Anonymous Two-Factor Authentication: Certain Goals Are Beyond Attainment

Ding Wang[1,3], Ping Wang[1,3], and Debiao He[2]

[1] School of EECS, Peking University, Beijing 100871, China
[2] National Engineering Research Center for Software Engineering, Beijing 100871
[3] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
wangdingg@pku.edu.cn; pwang@pku.edu.cn; hedebiao@163.com

Abstract. Despite a decade of intensive research, it still remains a challenge to design a practical dynamic id-based two-factor authentication scheme, for the designers are confronted with an impressive list of security requirements (e.g., resistance to mart card loss attack) and desirable attributes (e.g., local and secure password update). Dozens of solutions have been proposed, yet most of them are shortly found either unable to satisfy some security requirements or short of important features. To overcome this unsatisfactory situation, researchers often work around it in hopes of a new solution (but no one has succeeded so far), while paying little attention to the question: Whether or not there are inherent limitations (conflicts) that prevents us from designing an ideal scheme that satisfies all of these goals?

In this work, we attempt to provide an answer to this question. We revisit two latest (and reprentative) proposals, i.e. Xie's scheme and Li's scheme, and explore some inherent conflicts and unavoidable trade-offs in designing such schemes. Our results highly indicate that, under the current widely accepted adversary model, certain goals are beyond attainment. To the best of knowledge, the present study makes the first step towards understanding the underlying evaluation metric for dynamic id-based two-factor authentication, which we believe will facilitate better design of two-factor protocols that offer acceptable trade-offs between usability, security and privacy.

Keywords: Authentication protocol, User anonymity, Smart card, Non-tamper resistant, Smart card loss attack.

1 Introduction

Password authentication with smart card is one of the most convenient and effective two-factor authentication mechanisms for remote systems to assure one communicating party of the authenticity of the corresponding party by acquisition of corroborative evidence. Although this technique has been widely deployed for various kinds of daily applications, such as e-government, e-banking [28] and e-health [5], there are severe challenges on issues of security [2, 65], privacy [27] and usability [30] due to the great system complexity and resource-constrained characteristics of mobile devices.

Since Chang and Wu [17] introduced the first smart-card-based password authentication scheme in 1993, there have been many of this type of schemes proposed [19, 22, 34, 50, 57]. In most of the previous two-factor schemes, the user's identity is transmitted in plain-text over public networks during the login process, which may leak the identity of the logging user once the login transcripts were eavesdropped, resulting in violation of the user's privacy and raising legal issues in some scenarios,

e.g. electronic auditing or secret online-order placement. In many cases, an adversary may exploit the static user identity to link different login sessions together to trace user activities. For example, in e-commerce applications, once user activities are traced, the sensitive information such as shopping patterns, individual preferences, etc., can be obtained and abused for marketing purposes [8]. What's more, the disclosure of user identity and activities may also facilitate an unauthorized entity to trace the user's login history and even current location [79]. To address such static-ID-related issues, a feasible approach is to adopt the "dynamic ID technique": the user's real (static) identity is concealed in session-variant pseudo-identities. And schemes that employ this technique are known as dynamic ID-based schemes [24]. Generally, there are two approaches to implement "dynamic ID technique": (1) by a cryptographic primitive (e.g., a CCA-2 public key encryption [29]) like that of [83]; and (2) by a non-cryptographic synchronization mechanism like that of [55].

In 2004, Das [24] introduced the first dynamic ID-based two-factor authentication scheme to preserve user privacy. Das's seminal work has been followed by a number of proposals [18, 21, 32, 42, 86] with various levels of security and diversity of attributes. A common feature of these schemes is that their security is based on the tamper resistance assumption of smart cards, i.e., they simply assume that the security parameters stored in the smart card cannot be extracted. However, recent research results have illustrated that common commercial smart cards can no longer be considered to be fully tamper-proof: the secret information stored in the smart card memory could be revealed by power analysis [61, 63, 69], reverse engineering techniques [45, 68] or fault injection attacks [9,31]. As a consequence, such schemes based on the tamper-resistance-assumption of the smart cards are susceptible to some types of attacks, such as user impersonation attack, offline dictionary attack, etc., once an adversary has breached the smart card. Therefore, it is more desirable and practical to assume that once a smart-card is in the possession of an adversary, all the sensitive data stored in it are no longer secret. With this in mind, a number of dynamic ID-based schemes [44, 55, 85, 88, 90] based on *non-tamper* resistance assumption of the smart cards are put forward, and each is claimed to meet a self-imposed list of ambitious design goals.

Motivations. More often than not, the proponents assert the superiorities of their scheme, while (unintentionally or intentionally) ignoring the benefits that their scheme fails to support, thus overlooking dimensions on which it fares poorly. This has contributed to a long-standing lack of progress on how best to evaluate proposals intended for practical applications. To address this issue, In 2012 Madhusudhan and Mittal [59] developed a new set of design goals (including nine security requirements and ten desirable attributes) for fairly evaluating this type of schemes. Their set is a refinement of some previously proposed criteria sets (e.g., [57, 80, 85]) and as far as we know, it is so far the most comprehensive, explicit and systematic criteria set for evaluating dynamic ID-based two-factor schemes. In Madhusudhan-Mittal's work [59], it is concluded that all existing dynamic ID-based two-factor schemes are far from ideal and each has its own pros and cons, it still remains an open problem as to *how to* design an ideal scheme that can satisfy all the criteria in their evaluation set.

The pattern of progress on this problem has been of suggested new solutions (e.g., [23,48,78,83]), followed by cryptanalysis reports (e.g., [62,71]), which, once again, falls into the unsatisfactory "break-fix-break-fix" cycle (see Fig.1 of [84]). In this undesirable cycle, protocol designers work around the above problem by presenting "improved" schemes but with not much confidence, cryptanalysts respond to the above problem with concrete rebuttals to new proposals, while no one pays attention to the underlying (fundamental) question: Whether a proposal fails due to improper designs or due to some inherent limitations of this type of schemes? Or equally, this question can be expressed as: *Whether is it possible* to construct an ideal scheme which satisfies all the design goals in [59]? Without an answer to this fundamental question, we can only be kept stuck in the rut: lots of attempts are continually being contributed (and subsequently being defeated), yet little progress will be made.

Contributions. This study aims to provide a definite answer to the above question. We first revisit the security and attribute provisions of two recent proposals, namely Xie's scheme [93] and Li's scheme [87], and reveal some new challenges and subtleties in designing dynamic ID-based two-factor schemes. As far as we know, these two schemes are among the foremost ones and claimed to be secure against various known attacks and provide many admired features. Remarkably, we identify two new practical threats, i.e. undetectable online password guessing attack and de-synchronization attack, which can be specially targeted at dynamic ID-based two-factor schemes.

Using these two representative schemes as case studies, we further investigate into the relationships among the criteria in Madhusudhan-Mittal's evaluation set [59], showing some inherent conflicts and unavoidable trade-offs in designing dynamic ID-based two-factor schemes. Our results highly indicate that, under the current widely accepted adversary model, certain goals are beyond attainment. To the best of knowledge, the present study makes the first step toward exploring the inner relationships of evaluation criteria for dynamic ID-based two-factor authentication, which we believe will provide a better understanding of how to design two-factor protocols that offer acceptable trade-offs between usability, security and privacy.

