famous S/KEY one-time password system [4]. Later on, Chang and Wu [5] introduced the smart cards

James J. Park et al. (Eds.): NPC 2012, LNCS 7513, pp. 110–118, 2012.
© IFIP International Federation for Information Processing 2012

into remote user authentication schemes, since then there have been many smart card based password authentication schemes proposed [6-10]. In such schemes, the user is equipped with a smart card and a password as identification verifiers. When the user wants to login to the server, she provides the card with her password, which is used to construct a login request that is sent to the server. Upon receiving the request, the server authenticates these messages and provides the desired service if the verifiers are found valid. If mutual authentication occurs, the client is also convinced that the corresponding server is authentic. More admired schemes also achieve session key agreement for securing the subsequent data communications.

The common adversary model to evaluate the security of authentication protocols using smart cards assumes an attacker with full control over the communication channel between the user and the remote server [7,10,11]. Accordingly, all the messages exchanged can be blocked, intercepted, deleted, or modified by the attacker, and the attacker can also insert his/her own fabricated messages. Secondly, protocols must assume that the attacker can temporarily get access to the legitimate user's smart card, which is reasonable in practice. What's more, since recent research results have shown that the secret data stored in the common smart card could be extracted by some means, such as monitoring the power consumption [12,13] or analyzing the leaked information [14], the smart card should be assumed to be non-tamper resistant, i.e., the secret information stored in the smart card can be revealed.

As mentioned in [10,15], a sound password authentication scheme should be able to withstand a number of sophisticated distinct types of attacks, such as replay attack, password guessing attack, parallel session attack, denial of service attack, stolen verifier attack, and user/server impersonation attack. As resistance to these passive and active attacks is a basic security requirement for authentication protocols, the following desirable attributes are also of great importance in the case of an authentication scheme with session key establishment [16,17]:

i. Resistance to known key attack. A protocol still achieves its security goal in the face of an adversary who has learned some previous session keys.
ii. Provision of forward secrecy. Even if long-term private key of one or more entities are compromised, the secrecy of previous session keys is not affected.
iii. Resistance to unknown key-share attack. The entity i cannot be coerced into sharing a key with entity j without i's knowledge, i.e., when i believes the key is shared with some other entity k, where $k \neq j$.
iv. Resistance to key-compromise impersonation attack. It is desirable that the leakage of entity i's long term private key does not enable an adversary to impersonate other entities to i.

In 2009, Hsiang and Shih [8] showed that Yoon et al.'s scheme [6] is susceptible to user impersonation attack, offline password guessing attack and parallel session attack. To overcome these defects, Hsiang and Shih presented an enhanced version. Later on, Sood et al. [9] showed that Hsiang and Shih's scheme still suffers from offline password guessing attack and user impersonation attack, and user anonymity is not preserved. More recently, Yeh et al. [18] identified that, besides the security flaws found by Sood et al., Hsiang and Shih's scheme is also prone to undetectable online password guessing attack. Consequently, Yeh et al. proposed a further improved version to eliminate the aforementioned security flaws. In this paper, however, we

will demonstrate that Yeh et al.'s scheme is still vulnerable to the offline password guessing attack and key-compromise impersonation attack. Moreover, their scheme fails to provide the property of forward secrecy and user anonymity.

The remainder of this paper is organized as follows: in Section 2, we briefly review Yeh et al.'s authentication scheme. Section 3 describes the weaknesses of Yeh et al.'s scheme. Finally, we conclude this paper in the last section.

2 Review of Yeh et al.'s scheme

In this section, we briefly review the first scheme, i.e. the improvement on Hsiang and Shih's scheme, proposed by Yeh et al. in [18]. Their scheme, summarized in Fig.1, consists of four phases, namely, the registration phase, the login phase, the verification phase and password update phase. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 1.

