On the Usability of Two-Factor Authentication

Ding Wang[1,2(✉)] and Ping Wang[2,3]

[1] School of EECS, Peking University, Beijing 100871, China
[2] National Engineering Research Center for Software Engineering, Beijing 100871, China
wangdingg@mail.nankai.edu.cn
[3] School of Software and Microelectronics, Peking University, Beijing 100260, China
pwang@pku.edu.cn

Abstract. Smart-card-based password authentication, known as two-factor authentication, is one of the most widely used security mechanisms to validate the legitimacy of a remote client, who must hold a valid smart card and the correct password in order to successfully login the server. So far the research on this domain has mainly focused on developing more secure, privacy-preserving and efficient protocols, which has led to numerous efficient proposals with a diversity of security provisions, yet little attention has been directed towards another important aspect, i.e. the usability of a scheme. This paper focuses on the study of two specific security threats on usability in two-factor authentication. Using two representative protocols as case studies, we demonstrate two types of security threats on usability: (1) Password change attack, which may easily render the smart card completely unusable by changing the password to a random value; and (2) De-synchronization attack, which breaks the consistence of the pseudo-identities between the user and the server. These threats, though realistic in practice, have been paid little attention in the literature. In addition to revealing the vulnerabilities, we discuss how to thwart these security threats and secure the protocols.

Keywords: Two-factor authentication · Usability · User anonymity

1 Introduction

With the rapid advancement of wireless network technologies and micro-electro-mechanical systems, more and more electronic transactions (e.g., Internet banking, online shopping, online voting, pay-TV and remote home automation) are processed on mobile devices such as PDAs, laptops, and smart phones. To enable electronic transactions to be securely processed anytime and anywhere, it is of great concern that users are authenticated by the server to prevent unauthorized access to services. Among numerous mechanisms for user authentication, smart-card-based password authentication, known as two-factor authentication [17], becomes one of the most effective and promising techniques due to its cryptographic capability and simplicity.

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 141–150, 2015.
DOI: 10.1007/978-3-319-23829-6_11

As the name implies, the most essential aim of a two-factor authentication protocol is to achieve "two-factor security" [14], which ensures that only the user who is both in possession of a valid smart card and a correct password can pass the verification of the remote server. The past thirty years of research in the domain of password authenticated key exchange (PAKE) have proved that it is incredibly difficult to get even a single-factor scheme right, designing a two-factor protocol that provides truly "two-factor security" can only be harder [7]. Besides various security goals to be met, a sound two-factor protocol shall also support a number of desirable properties such as user anonymity, forward secrecy and no password-related verification table [11]. For example, in 2012, Madhusudhan and Mittal [11] put forward a metric to evaluate the goodness of a two-factor scheme in terms of nine security goals and ten desirable properties, and they concluded that, to date though there have been abundant new proposals, yet none of them can satisfy all the nineteen design goals.

One crux of this embarrassing situation lies in how to design an efficient two-factor scheme that can achieve "two-factor security" under the assumption that smart card can be tampered when lost. Recent progresses in side-channel attacks reveal that the sensitive data stored in common smart card could be extracted [12,13]. As a result, previously deemed secure schemes (e.g., [4,5]) are prone to various attacks under this new assumption about smart cards, and hence it is more desirable to design schemes based on this new assumption. Several latest attempts [3,9] have been made, yet invariably they all have been shown to be unable to achieve "two-factor security" under such an assumption [7,14].

This paper shall study two types of serious threats that specifically target at usability but not "two-factor security". As is well known, besides desirable security and high efficiency, good usability is another indispensable criteria that a practical scheme shall satisfy. However, so far little attention has paid to this criteria. Regarding usability, as far as we know, only two properties have been mentioned in the literature [11,17]: (1) A user shall be able to choose the password by herself when registration or in the password-changing phase, hereafter we use the term "freely password choice" for short; and (2) It is admired that a user can change her password without interaction with the remote server, hereafter we use the term "freely password change" for short. As we will demonstrate in this work, there are two realistic threats that greatly undermine the usability and hence practicality of a scheme, even if this scheme is efficient and can provide the precious goal of "two-factor security".