Organization. The remainder of this paper is organized as follows: in Section 2, we elaborate on the system model, adversary model and evaluation criteria. Xie's scheme is cryptanalyzed in Section 3. Section 4 describes the weaknesses of Li's scheme. The relationships among evaluation criteria in Madhusudhan-Mittal's set are explored in Section 5, while conclusions are offered in Section 6.

2 System architecture, adversary model and evaluation criteria

In this section, we describe the system architecture, adversary model and evaluation criteria. It is worth noting that, these three elements are key factors in determining whether a scheme has been evaluated systematically or fairly. There have been hundreds of papers dealing with smart-card-based password authentications schemes quite recently (e.g., [19, 39, 42, 55, 86–88, 90, 98]), yet as far as we know, only a few ones [83,90,96] explicitly define these three elements (especially the later two elements)

in their work, which may well explain why despite two decades of intensive research, there is still little consensus reached. Consequently, before stepping into the details of protocol specifications, we describe the system model, define the currently widely accepted adversary model and introduce Madhusudhan-Mittal's criteria set [59].

2.1 System architecture

In this work, as with [35, 36, 90], we mainly focus on the most general case of mart-card-based password authentication (see Fig.1), in which the participants involve a set of users and a single remote server. Typically, this kind of schemes consist of three basic phases, i.e. registration, authentication and password change, as well as some supplementary phases like eviction and revocation [90]. In the registration phase, a user submits some personal information to the server, and the server issues a smart card to the user. The smart card may contain some public and sensitive security parameters, which will be used later for the authentication. This phase is carried out only once unless the user re-registers. Upon accomplishment of the registration phase, the user is able to access the server in the authentication phase. This phase can be performed as many times as needed. What a truly two-factor scheme can guarantee is that, only the user who is in possession of a valid (i.e., not revoked) smart card and the corresponding password can pass the verification the server. In the password change phase, the user can change her password and update the information in the card either locally or by interacting with the server. To evict a malicious user and to revoke a lost card, admired schemes may also provide additional phases such as eviction phase and revocation phase, respectively.

Fig. 1. Smart-card-based password authentication

2.2 Adversary model

In the conventional password authenticated key exchange (PAKE) protocols (e.g., [1, 41]), the attacker $\mathcal{A}$ is generally assumed to be able to eavesdrop, block, alter or insert messages exchanged between the communicating parties, i.e., in full control of the communication channel. Besides, previous session key(s) may also be learnt by $\mathcal{A}$ due to a variety of reasons [46]. To capture the notion of forward secrecy, $\mathcal{A}$ may also be allowed to corrupt legitimate parties to learn long term secrets.

Though these assumptions are reasonable for password-only authentication scenarios, it is inadequate for capturing practical threats in smart-card-based password authentication environments. As mentioned earlier, the secret data stored in the smart card, which was once believed to be free from breach, could be extracted by state-of-the-art side-channel attacks [9,61,63,69]. What's more, malicious card readers also contribute to the security failures of such schemes: a user's input password may be easily intercepted (key-logged) by a malicious card reader. It shall be noted that, as observed in [87] and further investigated in [84], $\mathcal{A}$ is unlikely to extract the secret information stored in the card while intercepting a victim's input password through malicious card readers, for the victim is on the scene and thus there is little chance for $\mathcal{A}$ to perform abnormal operations such as side-channel attacks.

Last but not least, it is practical to assume that a determined adversary can somehow learn the victim's identity when having obtained the victim's card. Firstly, user's identity is static and often confined to a predefined format, and thus it is of little cryptographic strength [14], and thus it is easily guessed. Secondly, it probably can be harvested from popular forums and other open resources. Thirdly, users while used to the idea of keeping passwords a secret would not normally be expecting to keep their identities a secret as well [71], e.g., writing their identities directly on the card. After all, $\mathcal{A}$ can know more or less about the personal information of the card holder when she has obtained the card. In a word, it is more reasonable to *do not* consider user identity as a surrogate extra password.

Here arises a subtlety to be explicated. This assumption about user identity here is to emphasize that the security of a two-factor scheme shall not rely on the secrecy of user identities, and it is completely different from the assumption that $\mathcal{A}$ can determine a user's identity merely from the protocol transcripts. In other words, when dealing with *the security* of a scheme, it is more practical to regard user identity as a known value; however, when dealing with *the privacy provisions* of a scheme, the target user's identity is just what $\mathcal{A}$ is looking for from the publicly available protocol transcripts.

Table 1. Capabilities of the adversary

C-01	The adversary $\mathcal{A}$ can enumerate offline all the elements in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$ within polynomial time, where $\mathcal{D}_{pw}$ and $\mathcal{D}_{id}$ denote the password space and the identity space, respectively.
C-02	The adversary $\mathcal{A}$ has the capability of somehow learning the victim's identity when evaluating security strength (but not privacy provisions) of the protocol.
C-1	The adversary $\mathcal{A}$ is in full control of the communication channel between the communicating parties.
C-2	The adversary $\mathcal{A}$ may either learn (i) the password of a legitimate user via malicious card reader, or (ii) extract the secret information of the smart card by side-channel attacks, but cannot achieve both.
C-3	The adversary $\mathcal{A}$ can learn the previous session key(s).
C-4	The adversary $\mathcal{A}$ has the capability of learning server's long-time private key(s) only when evaluating the resistance to eventual failure of the server (e.g., forward secrecy).

The capabilities of the adversary $\mathcal{A}$ are summarized in Table 1. The adversary model presented here is based on the models introduced in [33, 49, 83, 84, 87, 90, 94]. The key difference is that in our model, we for the first time explicitly define $\mathcal{A}$'s capabilities related to user identity from both the security prospective and the privacy prospective. This separation enables us to deal specifically with dynamic ID-based two-factor schemes.[4] Otherwise, some effective attacks specifically aiming at dynamic ID-based schemes can never be captured, for example, the smart card loss attack presented in [71] and the undetectable online password guessing attack presented in Section 3.2.

2.3 Evaluation criteria

In 2012, Madhusudhan and Mittal [59] pointed out that earlier criteria sets, e.g. [57,80], have ambiguities and redundancies, and thus they proposed a new criteria set of nine security requirements (see Table 2) and ten desirable attributes (see Table 3) to evaluate the goodness of dynamic ID-based schemes. This criteria set is a refinement of earlier criteria sets, and interested readers are referred to [59] for the specific definition of each criterion. Here we only point out some subtleties, the inner relationships among the criteria will be investigated in Section 5.