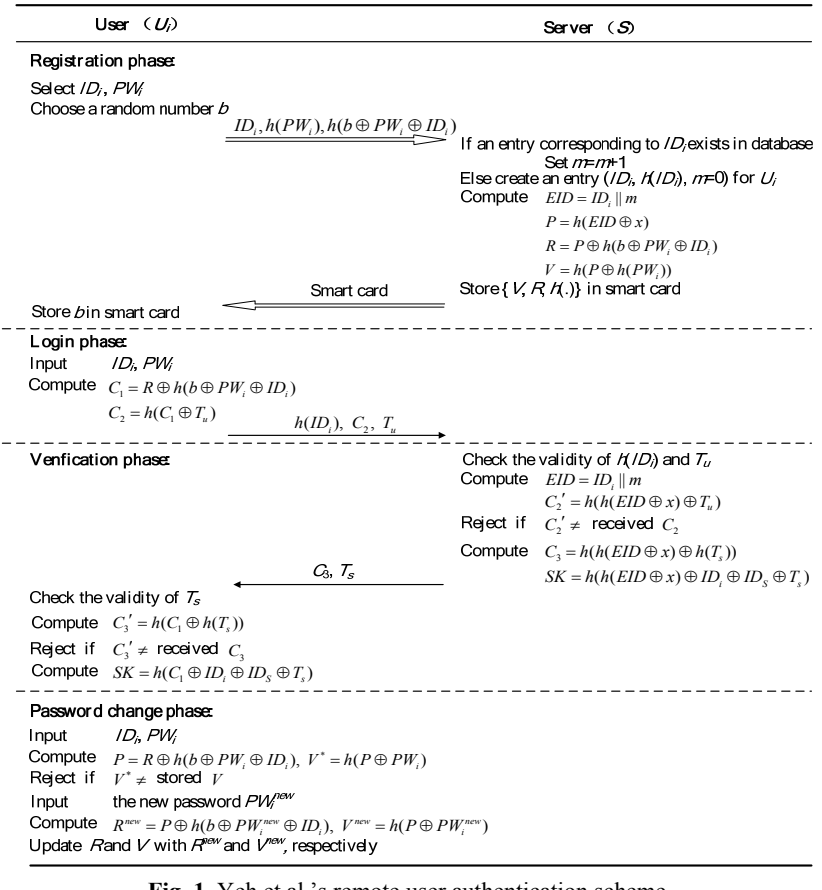

Fig. 1. Yeh et al.'s remote user authentication scheme

Table 1. Notations

Symbol	Description
U_i	i^{th} user
S	remote server
ID_i	identity of user U_i
PW_i	password of user U_i
x	the secret key of remote server S
$h(\cdot)$	collision free one-way hash function
$\oplus$	the bitwise XOR operation
$\|\|$	the string concatenation operation
$A \Rightarrow B : M$	message M is transferred through a secure channel from A to B
$A \rightarrow B : M$	message M is transferred through a common channel from A to B

2.1 Registration phase

The registration phase involves the following operations:

Step R1. U_i chooses his/her identity ID_i, password PW_i, and a random number b.

Step R2. $U_i \Rightarrow S$: $\{ID_i, h(PW_i), h(b \oplus PW_i \oplus ID_i)\}$.

Step R3. On receiving the registration message from U_i, the server S creates a new entry $(h(ID_i), ID_i, m)$ with the value $m = 0$ for U_i in the backend database, or sets $m=m+1$ in the existing entry. Then, S computes $EID=ID \| m$, $P= h(EID \oplus x)$, $R=P \oplus h(b \oplus PW_i \oplus ID_i)$ and $V=h(P \oplus h(PW_i))$.

Step R4. $S \Rightarrow U_i$: A smart card containing security parameters $\{V, R, h(\cdot)\}$.

Step R5. U_i enters b into his/her smart card.

2.2 Login phase

When U_i wants to login to S, the following operations will be performed:

Step L1. U_i inserts his/her smart card into the card reader, and inputs ID_i and PW_i.

Step L2. The smart card computes $C_1= R \oplus h(b \oplus PW_i \oplus ID_i)$ and $C_2=h(C_1 \oplus T_u)$, where T_u is the current timestamp on user side.