In 2008, Yang et al. [17] showed that a traditional PAKE can be efficiently transformed into a secure two-factor authentication scheme if there exist pseudo-random functions and target collision resistant hash functions. They suggested an evaluation criteria set with five requirements, proposed a new scheme and constructed a generic framework for two-factor authentication. The fourth and fifth criteria in Yang et al.'s criteria set [17] are essentially the afore-mentioned two properties regarding usability, and these two criteria have also been incorporated into most of the later evaluation sets (e.g., [11]). However, we will show that these two criteria is subtly in contradiction with each other by demonstrating a realistic and devastating usability threat on Yang et al.'s scheme. This kind

of usability problem exists in many of the subsequent schemes, some latest ones include [6,16]. To deal with this new threat, we believe that a practical scheme shall have a property called "password change with verification".

To accommodate the privacy concerns rapidly raised among individuals and organizations, a number of two-factor schemes with user anonymity have been proposed (e.g., [5]). In 2010, Li et al. [10] pointed out that most of the previously presented anonymous two-factor schemes can only provide the basic level of user anonymity (i.e., user identity protection) and fail to preserve the more advanced anonymity property (i.e., user un-traceability) if the smart cards are assumed to be non-tamper resistant. Accordingly, they further developed a new scheme that can support the advanced anonymity property under the new assumption about smart cards. Their main strategy is by employing a synchronization mechanism to maintain the consistence of the session-variant pseudo-identities between the user and the server. This scheme is a great success in simultaneously achieving efficiency, two-factor security and user un-traceability.

However, in this work, we use Li et al.'s scheme [10] as a case study and highlight that this *initial scheme* as well as its successors (e.g., [8,16]), which using a synchronization mechanism to achieve user un-traceability, all have a fatal design flaw being overlooked. An active attacker, by simply blocking or altering a single transcript, can break the synchronization of the pseudo-identities between the user and the server, resulting in permanent authentication failure in *any* of their following protocol runs, which is "too high a price" to pay for privacy. We hope that future anonymous schemes shall be designed to avoid such a pitfall. To address this new threat, we believe that any anonymous scheme shall have a property called "no synchronization mechanism employed".

2 Review and Cryptanalysis of Yang et al.'s Scheme

2.1 Review of Yang et al.'s Scheme

In 2008, Yang et al. [17] proposed a generic construction framework to convert the conventional provably secure PAKE protocols to smart-card-based versions and further proposed a new two-factor authentication scheme to demonstrate its effectiveness. Yang et al.'s scheme consists of four phases, and here we just follow the original notations in [17] as closely as possible.

Notations. Let G be a subgroup of prime order q of a multiplicative group $\mathcal{Z}_p^*$. Let g be a generator of G. Let (PK_S, SK_S) denote a public/private key pair of the server S. Besides (PK_S, SK_S), the server S also maintains a long-term secret x which is a random string of length k. Let $H : \{0,1\}^* \rightarrow \{0,1\}^k$ denote a collision resistant hash function and $PRF_K : \{0,1\}^k \rightarrow \{0,1\}^k$ a pseudo-random function keyed by K. Let U_i stands for the ith user in the system.

Registration Phase. The registration phase involves the following steps:

R1. U_i arbitrarily selects a unique identity ID_i and sends it to S.

R2. S calculates $B_i = PRF_x(H(ID_i)) \oplus H(PW_0)$ where PW_0 is the initial password (e.g. a default such as a string of all '0's).
R3. S issues U_i a smart card which stores the security parameters $\{PK_S, ID_i, B_i, p, g, q\}$. In practice, we can have them except B_i be "burned" in the read-only memory of the card when it is manufactured.
R4. On receiving the card, U_i updates the password immediately by carrying out the password change phase as described below.

Login-and-Authentication Phase. In this phase, U_i and S interact to verify each other. As it has little relevance to our discussions, it is omitted here.

Password Change Phase. The password change phase is provided to allow users to change their passwords freely and locally. If U_i wants to change her password, the following steps is carried out:

C1. U_i keys her old password PW_i and selects a new one PW_i^{new}.
C2. Compute $B_i^{new} = B_i \oplus H(PW_i) \oplus H(PW_i^{new})$.
C3. Replace B_i with B_i^{new} in the smart card.

2.2 Cryptanalysis of Yang et al.'s Scheme

Yang et al. [17] claimed that their new scheme can satisfy all their proposed criteria, and in particular it achieves truly "two-factor security" even if the user's smart card has been lost and the secret data stored in the card is revealed. However, in the following, we will show that this scheme is actually vulnerable to a kind of denial of service attack in which an attacker can easily render the victimized smart card completely unusable once getting temporary access to it, thereby contradicting the claim made in [17] that the new scheme is secure even if the smart card is lost. In addition, this usability problem is worsened due to the fact that user herself sometimes may input a wrong password accidentally.