Table 2. Security requirements

SR1	DoS attack
SR2	Impersonation attack
SR3	Parallel session attack
SR4	Password guessing attack
SR5	Replay attack
SR6	Smart card loss attack;
SR7	Stolen-verifier attack
SR8	Reflection attack
SR9	Insider attack

Table 3. Desirable attributes

DA1	No password-related verifier table
DA2	Freely user password choice
DA3	No password reveal
DA4	Password dependent
DA5	Mutual authentication
DA6	Session key agreement
DA7	Forward secrecy
DA8	User anonymity
DA9	Smart card revocation
DA10	Efficiency for wrong password login

"To be called an ideal", a scheme should be able to satisfy all the nine security requirements and achieve all the ten desirable attributes. After putting forward the criteria set, Madhusudhan and Mittal [59] analyzed six recently proposed dynamic ID-based schemes, and found none of them can satisfy all the above seventeen criteria. Accordingly, they concluded that it still remains an open problem to construct a scheme that is to be called an ideal. Madhusudhan-Mittal's criteria set is superior to other proposed criteria for the following two reasons: (1) The security requirements of their criteria set are based on the non-tamper resistance assumption of the smart cards, which is desirable when taking into consideration the state-of-the-art side-channel attacks; (2) The separation of security requirements and desirable attributes makes the criteria set

[4] Note that, user identities are transmitted in plain over the public channel in common two-factor schemes (i.e., without consideration of user privacy).

more concrete and facilitates protocol designers to establish a systematic approach for analyzing this type of schemes. Consequently, we prefer Madhusudhan-Mittal's criteria set as a representative benchmark to other criteria sets (e.g. [57,80,90,96]) in this study.

For a better comprehension, we address two subtleties (which are not made clear in the original work [59]) related to the definitions of the above criteria. Firstly, security requirement SR6 relates to an adversary who has obtained the victim user's smart card, while all the other security requirements (e.g., SR2 and SR4) are faced with an adversary who is without the victim user's smart card. Secondly, in the context of remote user authentication, user anonymity (i.e., DA8) generally includes two aspects, i.e. identity protection and user un-traceability [55, 85]. Both are defined against the public (eavesdropping attackers) rather than the server [60], because the server has to first identify the legitimacy of the user and then obtain the user's real identity for accounting and/or billing purposes.

3 Review and cryptanalysis of Xie's scheme

In 2012, noticing that all the existing dynamic ID-based password authentication protocols are vulnerable to various attacks, Xie [93] proposed a new dynamic ID-based password authentication protocol using non-tamper resistant smart cards. It is claimed that "this scheme is the first secure dynamic ID-based password authentication protocol". In this work, however, we will demonstrate that, as the results of overlooking some inherent trade-offs (conflicts), Xie's scheme cannot withstand undetectable online password guessing attack and smart card loss attack, and is vulnerable to several practical pitfalls.

3.1 Review of Xie's scheme

In this section, we briefly illustrate the remote user authentication scheme proposed by Xie [93] in 2012. Their scheme consists of four phases: the registration, the login, the verification and password update. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 2.

Registration phase. The registration phase involves the following operations:

1) User U_i chooses his/her identity ID_i, password PW_i.
2) $U_i \Rightarrow S : \{ID_i\}$.
3) On receiving the registration message from U_i, S computes $N_0 = h(ID_i)^x \bmod p$.
4) $S \Rightarrow U_i$: A smart card containing security parameters $\{ID_i, N_0, h(\cdot)\}$.
5) Upon receiving the smart card, user U_i keys her password PW_i into the smart card SC, then SC computes $N_i = N_0 \oplus h(PW_i)$.

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

1) U_i inserts her smart card into the card reader, and inputs PW_i. The smart card SC generates two random numbers b and k, and then computes $N_s = g^k \bmod p$,

Table 4. Notations

Symbol	Description
U_i	i^{th} user
S	remote server
ID_i	identity of user U_i
CID_i	dynamic identity of user U_i
PW_i	password of user U_i
SC	smart card
x	the secret key of remote server S
p, q	two large prime numbers, such that $p-1$ is divisible by q
g	a primitive element in Galois $GF(n)$
$h(\cdot)$	a common collision free one-way hash function, e.g. SHA-1
$\hbar(\cdot)$	a special one-way hash function $\{0,1\}^* \rightarrow \{0,1,2,3,\ldots,255\}$
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$A \rightarrow B : C$	message C is transferred through a common channel from A to B
$A \Rightarrow B : C$	message C is transferred through a secure channel from A to B

$CID_i = h(ID)^b \bmod p$, $C_0 = (N_i \oplus h(PW_i))^b = h(ID_i)^{xb} \bmod p$, $C_1 = C_0 \oplus N_s = h(ID_i)^{xb} \oplus N_s \bmod p$.

2) $U_i \rightarrow S : \{CID_i, C_1\}$.

Verification phase. After receiving the login request message from user U_i, the server S performs:

1) The server S generates a random number d, computes $N_u = g^d \bmod p$, $N_s = C_1 \oplus {CID_i}^x \bmod p$, $C_2 = h({N_s}^d)$.

2) $S \rightarrow U_i : \{C_2, N_u\}$.

3) On receiving the response from server S, smart card SC computes $C_2' = h({N_u}^k)$ and compares C_2' with the received C_2. If they are equal, U_i successfully authenticates S. Then, SC computes $C_3 = h({N_u}^k + 1)$.

4) $U_i \rightarrow S : \{C_3\}$

5) On receiving C_3, the serve S compares the computed $C_3' = h({N_s}^d + 1)$ with the received value of C_3 . If they are not equal, the connection is terminated.

6) The user U_i and the server S agree on the session key $SK = h({N_u}^k{+}2) = h({N_s}^d{+}2)$ for securing future data communications.

Password change phase. The password change phase is provided to allow users to change their passwords freely and locally. When U_i wants to change his password, she inserts smart card into a terminal device, keys his old password PW_i and the new password PW_i^{new} , the SC computes $N_i^{new} = N_i \oplus h(PW_i) \oplus h(PW_i^{new})$ and replaces N_i with N_i^{new}.

3.2 Cryptanalysis of Xie's scheme

Although Xie's scheme has many attractive properties, such as provision of forward secrecy, resistance to offline password guessing attack even if the user's smart card is

lost and the secret data stored in the card is revealed, it is still far from an "ideal" dynamic ID-based password protocol to be applicable for practical applications. In this section we will show that, it fails to achieve some critical security goals under the author's non-tamper resistance assumption of the smart card.

Undetectable online password guessing attack. Password-based authentication protocols are prone to guessing attacks as passwords are generally drawn from a small space enumerable by an adversary. Based on how passwords are tested by an adversary, there are generally three types of password guessing attacks, namely, offline password guessing attack, undetectable on-line dictionary attack and detectable on-line dictionary attack. As discussed in [97], detectable on-line dictionary attack can be effectively thwarted by accounting the number of unsuccessful protocol runs, however, the other two password guessing attacks still pose serious security threats.

In an undetectable on-line dictionary attack, the adversary randomly selects a password from the password space and impersonates as a legitimate client by carrying out an honest run of the protocol. If the protocol run succeeds, it indicates that the adversary has correctly guessed the password; otherwise the adversary excludes her guess from the password space and restarts a new session with the server by using another guessed password. A failed password guess *cannot* be detected and logged by the server, and thus there is no way for the server to distinguish an honest request from a malicious one. This type of attack is in-depth investigated by Ding et al. in [97], more details can be found there.