Step L3. $U_i \rightarrow S$: $\{C_2, h(ID_i), T_u\}$.

2.3 Verification phase

After receiving the login request from user U_i, S performs the following operations:

Step V1. S first checks the validity of $h(ID_i)$ and T_u, computes $EID=ID_i \| m$ and $C_2' =h(h(EID \oplus x) \oplus T_u)$, and then compares the computed C_2' with the received C_2. If they are equal, S computes $C_3=h(h(EID \oplus x) \oplus h(T_s))$ and session key $SK=h(h(EID \oplus x) \oplus ID_i \oplus ID_S \oplus T_s)$, where ID_S denotes the identity of S. Otherwise, S rejects the request.

Step V2. $S \rightarrow U_i$: $\{C_3, T_s\}$.

Step V3. Upon receiving the reply message, U_i checks the validity of T_u. If the verification fails, U_i terminates the session. Then, U_i computes $C_3' = h(C_1 \oplus$

$h(T_s)$), and then compares the computed C_3' with the received C_3. If the verification holds, U_i computes $SK=h(h(EID \oplus x) \oplus ID_i \oplus ID_S \oplus T_s)$. Otherwise, U_i terminates the session.

Step V4. After authenticating each other, U_i and S use the same session key SK to secure ensuing data communications.

2.4 Password change phase

When U_i wants to change the old password PW_i to the new password PW_i^{new}, the following operations will be involved:

Step P1. U_i insert his/her own smart card into card reader, keys ID_i and PW_i.

Step P2. The smart card computes $P=R \oplus h(b \oplus PW_i \oplus ID_i)$ and $V^*=h(P \oplus h(PW_i))$, and checks whether V^* equals V. If the verification fails, smart card rejects.

Step P3. U_i keys his/her new password PW_i^{new}.

Step P4. The smart card computes $R^{new}=P \oplus h(b \oplus PW_i^{new} \oplus ID_i)$ and $V^{new}=h(P \oplus h(PW_i^{new}))$, and updates R and V with the new R^{new} and V^{new} respectively.

3 Cryptanalysis of Yeh et al.'s scheme

There are two assumptions explicitly made in Yeh et al.'s scheme [18]:

(i) The adversary $\mathbb{A}$ has total control over the communication channel between the user U and the remote server S. In other words, the adversary can insert, delete, alter, or intercept any messages transmitted in the channel.

(ii) The secret parameters stored in the smart card could be extracted out once a legitimate user's card is somehow (e.g. stolen or picked up) obtained by $\mathbb{A}$.

Note that the above two assumptions, which are also made in the latest works [7,9,10], are indeed reasonable: (1) Assumption *i* is accordant with the common adversary model introduced in Section 1; and (2) Assumption *ii* is also practical in consideration of the state-of-art side-channel attack techniques [12-14]. In the following discussions of the security flaws of Yeh et al.'s scheme, based on the above two assumptions, we assume that $\mathbb{A}$ can extract the secret values $\{V, R, b\}$ stored in the legitimate user's smart card, and the attacker can also intercept or block the login request message $\{C_2, h(ID_i), T_u\}$ from U_i and the reply message $\{C_3, T_s\}$ from S.

As described in Yeh et al.'s scheme, mainly two countermeasures are employed to remedy the identified flaws in Hsiang and Shih's scheme: (1) user's ID is concealed by use of a non-invertible hash function to double the difficulty of mounting an offline password guessing attack; (2) a session key is agreed to resist against server impersonation attack. However, as will be shown in the following, the first countermeasure is not effective enough, and the later one lacks key security considerations yet.