Yang et al. put forward a new set of five independent requirements for two-factor authentication, the last two of which are "Short Password" and "Freedom of Password Change" (See Sect. 3.1 of [17] for more details). These two requirements are essentially identical with the two usability properties introduced in Sect. 1, i.e. "freely password choice" and "freely password change", respectively. These two requirements are in favor of user friendliness, and in this light they are really reasonable. They have been incorporated into most of the later influential evaluation sets (e.g., [11]). However, a scheme achieving "freely password change" probably will go into a dilemma. Let us see what's the dilemma.

To achieve "Freedom of Password Change" (i.e., "freely password change"), the password change phase of Yang et al.'s scheme (see Sect. 2.1 (Password Change Phase)) is performed locally and does not need to interact with the remote server, which not only improves user friendliness but also reduces communication cost and the danger of disclosure of password-related transcripts. Note that, there is no verification of the old password that is input by the user when changing the old password stored in the card memory in Yang et al.'s

password change phase. In the following, we show that this practice introduces a serious usability problem.

Usability Problem. If an attacker gains temporary access (e.g., a few seconds) to U_i's smart card, then this will give rise to a quite realistic attacking scenario:

"$\cdots$ The attacker inserts U_i's smart card into a card reader and issues a password change request. Then, she selects a random string $\boldsymbol{X}$ as U_i's original password and a new string PW_i^{new} as the new password. As there is no way to determine the correctness of the old password, and the smart card will update B_i to $B_i^{new} = B_i \oplus H(\boldsymbol{X}) \oplus H(PW_i^{new})$. Since then, legitimate user U_i cannot login successfully even after getting her smart card back, because $B_i^{new} \oplus H(PW_i) \neq PRF_x(H(ID_i))$. $\cdots$"

The Dilemma. Although the above usability problem seem rather simple, it cannot be well remedied just with minor revisions. It is not difficult to see that, its root lies in the fact that no verification of the authenticity of the original password is performed before updating the long-term secret in the card memory. Accordingly, the corresponding solution would be to add this verification (either locally or by interacting with the server) when changing password, and we call schemes that perform this verification support the property "password change with verification". To provide local "password change with verification", besides B_i, some additional parameter(s) should be stored in the smart card.

Let us assume an additional parameter $A_i = H(H(PW_i))$ is kept in the smart card. Whenever U_i wants to change her password, first she must submit her old password PW_i^*, then the card checks whether $H(H(PW_i^*))$ equals the stored A_i. One can easily find that, if an adversary $\mathcal{A}$ compromises the card and obtains A_i, $\mathcal{A}$ could exhaustively search the correct password PW_i in the password dictionary $\mathcal{D}_{pw}$ in an offline manner, for the scheme satisfies the requirement "Short Password" (i.e., "freely password choice") and thus the password dictionary size is very limited, e.g., $|\mathcal{D}_{pw}| \leq 10^6$ [1]. This leads to the breach of the goal of "two-factor security", which essentially means a compromise of one factor shall not endanger the security of the other factor.

Now a dilemma arises: For a two-factor scheme that achieves "freely password change" (and "freely password choice"), if the scheme does not perform a verification of the old password, it suffers from the above usability problem; however, if the scheme performs a verification of the old password, there shall be some password-related verifier stored in the card and an attacker can just exploit this data to breach the "two-factor security".

Fuzzy Verifier. In general, there are three possible ways to take. The first one is to abandon the property "freely password change" and instead let the user change her password by interacting with the server (i.e., password verification is performed by the server). Actually, several schemes [3,10] have taken this approach, yet they have neither justified their choice nor explained why they do not favor the property "freely password change". An alternative way is to overlook the above usability problem, just like the schemes in [6,17]. The third solution is to make an acceptable tradeoff to accommodate conflicts among the four goals

"freely password choice", "freely password change", "two-factor security" and "password change with verification".

We note that, if we compute A_i as $A_i = H(\hbar(PW_i))$, then there exists $\frac{|\mathcal{D}_{pw}|}{2^8} \approx 2^{12}$ candidates of password (this space is denoted by $\mathcal{D}_{re}$, and $|\mathcal{D}_{re}| = 2^{12}$) to frustrate $\mathcal{A}$, even if $\mathcal{A}$ has extracted A_i, where $|\mathcal{D}_{pw}| = 10^6$ [1] denotes the size of the password space, and $\hbar(\cdot)$ is a special one-way hash function $\{0,1\}^* \rightarrow \{0,1,2,\ldots,255\}$. In this way, $\mathcal{A}$ is prevented from obtaining the exactly correct password and we call A_i calculated through this new method "a fuzzy verifier". This notion was discussed in [14,15], yet its effectiveness is left as an open issue.