Let's see how this attack could be successfully launched with Xie's scheme in place. In case a legitimate user U_i's smart card is somehow obtained (stolen or picked up) by an adversary, and the stored secret N_i can be revealed by some means, such as monitoring the power consumption [61, 63, 69] or analyzing the leaked information [40, 45]. Now an adversary pretends to be user U_i and tries to guess U_i's password. The following is performed by the adversary $\mathcal{A}$:

Step 1. Randomly choose a pair (ID_i^*, PW_i^*) from $\mathcal{D}_{id} \times \mathcal{D}_{pw}$, where $\mathcal{D}_{id}$ denotes the identity space and $\mathcal{D}_{pw}$ denotes the password space.
Step 2. Computes $N_s = g^k \bmod p$,where k is randomly selected.
Step 3. Computes $CID_i = h(ID_i^*)$,$C_0 = N_i \oplus h(PW_i^*)$ and $C_1 = C_0 \oplus N_s = N_i \oplus h(PW_i^*) \oplus N_s \bmod p$, where N_i is extracted from the smart card.
Step 4. Sends $\{CID_i, C_1\}$ to server S.
Step 5. On receiving the response $\{C_2, N_u\}$ from S, $\mathcal{A}$ computes $C_2' = h({N_u}^k)$.
Step 6. Verifies the correctness of PW_i^* by checking if C_2' equals the received C_2. If C_2' does not equal C_2 , the session is terminated.
Step 7. Repeats Steps 1, 2, 3, 4, 5, 6 and 7 of this procedure until the correct value of PW_i and ID_i is found.

Upon receiving $\{CID_i, C_1\}$ sent by $\mathcal{A}$ in above Step 5, the server does not proceed to find out who the communicating party is, and actually it is also easy to see that in no way can the server S obtain the identity of the attacker. As a result, a failed

protocol run cannot be detected and logged by the server and the server is not able to distinguish an honest request from a malicious request. Being totally unware of whose account is being attacked, the server can do nothing to thwart the above attack.

Let $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the number of identities in $\mathcal{D}_{id}$ and the number of passwords in $\mathcal{D}_{pw}$, respectively. The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (3T_E + 2T_H + 2T_X))$, where T_E is the running time for modular exponentiation, T_H is the running time for Hash operation and T_X is the running time for XOR operation. Since both password and user identity are human-memorable short strings but not high-entropy keys, that is to say, they are often selected from two corresponding dictionaries of small size, e.g. $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^6$ [13, 26, 91].

Now whether or not the above attack can be completed in polynomial time (e.g., a few days or less) essentially depends on whether it can be carried out automatically (i.e., without the involvement of human labour). We note that many systems have implemented a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) scheme [4] as a standard security mechanism to prevent automated attacks by bots or other automated malicious programs, yet this is not a hard obstacle for $\mathcal{A}$ to cross. Recent studies [15, 52, 67] have shown that, to date most of the existing CAPTCHA schemes can be successfully broken. In particular, Li et al. [52] conducted a comprehensive study on e-banking CAPTCHAs deployed around the world, in which they investigated three e-banking CAPTCHA schemes for transaction verification and 41 schemes for login. Their results are quite alarming: "all of the e-banking CAPTCHA schemes are broken with a success rate equal to or close to 100% in seconds". That is, even being against real systems with CAPTCHAs (e.g., e-banking), the above attack can be performed automatically with recourse to the state-of-the-art techniques in attacking CAPTCHAs.

Of course, the server may thwart this type of attacking strategy by blocking enormous login attempts (e.g., the login frequency exceeding a predefined value) from the same IP address, but then there are chances that the attacker constructs botnets [74] running coordinated distributed attacks from thousands different computers, each with unique IP addresses. At present, there is no effective countermeasure against this type of threat other than not using schemes with such vulnerability. In a nutshell, the identified attack really constitutes a practical threat.

Remark 1. One may have already noticed that, the user anonymity property achieved in Xie's scheme is bit different from the common user anonymity notion as introduced in Section 2.3. In Xie's scheme, not only the (external) attacker but also the server can not link two sessions performed by the same user. This means user anonymity is also defined against the server. It is just this "strongly anonymous" property that makes the scheme susceptible to the above attack. Surprisingly, this property is consistent with the definition of DA8 in [59]. As far as we know, Xie's scheme is so far the only one that achieves such a "strongly anonymous" property. Since this property ensures stronger user privacy than the common notion of "user anonymity" property (i.e., DA8), we thereafter call it "DA8-Strong".

Smart card loss attack. To support local user password update (i.e., DA2) like that of the schemes in [42, 85, 88, 90, 96], the password change phase of Xie's scheme is performed locally and does not need to interact with the remote server, which is in favor of user friendliness. However, this phase introduces an inherently insecure factor: there is no verification of the authenticity of the old password before the update of new password. If an attacker manages to gain temporary access to the smart card of legitimate user U_i (note that this is a quite realistic assumption), he can easily change the password of user U_i as follows:

Step 1. The attacker inserts U_i's smart card into a card reader and initiates a password change request.
Step 2. The attacker submits a random string R as U_i's original password and a new string PW_i^{new}as the targeting new password.
Step 3. The smart card computes $N_i^{new} = N_i \oplus h(R) \oplus h(PW_i^{new})$ and updates N_i with N_i^{new}.

Once the value of N_i is updated, legitimate user U_i cannot login successfully even after getting his/her smart card back because $N_i^{new} \oplus h(PW_i) \neq h(ID_i)^x \bmod p$, and thus U_i's login request will be denied by the server S during the verification phase. Hence, denial of service attack can be launched successfully.[5]

It should be noted that, although this vulnerability seems too basic to merit discussion, it cannot be well remedied just with minor revisions. To conquer this vulnerability while preserving DA2 and fulfilling SR6, a verification of the authenticity of the original password before updating the value of N_i in the memory of smart card is essential. And thus, besides N_i, some additional parameter(s) should be stored in the smart card, which may introduce new vulnerabilities, such as offline guessing attack and user impersonation attack. This subtlety has also been observed by Nam et al. [66] and Xiang et al. [92], unfortunately, they left it as an open problem. Most subsequent works either just overlook this issue [42, 81, 85, 86, 88, 90, 96] or choose not to provide local user password change [19, 22, 51, 55, 94], while the few rest [20, 56, 76, 98] that are ambitious to support local password change while withstanding smart card loss attack are all found prone to offline password guessing attack [58, 82, 95].

To gain more insights into this problem, here we give a concrete example. let's assume an additional parameter $A_i = h(ID_i \parallel h(PW_i))$ is stored in the smart card. Whenever U_i wants to change her password, first she must submit her identity ID_i^* and password PW_i^*, then the smart card checks whether $h(ID_i^* \parallel h(PW_i^*))$ equals the stored A_i. One can easily find that an adversary can exhaustively search the correct pair (ID_i, PW_i) in an offline manner once the parameter A_i is obtained, which definitely leads to an offline guessing attack.