3.1 Offline password guessing attack

A remote user authentication scheme vulnerable to the offline password guessing

attack must satisfy the two conditions: the user's password is weak, and there exists a piece of password-related information used as a comparison target for password guessing. In Yeh et al.'s scheme, a user is allowed to choose his/her own password *PW* at will during the registration and password change phases; the user usually tends to select a password, e.g., his phone number or birthday, which is easily remembered for his convenience. Hence, these easy-to-remember passwords, called weak passwords [19], have low entropy and thus are potentially vulnerable to offline password guessing attack. Inevitably, user's *ID*, chose by the user in the same way with *PW* as described in the scheme, is exposed to the same threat.

Let us consider the following scenarios. In case a legitimate user U_i's smart card is stolen by an adversary $\mathbb{A}$, and the stored secret values such as *R, V* and *b* can be extracted. With a previously eavesdropped message $\{C_2, h(ID_i), T_u\}$, $\mathbb{A}$ can acquire U_i's password PW_i by performing the following malicious attack procedure:

Step 1. Guesses all possible values ID_i^* of U_i's identity, and compares the value of $h(ID_i^*)$ with $h(ID_i)$. If the computed $h(ID_i^*)$ equals the intercepted $h(ID_i)$, it implies $ID_i^* = ID_i$ and U_i's identity is found, and proceeds to Step 2.
Step 2. Guesses the value of PW_i to be PW_i^* from the password space $\mathbb{D}$.
Step 3. Computes $C_1^* = R \oplus h(b \oplus PW_i^* \oplus ID_i)$, where the value of *b* and *R* are revealed from the smart card and the value of ID_i is obtained through Step 1.
Step 4. Computes $C_2^* = h(C_1^* \oplus T_u)$, as T_u is previously intercepted.
Step 5. Verifies the correctness of PW_i^* by checking if C_2^* equals the intercepted C_2.
Step 6. Repeats Steps 2, 3, 4 and 5 of this phase until the correct value of PW_i is found.

Since the size of password dictionary, i.e. $|\mathbb{D}|$, often is very limited, the above attack procedure can be completed in polynomial time. Halevi and Krawczyk [20] have proved that, under the Dolev-Yao adversary model [11], no password protocol can be free from offline password guessing attack if the public-key techniques are not employed. Therefore, the feasible solution is to reduce the success probability of this attack. Following this principle, Yeh et al.'s scheme thwarts this threat to nearly half success probability as compared to that of Hsiang and Shih's original scheme, which can be easily confirmed from the above attack procedure.

However, we have found that, some minor technical modifications to Yeh et al.'s scheme can quadratically but not linearly reduce the success possibility of this attack. Due to space constraints, we do not give the complete remedy here, and recommend readers to refer the literature [10] for details. The idea of the remedy is not particularly complicated: whenever PW_i appears, it is concatenated with the identity ID_i, while ID_i is concealed in dynamic-ID(s). Therefore, the mechanism employed by Yeh et al. to resist against offline password guessing attack is not effective enough as minor revision may thwart this threat to a more desirable extent.

3.2 Key-compromise impersonation attack

In the case of key-compromise impersonation, the question is whether the knowledge of a communicating party *A*'s private key allows a malicious attacker $\mathbb{A}$ not only to impersonate *A* to others but also to impersonate other uncorrupted parties to *A*. Schemes that prevent this kind of reverse impersonation are said to withstand key-

compromise impersonation attack.

Suppose the long-term secret key x of the server S is leaked out by accident or intentionally stolen by the adversary A. Once the value of x is obtained, with previously intercepted $h(ID_i)$ transmitted in U_i's authentication process, A can impersonate the legitimate user U_i through the following method:

Step 1. Guesses U_i's identity to be ID_i^* from a dictionary of all possible 'weak' identities, and verify the guess by checking whether $h(ID_i^*)$ equals $h(ID_i)$.
Step 2. Assumes $m = 0$, where m denotes the re-registration times of U_i.
Step 3. Computes $EID=ID_i \| m$ and $P=h(EID \oplus x)$, where ID_i is derived through Step 1 and x has also been learned.
Step 4. Let $C_1= P$ and $C_2=h(C_1 \oplus T_m)$, where T_m is the current timestamp.
Step 5. Sends the fabricated login request $\{C_2, h(ID_i), T_m\}$ to server S.
Step 6. Waits for the reply for a reasonable interval. If no response comes, set $m = m + 1$ and goes back to Step 3, else proceeds to the next step.
Step 7. Receives the reply $\{C_3, T_s\}$ from server S and computes the session key $SK= h(h(EID \oplus x) \oplus ID_i \oplus ID_S \oplus T_s)$.