Effectiveness. Now we investigate the effectiveness of this solution. For every password in $\mathcal{D}_{re}$, if it is indistinguishable from all the other ones by logical inference or statistical analysis, this is an ideal case. In reality, there might be some passwords that are more likely to be the password of a specific user, while some passwords more *un*likely to be the password of a specific user. For example, $\mathcal{A}$ knows the victim's family name is "Wang", it is unlikely that `Zhao****`$\in \mathcal{D}_{re}$ is the victim's real password; on the other hand, `Wang****`$\in \mathcal{D}_{re}$ is highly likely to be; `Wang****`$\in \mathcal{D}_{re}$ is more likely than `vfr4nji9`$\in \mathcal{D}_{re}$ to be. Except for such highly unlikely passwords for the victim (we assume such passwords constitute the space $\mathcal{D}_{unlikely}$), $\mathcal{A}$ has to launch an online password guessing attack to exclude every spurious password in $\mathcal{D}_{re} - \mathcal{D}_{unlikely}$ to finally determine the correct one. Now, if $|\mathcal{D}_{re}| - |\mathcal{D}_{unlikely}| \geq 2^{10}$, according to the NIST SP800-63-1 [2], our approach meets a Level 1 certification which requires that the chance of $\mathcal{A}$ succeeding in an online password guessing attempt should be less than $1/2^{10}$. The remaining question is, whether will $|\mathcal{D}_{re}| - |\mathcal{D}_{unlikely}| \geq 2^{10}$ for every password candidate in $\mathcal{D}_{pw}$, or it at least holds in most cases? This can only be testified by real-life password datasets.

Table 1. Guessing entropy (GE) distributions of password datasets that are randomly divided into 256 equally-sized password pools

Password datasets	Percentage of pools with GE $\geq$ 1024
Rockyou_Top1Million	0.00 %
CSDN_Top1Million	10.54 %
Rockyou_Top2Million	84.63 %
CSDN_Top2Million	97.66 %
Rockyou_Top$x$Million($x \geq 3$)	99.60 %
CSDN_Top$x$Million($x \geq 3$)	100.00 %

Fortunately, a number of recent catastrophic leakages of millions of web accounts (e.g., 6 million CSDN passwords[1] and 32 million Rockyou passwords[2])

[1] http://dazzlepod.com/csdn/.
[2] http://www.hardwareheaven.com/news.php?newsid=526.

have provided wonderful material for this use, and we use the metric of guessing entropy [1] to demonstrate the effectiveness of our "fuzzy verifier". This metric relates to the expected number of tries for finding the correct password using an optimal guessing strategy, i.e. trying the most likely passwords first. As far as we know, using guessing entropy to measure the effective candidates in a given password dataset is currently the best strategy that can be adopted while corresponding user-specific contextual information is unavailable (or difficult to be appropriately used due to ethic reasons). The results on guessing entropy [1] distributions of these two datasets are summarized in Table 1. Due to space constraints, the experimental designs and related calculations are omitted here. From Table 1, we can conclude that the CSDN dataset is much stronger than the Rockyou dataset in term of guessing entropy. For a password dataset as strong as the CSDN dataset (i.e., they are created under a similar password creation policy), its space shall be as large as 2 million to be able to reach a guessing entropy no less than 1024 (i.e., to meet a Level 1 certification).

3 Cryptanalysis of Li et al.'s Scheme

In 2010, Li et al. [10] made the first step towards constructing an efficient and secure two-factor scheme with user un-traceability. We now show both Li et al.'s scheme [10] and several subsequent schemes [8,16] achieve user un-traceability by largely reducing usability: an attacker who merely alters or blocks a single message flow (e.g., the second flow of [10], fourth flow of [16]), as shown in Fig. 1, can render the user *permanently* unable to login. Due to space constraints, here we do not review the scheme and readers are referred to [10] for more details.