However, if the parameter A_i is computed as $A_i = h(\hbar(ID_i) \oplus \hbar(PW_i))$, it is not difficult to check that there exists $\frac{|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|}{2^8} \approx 2^{32}$ candidates of (ID, PW) pair to

[5] Actually, this design flaw may also give rise to another quite practical problem: if a legitimate user accidently keys an incorrect value for the current (old) password in the password change process, N_i will be updated to a unpredictable value and the smart card will become unusable unless she re-register with the server.

frustrate $\mathcal{A}$ when $|\mathcal{D}_{id}| = |\mathcal{D}_{pw}| = 10^6$ [13,26,91], where $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the size of the identity space and password space, respectively. In this way, $\mathcal{A}$ is prevented from obtaining the exactly correct (ID, PW) pair and we call the parameter A_i calculated through this new method "a fuzzy verifier". An obvious "side effect" of this "fuzzy verifier" is that it can be used to fulfill criterion DA10. One may argue that what if U_i happens to submit a wrong (ID_i^*, PW_i^*) pair such that $h(\hbar(ID_i^*) \oplus \hbar(PW_i^*)) = A_i$, while $(ID_i^*, PW_i^*) \neq (ID_i, PW_i)$? The reality is that this possibility is only $\frac{1}{256}$, which is too small to significantly degrade the effectiveness of DA10.

Therefore, only radical changes in Xie's scheme can overcome the above vulnerability, and we conjecture that there is an unavoidable trade-off when fulfilling the criteria DA2, DA10 and SR6, which will be further discussed in Section 5.2. Employing "a fuzzy verifier" seems a good choice to deal with this problem, yet its practical effectiveness can only be testified by real-life password data sets. Fortunately, a number of recent catastrophic leaks of thousands of millions web accounts (e.g., CSDN [25] and Linkedin [72]) have provided wonderful material for this use, and we leave it for future work.

Key compromise impersonation attack. Besides resistance to basic passive and active attacks, an authentication scheme with key agreement is more desirable and practical if the following advanced security features are also taken into consideration [11, 55]: known session keys, forward secrecy, unknown key share, key-compromise impersonation (KCI), and key control. As with the well-known feature of forward secrecy, resistance to KCI attack is also concerned with limiting the effects of ultimate failures, in case of the disclosure of server's long-term private keys [46]. In the case of KCI, the question is whether the knowledge of a communicating party i's private key allows a malicious attacker $\mathcal{A}$ not only to impersonate participant i to others but also to impersonate other uncorrupted parties $j(j \neq i)$ to i. Schemes that can prevent this kind of reverse impersonation are said to withstand KCI attack. Suppose the long-time secret key x of the server S is leaked out by accident or intentionally stolen by the adversary $\mathcal{A}$. Without loss of generality, we assume one of U_i's previous login requests, say $\{CID_i, C_1^n\}$ (exchanged during U_i's nth login process), is intercepted by $\mathcal{A}$. Once the value of x is obtained, with the intercepted CID_i and C_1^n, $\mathcal{A}$ can impersonate the legitimate user U_i since then through the following method:

Step 1. Computes $N_s = g^k \bmod p$,where k is randomly selected.
Step 2. Computes $C_0 = (CID_i)^x \bmod p$ and $C_1 = C_0 \oplus N_s = h(ID_i)^{xb} \oplus N_s \bmod p$, where CID_i is previously intercepted.
Step 3. Sends $\{CID_i, C_1\}$ to server S.
Step 4. On receiving the response C_2, N_u from S, $\mathcal{A}$ computes $C_3 = h({N_u}^k + 1)$ and the session key $SK = h({N_u}^k + 2)$.
Step 5. Sends $\{C_3\}$ to server S.

After receiving $\{CID_i, C_1\}$ sent by $\mathcal{A}$ in Step 3, S will perform Step $V1$ of the verification phase. It is easy to see that the value of N_s computed by S will be the same with the value of N_s generated in Step 1 by $\mathcal{A}$, because both CID_i and C_1

have been computed with the correct x. As a result, the following relationship holds true: $C_3' = h({N_s}^d + 1) = h({N_u}^k + 1) = C_3$. That is, upon receiving $\{C_3\}$ sent by $\mathcal{A}$ in Step 5, S will perform Step $V5$ of the verification phase without observing any abnormality. Therefore, server S will accept $\mathcal{A}$'s login request after receiving C_3. In the end, server S and $\mathcal{A}$ will hold the same session key $SK = h({N_u}^k + 2) = h({N_s}^d + 2)$. By generalizing the above attack, $\mathcal{A}$ can easily imitate any user to login S at any time without employing any special cryptographic techniques. Hence, Xie's scheme cannot withstand KCI attack. [6]

No smart card revocation. If the server wants to suspend a lost card, she should know some specific information about this lost card. It is not difficult to see that, in Xie's scheme, the old smart card cannot be revoked since there isn't any piece of information about the lost smart card that is stored on the server side.

Poor repairability. Xie's scheme is not easily repairable when the user re-registers with the server after the user finds that smart card has been lost and the critical parameter N_i is somehow obtained by an adversary. As described in the previous sections, there are a variety of ways that may lead to the leakage of N_i once the user's card is lost, such as the compromise of the U_i's password or reverse engineering. Impersonation attacks cannot be restricted and stopped even if U_i has found that her card has been out of control and then re-registers with S. As the value of N_i is determined only by U_i 's identifier ID_i and S's permanent secret key x, S cannot change N_i for U_i unless ID_i or x can be changed. However, since x is commonly used for all users rather than specifically used for only U_i, it is unreasonable and inefficient if x should be changed to recover the security of U_i only. On the other hand, it is also impractical to change ID_i, which may be tied to U_i in most application systems.

Poor efficiency. In Xie's scheme, the computation cost on user side and server side are $\mathcal{O}(4T_E + 4T_H + 2T_X)$ and $\mathcal{O}(4T_E + 3T_H + T_X)$, respectively. With a total computation cost of $\mathcal{O}(8T_E + 7T_H + 3T_X)$, Xie's scheme is among the most inefficient schemes that have ever been proposed. Hence, Xie's scheme is not comparatively efficient.

4 Review and cryptanalysis of Li's scheme

In the above-analyzed scheme, the feature of user un-traceability is achieved by using a public key encryption which randomizes the user's real identity in session-variant pseudonym identities. In contrast, the scheme discussed in this section, i.e. Li's scheme [51], adopts a completely difficult strategy: each party updates the user's session-variant pseudonym identity after having authenticated the party on the other end. Though this strategy can indeed support user un-traceability, as we shall show in the following, it

[6] Although the resistance to KCI attack is not among the explicit security goals in [93], we think it is an important factor that should be taken into account.

is highly impractical, for it introduces a serious vulnerability that greatly downgrade the usability of the scheme. Moreover, this scheme fails to provide two-factor security which is the most essential goal a two-factor shall achieve.

4.1 Review of Li's scheme

In [51], Li proposed two ECC-based two-factor authentication schemes, one without user anonymity and one with user anonymity. Here we are only interested in the one with user anonymity. This scheme consists of four phases: the registration, the authentication, password update and user eviction. For clarity, the intuitive abbreviations used are listed in Table 4, and some additional notations are illustrated in Table 5.