Since the value of m, i.e. the re-registration times of U_i, should be very limited in common practice, at most a few dozen, the iteration of the above procedure will come to an end very quickly. The rest of the question is whether Step 1 can be completed in polynomial time. In Yeh et al.'s scheme, a user is allowed to choose his/her own identity ID at will during the registration and password change phases; the user usually tends to select an identity that is human-memorable short strings but not high-entropy keys. In other words, they are chosen from the dictionaries of small size. Therefore, the above attack is feasible.

3.3 Failure to achieve forward secrecy

As with resistance to key-compromise impersonation attack, the property of forward secrecy is also concerned with limiting the effects of eventual failures, in case the disclosure of server's long-term private keys. Let us consider the following scenarios. Suppose the server S's long-term private key x is leaked out by accident or intentionally stolen by an adversary A. Once x is obtained, with previously intercepted messages $\{h(ID_i), T_s^j, C_3^j\}$ transmitted during any one of U_i's authentication process (without loss of generality, assume it is U_i's jth authentication process), A can derive the session key SK^j of S and U_i's jth encrypted communication through the following method:

Step 1. Guesses U_i's identity to be ID_i^* from a dictionary of all possible 'weak' identities, and verify the guess by checking whether $h(ID_i^*)$ equals $h(ID_i)$.
Step 2. Assumes $m = 0$, where m denotes the re-registration times of U_i.
Step 3. Computes $EID=ID_i \| m$ and $C_3^{j*} = h(h(EID \oplus x) \oplus h(T_s^j))$, where ID_i is derived through Step 1 and x has also been learned.
Step 4. Compares C_3^{j*} with the intercepted C_3^j, this equivalence implies the correct value of m is found. Otherwise, sets $m = m + 1$ and goes back to Step 3.
Step 5. Computes $SK^j = h(h(EID \oplus x) \oplus ID_i \oplus ID_S \oplus T_s)$.

The computation complexity of Step 1, Step 3 and Step 4 has been analyzed in Section 3.2, and it's evident that the whole procedure described above can be completed in polynomial time. Once the session key SK^j is obtained, the entire jth session will become completely insecure. Consequently, the property of forward secrecy is not provided in Yeh et al.'s scheme, while the provision of forward secrecy is a basic requirement for a secure key agreement scheme.

3.4 No provision of user anonymity

In Yeh et al.'s scheme, the user U_i's identity ID_i is wrapped up in hashing, i.e. $h(ID_i)$, which is static and specific to user U_i in all the transaction sessions, an adversary can easily obtain the hashed identity of this communicating client once the login messages were eavesdropped, and hence, different login request messages belonging to the same user can be traced out and may be interlinked to derive some secret information related to the user. Furthermore, U_i's identity ID_i may be derived from $h(ID_i)$ through the method introduced in Section 3.1. Hence, user anonymity is not preserved.

3.5 Some practical pitfalls

In the registration phase, U_i's password PW_i is just submitted in a hashed form to S, and thus it can be easily derived by S. If U_i uses this PW_i to access several servers for his/her convenience, the insider of S can impersonate U_i to access other servers. Hence, it's an insecure factor to commit just a hashed password to the server.

Another pitfall in Yeh's scheme is the slow wrong password detection [15]. If U_i inputs a wrong password by mistake, this wrong password will be only detected by the remote system in the verification phase. Therefore, their scheme is slow to detect the user's wrongly input password.