3.1 De-synchronization Attack

To provide un-traceability, Li et al.'s "effective trick" is to randomize the transcripts in such a way that no adversary over the channel can link different conversations and that only the legitimate parties can recognize the received messages. Most essentially, the user updates its session-variant pseudonym identity $b_{ID_i}^{N_0}$ to $b_{ID_i}^{N_1}$ after having received the response from the server S and validated the legitimacy of S, while the server updates the related parameters $\{ID_i, CI_i, N_0\}$ to $\{ID_i, CI_i, N_1\}$ in its registration table before sending out its response. In this way, both U_i and S will keep the same one-time identity $b_{ID_i}^{N_1}$ that will be used in U_i's next login request. Quite a number of subsequent privacy-preserving schemes [8,16] attempt to achieve user un-traceability by adopting a similar strategy. However, the following effective de-synchronization attack demonstrates the infeasibility of such an "effective trick".

We notice that the synchronization of the one-time identities between the user U_i and the server S, i.e. $b_{ID_i}^{N_1}$ on the user and $\{ID_i, CI_i, N_1\}$ on the server, is crucial for the success of their following protocol runs. Once this consistency is broken, the user will no longer be able to login S. Actually, many factors can lead to the inconsistency between these two parties. Let us see a concrete

example. Suppose S sends $\{N_{b1}, u \oplus h_{64}(b_{ID_i}^{N_1}), M_S\}$ to U_i as per the protocol specification, which implies S has replaced $\{ID_i, CI_i, N_0\}$ with $\{ID_i, CI_i, N_1\}$ in its database. Before $\{N_{b1}, u \oplus h_{64}(b_{ID_i}^{N_1}), M_S\}$ reaches U_i, the attacker $\mathcal{M}$ intercepts this message and alters it to $\{N_{b1}, u \oplus h_{64}(b_{ID_i}^{N_1}), X\}$, where X is a random value. Upon receiving S's response, U_i will find $X \neq h(c\|u\|V_{ID_i})$, and of course, will not update $b_{ID_i}^{N_0}$ to $b_{ID_i}^{N_1}$ in the card memory. As a result, the consistency of the one-time identities between U_i and S is broken. From then on, U_i's subsequent login requests will always be rejected by S due to $N_0 \neq N_1$.

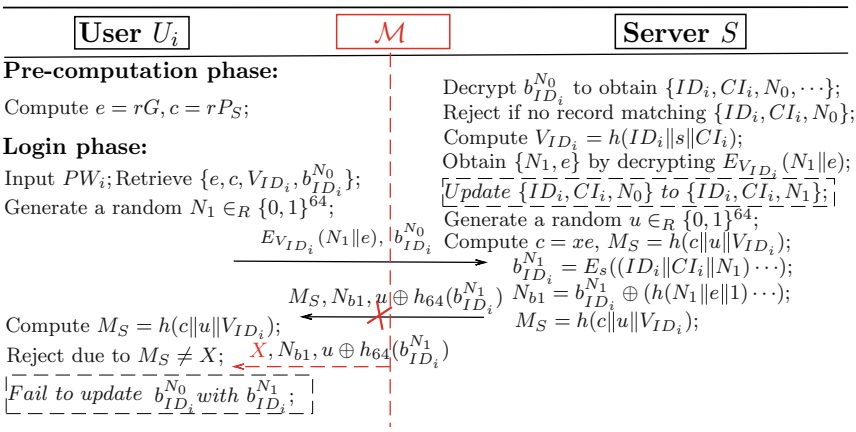

Fig. 1. De-synchronization attack on Li et al.'s scheme

Remark 1. The above attack is rather efficient and realistic, yet as far as we know, little attention (except [14]) has been given to this destructive threat *in the domain of two-factor authentication*. As with Li et al.'s scheme, its successors (e.g., [8,16]) all overlook the damaging threat of de-synchronization. This repeated failure suggests the urgency of this work to highlight the importance of being aware of potential attacks when designing a practical protocol.

Remark 2. Though the identified de-synchronization attack seems rather simple, to completely address it is not an easy task. A specious solution is that, server S defers replacing $\{ID_i, CI_i, N_0\}$ with $\{ID_i, CI_i, N_1\}$ in its database until having received the expected third message flow from U_i. However, the attacker $\mathcal{M}$ can still succeed by only blocking (or altering) the third messages flow. In this case, U_i has updated its data in the card memory before sending out the third flow, but S is waiting for the (third) message which never comes, resulting in failure in updating data on the server side.

Another seemingly workable (but unsatisfactory) fix is to store both $b_{ID_i}^{N_0}$ and $b_{ID_i}^{N_1}$ on the card memory. If a login with $b_{ID_i}^{N_1}$ succeeds, $b_{ID_i}^{N_0}$ is replaced with $b_{ID_i}^{N_2}$; otherwise, the user steps back to use $b_{ID_i}^{N_0}$ to login. While this patch alleviates

the presented attack, it may leads to the violation of user un-traceability: if $\mathcal{M}$ blocks the login request that using $b_{ID_i}^{N_2}$ (which means the previous login request has used $b_{ID_i}^{N_1}$), then U_i will step back to use $b_{ID_i}^{N_1}$ to login. This means U_i has using the same pseudo-identity $b_{ID_i}^{N_1}$ in two login request and thus can be traced.