Table 5. Additional notations used in Li's scheme

Symbol	Description
ID_A	identity of user A
PW_A	password of user A
d_s	secret key of remote server S
O	the point at infinity
G	base point of the elliptic curve group of order n such that $n \cdot G = O$
U_S	public key of remote server S, where $U_S = d_S \cdot G$
K_x	secret key computed using $K = PW_A \cdot r_A \cdot U_s = (K_x, K_y)$
$E_{K_x}(\cdot)/D_{K_x}(\cdot)$	symmetric encryption/decryption with K_x

Registration phase The registration phase involves the following operations:

1) User A chooses her identity ID_A, password PW_A and $r_A \in_R \mathbb{Z}_n^*$, and computes $U_A = PW_A \cdot r_A \cdot G$;

2) $A \Rightarrow S : \{ID_A, U_A\}$.

3) On receiving the registration message, the server S selects a pseudonym identity IND_A for A, and creates an entry (IND_A, U_A, *status-bit*) in its database, where *status-bit* indicates the status of A. More specifically, when A has logged-in to S, the *status-bit* is set to one, otherwise it is set to zero.

4) $S \Rightarrow U_i$: A smart card containing parameters $\{IND_A, G, h(\cdot), E_{K_x}(\cdot)/D_{K_x}(\cdot)\}$.

5) Upon receiving the smart card, user A keys her password PW_A into the card;

Authentication phase When U_i wants to login to S, the following steps are involved:

1) U_i inserts her smart card into the card reader, and inputs PW_A.

2) The smart card retrieves r_A, generates $r'_A \in_R \mathbb{Z}_n^*$, computes $R_A = r_A \cdot U_S = r_A \cdot d_S \cdot G$, $W_A = r_A \cdot r_A \cdot PW_A \cdot G$, $U'_A = PW_A \cdot r'_A \cdot G$ and $K = PW_A \cdot r_A \cdot U_s = (K_x, K_y)$.

3) $A \rightarrow S : \{IND_A, E_{K_x}(IND_A, R_A, W_A, U'_A)\}$.

4) Upon receiving the login request, S computes the decryption key K_x by computing $K = d_S \cdot U_A = PW_A \cdot r_A \cdot d_s \cdot G = (K_x, K_y)$ and decrypts $E_{K_x}(IND_A, R_A, W_A, U'_A)$ to reveal $\{IND_A, R_A, W_A, U'_A\}$. Then S compares decrypted IND_A with received IND_A and

$\hat{e}(d_S \cdot R_A, U_A)$ with $\hat{e}(W_A, U_S)$, respectively. If either is unequal, S rejects. Otherwise, the server will consider A as a legitimate user, which is justified by the following equalities:

$$\hat{e}(d_S \cdot R_A, U_A) = \hat{e}(d_S \cdot r_A \cdot G, r_A \cdot pw_A \cdot G) = \hat{e}(G, G)^{r_A \cdot r_A \cdot pw_A \cdot d_S}$$

$$\hat{e}(W_A, U_S) = \hat{e}(r_A \cdot r_A \cdot pw_A \cdot G, d_S \cdot G) = \hat{e}(G, G)^{r_A \cdot r_A \cdot pw_A \cdot d_S}$$

5) S proceeds to generates a new pseudonym identity IND_A for A, selects $r_S \in_R \mathbb{Z}_n^*$, computes $W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G$ and the session key $sk = r_S \cdot d_S \cdot W_A$.

6) $S \rightarrow A : \{W_A + W_S, h(W_S \| U'_A \| sk \| IND'_A), E_{sk}(IND'_A)\}$.

7) A derives W_S by subtracting W_A from $(W_A + W_S)$, computes $sk = r_A \cdot r_A \cdot PW_A \cdot W_S$, and gets IND'_A by decrypting $E_{sk}(IND'_A)$ using sk. S checks whether the hashed result of $\{W_S \| U'_A \| sk \| IND'_A\}$ is equal to the received $H(W_S \| U'_A \| sk \| IND'_A)$. If they are equal, A is assured that S is authenticated and replaces $\{r_A, IND_A\}$ with $\{r'_A, IND'_A\}$.

8) $A \rightarrow S : \{IND_A, h(sk \| IND'_A)\}$.

9) S checks whether the hashed result of $\{sk \| IND'_A\}$ equals the received $h(sk \| IND'_A)$. If it holds, S granted A's login request and replaces $\{IND_A, U_A\}$ with $\{IND'_A, U'_A\}$.

Password change phase The password change phase is provided to allow users to change their passwords freely and locally. When A wants to change his password, she needs to first go through the above authentication phase to make sure that the input password is valid and then can changes it to a new one.

User eviction phase In case the period of validity of user A has expired, then the user can be evicted by the remote server S by deleting (IND_A, U_A) from its database. Thereafter, A can no longer use IND_A and U_A to login S.

4.2 Cryptanalysis of Li's scheme

In [51], Li demonstrated that Islam-Biswas's scheme [38] is vulnerable to several serious attacks such as offline password guessing and stolen-verifier, and to overcome the identified weaknesses, a new scheme with user anonymity was further developed. Besides its high efficiency duo to the use of ECC, this scheme is claimed (and heuristically argued) to provide robust security (i.e., provision of SR1∼SR7) and support five attractive properties (i.e., DA1∼DA5). Accordingly, it seems very appealing and shows great application potential. However, after a careful investigation, we find it still far from practical. As demonstrated in the following, it is of poor usability and fails to achieve two-factor security under their non-tamper resistance assumption of the smart cards.

Smart card loss attack. Evidently, the most essential goal of a two-factor authentication scheme is to provide two-factor security, which means a compromise of either the password factor or the smart card factor will not lead to the compromise of the system. As pointed out in [36, 84], to date few schemes have achieved the "precious"

goal of two-factor security. Once more, Li's attempt [51] ends in vain, as we will show how an attacker in possession of a user's smart card can recover the victim's password with the help of an automated procedure.

Suppose an adversary $\mathcal{A}$ has somehow obtained (stolen or picked up) user A's smart card for a relative long period of time (e.g., a few hours), and extracted the secret information $\{IND_A, r_A, U_S\}$ by using side-channel attacks [61, 63, 69] herself (or with recourse to professional labs). Then, $\mathcal{A}$ returns the breached card back to A without A's awareness. Note that, here we have made two assumptions: (1) $\mathcal{A}$ can obtain and breach the victim's card; and (2) $\mathcal{A}$ can return the breached card *without detection*. The former assumption has been common in the literature (e.g., [75,87,94]), while the latter has only recently raised attention [36,84].[7] Recent studies [64,70,77] on the usability of real-life two-factor systems have confirmed that, users do tend to leave their smart card unattended: 54% users have forgot their smart cards in the card reader at least once during the study (i.e., a period of six weeks), which suggests that the latter assumption is also quite practical.