4 Conclusion

Smart card-based password authentication technology has been widely deployed in various kinds of security-critical applications, and careful security considerations should be taken into account when designing such schemes. In this paper, we have shown that Yeh et al.'s scheme still suffers from the offline password guessing attack and key-compromise impersonation attack. In addition, their scheme fails to provide the property of forward secrecy and user anonymity. Some other minor defects have also been found. In conclusion, although Yeh et al.'s scheme has many attractive features, it, in fact, does not provide all of the security properties that they claimed and only radical revisions of the protocol can possibly eliminate the identified defects. Therefore, the scheme under study is not recommended for practical application.

Acknowledgements. This research was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042.

References

1. Vicente, A.G., Munoz, I.B., Galilea, J.L.L., del Toro, P.A.R.: Remote automation laboratory using a cluster of virtual machines. IEEE Transactions on Industrial Electronics 57(10), 3276–3283 (2010)
2. Barolli, L., Xhafa, F.: JXTA-OVERLAY: A P2P platform for distributed, collaborative and ubiquitous computing. IEEE Transactions on Industrial Electronics 58(6), 2163–2172 (2010)
3. Lamport L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770 –772 (1981)
4. Hailer, NM.: The S/Key One-time Password System. In: Proceedings of the Symposium on Network and Distributed System Security, pp.151–158. IEEE press, New York (1994)
5. Chang, C.C., Wu, T.C.: Remote password authentication with smart cards. IEE Proceedings-E 138(3), 165–168 (1993)
6. Yoon, E.J., Ryu, E.K., Yoo, K.Y.: Further improvement of an efficient password based remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 50(2), 612–614 (2004)
7. Yang, G., Wong, D.S., Wang, H., Deng. X.: Two-factor mutual authentication based on smart cards and password. Journal of Computer and System Sciences 74(7), 1160–1172 (2008)
8. Hsiang, H.C., Shih, W.K.: Weaknesses and improvements of the Yoon-Ryu-Yoo remote user authentication scheme using smart cards. Computer Communications 32(4), 649–652 (2009)
9. Sood, S.K., Sarje, A.K., Singh, K.: An improvement of Hsiang-Shih's authentication scheme using smart cards. In: Proceedings of ICWET 2010, pp. 19–25. ACM Press, New York (2010)
10. Ma, C.G., Wang, D., Zhang, Q.M.: Cryptanalysis and improvement of sood et al.'s dynamic id-based authentication scheme. In: Ramanujam, R., Ramaswamy, S. (eds.) ICDCIT 2012, LNCS, vol. 7154, pp. 141–152. Springer, Heidelberg (2012)
11. Dolev, D., Yao, A. C.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2):198–208 (1983)
12. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
13. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Transactions on Computers 51(5), 541–552 (2002)
14. Mangard, S., Oswald, E., Standaert, F.X.: One for all-all for one: unifying standard differential power analysis attacks. IET Information Security 5(2), 100–110 (2011)
15. Tsai, C., Lee, C., Hwang, M.: Password authentication schemes: current status and key issues. International Journal of Network Security 3(2), 101–115 (2006)
16. Blake-Wilson, S., Johnson, D., Menezes, A.: Key agreement protocols and their security analysis. In: Darnell, M.J. (ed.) Cryptography and Coding 1997. LNCS, vol. 1355, pp. 30–45. Springer, Heidelberg (1997)
17. Krawczyk, H.: HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: Shoup, V.(ed) CRYPTO 2005. LNCS, vol. 3621, pp. 546–566. Springer, Heidelberg (2005)
18. Yeh, K.H., Su, C.H., Lo, N.W.: Two robust remote user authentication protocols using smart cards. Journal of Systems and Software 83(12), 2556–2565 (2010)
19. Klein, D.V.: Foiling the Cracker: A Survey of, and Improvements to, Password Security. In: 2nd USENIX Security Workshop, pp.5–14. USENIX Association, Portland (1990)
20. Halevi, S., Krawczyk, H.: Public-key cryptography and password protocols. ACM Transactions on Information and System Security 2(3), 230–268 (1999)