In a nutshell, there is no easy way to work out the identified problem on how to maintain the consistency of the one-time identities between U_i and S when using some synchronization mechanism to achieve user un-traceability. This suggests a call for a requirement that "no synchronization mechanism is employed".

4 Conclusion

In this work, we have employed two influential schemes, i.e. Yang et al.'s scheme [17] and Li et al.'s scheme [10], as case studies to show that the usability issues of previous two-factor authentication schemes should have been paid more attention. We propose the properties "password change with verification" and "no synchronization mechanism employed" as important usability criteria when designing and evaluating a two-factor scheme. We also discuss the solutions to cope with the identified issues. To the best of knowledge, this work is the first one that mainly focus on the useability problem of two-factor schemes, which we believe deserves attention from both the academia and the industry. A natural future work is to fully identify the practical threats on two-factor authentication and develop efficient and usable schemes with provable security.

Acknowledgment. This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No.61472016.

References

1. Bonneau, J.: The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: IEEE S&P 2012, pp. 538–552. IEEE Computer Society (2012)
2. Burr, W., Dodson, D., Perlner, R., Polk, W., Gupta, S., Nabbus, E.: NIST SP800-63-1 - electronic authentication guideline. Technical report, NIST, Reston, VA (2006)
3. Chen, B.L., Kuo, W.C., Wuu, L.C.: Robust smart-card-based remote user password authentication scheme. Int. J. Commun. Syst. **27**(2), 377–389 (2014)
4. Chen, C., Tang, S., Mitchell, C.J.: Building general-purpose security services on EMV payment cards. In: Keromytis, A.D., Di Pietro, R. (eds.) SecureComm 2012. LNICST, vol. 106, pp. 29–44. Springer, Heidelberg (2013)
5. Das, M., Saxena, A., Gulati, V.: A dynamic ID-based remote user authentication scheme. IEEE Trans. Consum. Electron. **50**(2), 629–631 (2004)
6. He, D., Kumar, N., Khan, M.K., Lee, J.H.: Anonymous two-factor authentication for consumer roaming service in global mobility networks. IEEE Trans. Consum. Electron. **59**(4), 811–817 (2013)

7. Huang, X., Chen, X., Li, J., Xiang, Y., Xu, L.: Further observations on smart-card-based password-authenticated key agreement in distributed systems. IEEE Trans. Parallel Distrib. Syst. **25**(7), 1767–1775 (2014)
8. Kumari, S., Khan, M.K.: Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme. Int. J. Commun. Syst. (2013). doi:http://dx.doi.org/10.1002/dac.2590
9. Li, C.T.: A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Inform. Secur. **7**(1), 3–10 (2013)
10. Li, X., Qiu, W., Zheng, D., Chen, K., Li, J.: Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. IEEE Trans. Ind. Electron. **57**(2), 793–800 (2010)
11. Madhusudhan, R., Mittal, R.: Dynamic ID-based remote user password authentication schemes using smart cards: a review. J. Netw. Comput. Appl. **35**(4), 1235–1248 (2012)
12. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Comput. **51**(5), 541–552 (2002)
13. Oswald, D., Paar, C.: Breaking mifare DESFire MF3ICD40: power analysis and templates in the real world. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 207–222. Springer, Heidelberg (2011)
14. Wang, D., He, D., Wang, P., Chu, C.H.: Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans. Depend. Secur. Comput. (2014). doi:http://dx.doi.org/10.1109/TDSC.2014.2355850
15. Wang, D., Ma, C.G., Wang, P., Chen, Z.: iPass: robust smart card based password authentication scheme against smart card loss problem. J. Comput. Syst. Sci. (in press, 2014). http://eprint.iacr.org/2012/439.pdf
16. Wen, F., Susilo, W., Yang, G.: A secure and effective anonymous user authentication scheme for roaming service in global mobility networks. Wireless Pers. Commun. **73**(3), 993–1004 (2013)
17. Yang, G., Wong, D., Wang, H., Deng, X.: Two-factor mutual authentication based on smart cards and passwords. J. Comput. Syst. Sci. **74**(7), 1160–1172 (2008)