Once user A uses the breached smart card to login, the attacker can intercept A's login request $\{IND_A, E_{K_x}(IND_A, R_A, W_A, U'_A)\}$ and then obtains PW_A as follows:

Step 1. Guesses the value of PW_A to be PW_A^* from the dictionary space $\mathcal{D}$.
Step 2. Computes $K^* = PW_A^* \cdot r_A \cdot U_s = (K_x^*, K_y^*)$, where r_A, U_s are extracted from A's smart card.
Step 3. Derives $\{IND_A^*, R_A^*, W_A^*, U'^*_A\}$ by decrypting the previously intercepted $E_{K_x}(IND_A, R_A, W_A, U'_A)$ using K_x^*;
Step 4. Verifies the correctness of PW_A^* by checking if IND_A^* is equal to the extracted IND_A.
Step 5. Repeats Step 1 $\sim$ 4 of this procedure until the correct value of PW_A is found.

Let $|\mathcal{D}|$ denote the size of space $\mathcal{D}$. The time complexity of the above attacking procedure is $\mathcal{O}(|\mathcal{D}| * (2T_P + T_S))$, where T_P is the running time for ECC point multiplication and T_S the running time for symmetric decryption. In other words, the time for $\mathcal{A}$ to recover U_i's password is linear with $|\mathcal{D}|$, and thus our attack is quite effective. To gain a better grasp of the effectiveness of this attack, we further implement the related operations on common PCs and obtain the corresponding running time (see Table 6), by using the publicly-available, rational arithmetic C/C++ library MIRACL [73]. In practice, due to the inherent limitations of human cognition, user passwords are often memorable shorting strings and hence the password space is very restricted, e.g., $|\mathcal{D}| \leq 10^6$ [13, 26, 91], and it follows that an attacker can complete the above attacking procedure in seconds on a common PC.

Our attack demonstrates that once the smart-card factor is compromised, the remaining password factor can be offline guessed by an automated attacking procedure, which is the so-called offline password guessing attack [7, 10]. Since then, there is no

[7] In [36], Huang et al. cryptanalyze Juang et al.'s scheme and state that "$\mathcal{A}$ can calculate the session key if $\mathcal{A}$ extracts the information in the smart card before the log-in phase". This indicates Huang et al. have implicitly made the assumption that $\mathcal{A}$ can return the breached card without detection.

Table 6. Computation evaluation of related operations on common PCs

Experimental Platform (common PCs)	ECC Point Multiplication T_P(ECC sect163r1 [16])	Symmetric decryption T_S(AES-128)	Hash operation T_H(SHA-1)	Other lightweight operations(e.g.,XOR)
Intel T5870 2.00 GHz	1.226 ms	2.049 µs	2.580 µs	0.011 µs
Intel E5500 2.80 GHz	0.617 ms	0.572 µs	0.753 µs	0.009 µs
Intel i3-530 2.93 GHz	0.508 ms	0.541 µs	0.693 µs	0.008 µs

way to prevent $\mathcal{A}$ from impersonate A to enjoy the system's services/resources, unless A re-registers with the server. This suggests that Li's scheme is intrinsically not a truly two-factor scheme and provides no better security than the original scheme (i.e., Islam-Bswas's scheme [38]).

De-synchronization attack. To provide user un-traceability, a number of dynamic-ID based authentication protocols (e.g., [43, 83, 99]) construct a new pseudo-identity for the user in each session by using cryptographic methods (e.g., public encryption algorithm) during the login process, while the other dynamic-ID based authentication protocols (e.g., [44,55,85,89]) adopt a different strategy: a new user pseudo-identity for the next login request is constructed during the current authentication process. In the latter strategy, it is evident that the new user pseudo-identity (which will be used for the next login request) shall be stored in somewhere on the user side, yet to recognize this user in the following protocol run, the sever also needs to maintain a copy of the new user pseudo-identity after the current protocol run. So the synchronization of this new user pseudo-identity between the user side and the serve side is crucial for their following successful protocol runs. However, there is no easy way to make sure that this synchronization is well maintained. As we will show and discuss in the following, a determined attacker can always somehow break this synchronization and render the user unable to login ever since, which suggests the infeasibility of the latter strategy (i.e., by employing a synchronization mechanism).

Let us see a concrete example. Suppose user A has performed Step 7 of the authentication phase (see Sec. 4.1) and sends $\{IND_A, h(sk\|IND'_A)\}$ to S as specified, which means A has replaced $\{r_A, IND_A\}$ with $\{r'_A, IND'_A\}$ in its card memory. Before $\{IND_A, h(sk\|IND'_A)\}$ reaches S, the attacker $\mathcal{A}$ intercepts this message and alters it to $\{IND_A, X\}$, where X is a randomly selected value. In Step 8 of the authentication phase, S will find $X \neq h(sk\|IND'_A)$, and surely enough, will reject A's login request and refuse to update $\{U_A, IND_A\}$ to $\{IND'_A, U'_A\}$ in its backend database. Consequently, the consistency of the user pseudo-identity between A and S is broken. Thereafter, in the ensuing login requests, A will send IND'_A to S, yet S stores the old IND_A and will always reject A due to the mismatch $IND'_A \neq IND_A$.

The above attack, as summarized in Fig.2, is practically effective, for the attacker is only required to alter a single protocol transcript (i.e., the third message from A to S) and then can completely destroy the "synchronization" between the user and the server, and there is no other expensive operations involved. What's more, it is not

difficult to see that, instead of altering this protocol transcript, $\mathcal{A}$ simply blocks this single protocol transcript might equally attain her end.

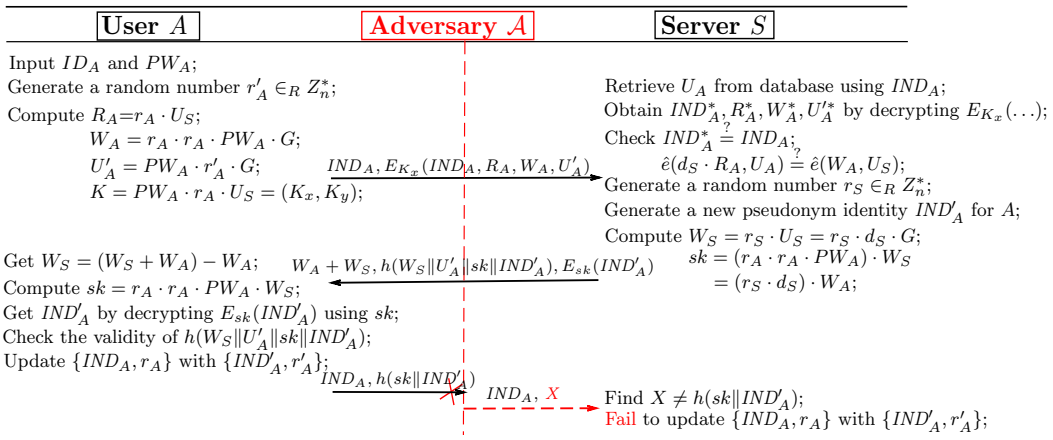

Fig. 2. De-synchronization attack on Li's scheme

It also should be noted that, although this vulnerability appears very simple, it cannot be well addressed just with minor revisions. One may think that, if an additional "ack" message is sent back to A and only when the user A has received this "ack" can she update $\{r_A, IND_A\}$ to $\{r'_A, IND'_A\}$, then the above presented attack will not work. Admittedly, this is true, but what will happen if $\mathcal{A}$ now simply alter (or block) this "ack" message? Apparently, in this case, user A will refuse to replace $\{r_A, IND_A\}$ with $\{r'_A, IND'_A\}$, but the server S has already updated $\{U_A, IND_A\}$ to $\{U'_A, IND'_A\}$ before sending "ack". Similarly, any attempt to overcome this vulnerability by adding new protocol flow(s) will be doomed to failure.

Remark 2. We have cryptanalyzed more than one hundred and twenty recently proposed two-factor schemes and more than thirty dynamic ID-based two-factor schemes (some of our cryptanalysis results include [anonymized]), and observe that all these schemes that employ a similar synchronization mechanism to maintain the consistence of user pseudo-identities are vulnerable to de-synchronization attack without no exception. Some recent ones include [44, 47, 55, 85, 89]. De-synchronization attack can hardly be seen as a new threat to cryptographic protocols, and actually its destructive effects have been intensively investigated in the cryptography community (e.g., RFID authentication protocols [3,53] and multimedia watermarking schemes [6]). Yet, to the best of knowledge, litter attention has been given to this threat in the area of dynamic-ID two-factor authentication, which well explains why vulnerable schemes are proposed wave upon wave and suggests the importance (and urgency) of this work.

5 Some observations on the relationships among evaluation criteria

In this section, we first explicate the definitions of some criteria in Madhusudhan-Mittal's set [59] to provide a clear basis for investigating their inner relationships. Then,

using these two schemes examined earlier in this paper as case studies and on the basis of previous cryptanalysis results [36, 58, 71, 78, 82, 84], we show that it is unlikely to construct an "ideal" dynamic ID-based two-factor scheme that satisfies all the criteria in Madhusudhan-Mittal's set.

5.1 Further explications of some criteria

A scheme supporting DA1 requires that there is no password-related verification data stored on the server, ensuring that a compromise of the server will not lead to the disclosure of all the users' passwords. Since the first scheme with DA1 was proposed by Hwang-Li [37] in 2000, DA1 has become one of the most basic design goals of two-factor schemes [57,80]. A number of schemes attempting to achieve DA1 (e.g., [18,24,47,86,93]) advocate that the remote server only keeps a secret key for verifying the users and there is no other user-specific information stored on the server. On the contrary, the other schemes with DA1 (e.g., [22, 51, 85, 90]) store some non-security-critical user-specific information such as $\{ID_i, T_{reg}\}$ on the server side, where T_{reg} is user U_i's registration time. We say the former kind of schemes provide the property "DA1-Strong", the latter kind of schemes support the property "DA1-Weak".

In password-related authentication, besides freely user password choice (DA2), it is a universally accepted practice that users shall regularly change their passwords [12]. Accordingly, as stated in 2.1, the password change phase has been a basic phase in any two-factor scheme. Obviously, this phase either may enable a user to locally change her password or requires a user to interact with the server in order to change her password. As investigated in Section 3.2, a scheme that facilitates a user to locally change her password but does not support secure password change is prone to smart card loss attack (i.e., no provision of SR6), while a scheme that facilitates a user to locally change her password as well as supporting secure password change is vulnerable to the same threat, too. For ease of presentation, we say the former scheme is with attribute "DA2-Local-Secure", the latter scheme with attribute "DA2-Local-Insecure". In addition, for a scheme that does not support local user password change, we say this scheme provides attribute "DA2-Interactive".

5.2 Relationships among the evaluation criteria

In this work, we mainly focus on the following three most basic relationships among the evaluation criteria: symbiotic, mutually exclusive and implicative, denoted by ∞, $\otimes$ and $\triangleright$, respectively. More specifically, DAi and SRj have a symbiotic relation (i.e., DAi∞SRj) means if either one is held by the scheme, both are held by the scheme; DAi and SRj have a mutually exclusive relation (i.e., DAi$\otimes$ SRj) means at any time, at most one of them is held by the scheme; DAi and SRj have an implicative relation means that either (1) whenever DAi is held by the scheme, SRj is held by the scheme (i.e., DAi$\triangleright$SRj) or (2) whenever SRj is held by the scheme, DAi is held by the scheme (i.e., SRj$\triangleright$DAi). Our observations are summarized in Table 7 and detailed as follows:

Table 7. Relationships among the criteria in Madhusudhan-Mittal's set

Design Goals	DA1: No password verifier table	DA2: Password friendliness	DA3: No password reveal	DA4: Password dependent	DA5: Mutual authentication	DA6: Session key agreement	DA7: Forward secrecy	DA8: User anonymity	DA9: Smart card revocation	DA10: timely typo detection	SR1: Resist to DoS attack	SR2: Resist impersonation attack	SR3: Resist parallel session attack	SR4: Resist password guessing attack	SR5: Resist replay attack	SR6: Resist smart card loss attack	SR7: Resist stolen-verifier attack	SR8: Resist reflection attack	SR9: Resist insider attack
DA1-Strong	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	*	*	▷	*	*
DA1-Weak	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	▷	*	*
DA2-Local-Secure	*	*	*	*	*	*	*	*	*	∞	*	*	*	*	*	⊗	*	*	*
DA2-Local-Insecure	*	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	⊗	*	*	*
DA2-Interactive	*	*	*	*	*	*	*	*	*	⊗	*	*	*	*	*	◁	*	*	*
DA3	*	*	*	*	*	*	*	*	*	*	*	▷	▷	*	▷	*	*	▷	*
DA4	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	◁	*	*	*
DA5	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA6	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA7	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA8	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA9	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
DA10	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	⊗	*	*	*

Notes: X∗Y means the relation between X and Y is unknown (probably independent); X ∞Y means a symbiotic relation between X and Y; X▷Y means Y is implied by X; X⊗Y means a mutually exclusive relation between X and Y.

1) DA1-Strong⊗DA9. A scheme supports the property DA1-Strong means there is no user-specific (or card-specific) information stored on the server. However, even if the authorized time of a card has expired, how can the server (e.g., the schemes in [18, 24, 47, 86, 93]) tell apart a valid card from an expired card? Apparently, there is no way for the server to accomplish this aim.

2) DA1-Strong▷SR7. Since there is actually no verifier stored in the server, of course, stolen-verifier-attack can be prevented. On the contrary, a scheme free from stolen-verifier-attack may support DA1-Weak but not DA1-Strong.

3) DA1-Weak▷SR7. Since there is actually no security-critical verifier stored in the server, stolen-verifier-attack can be eliminated. On the contrary, a scheme free from stolen-verifier-attack may support DA1-Strong but not DA1-Weak.

4) DA2-Local-Secure∞DA10. A scheme supports DA2-Local-Secure means a user can locally and securely update her password, this implies that there are some password-related verifiers (e.g., $A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$ in [54], or $\{b, R = p \times h(b \times PW_i), V = h_p(h(b \times PW_i))\}$ in [20]) stored in the card memory, and these password-related verifiers can just be used to check whether the user has accidentally input a wrong password when login.

5) DA2-Local-Secure⊗SR6. A scheme supports DA2-Local-Secure means a user can locally and securely update her password, this implies that there are some password-related verifiers (e.g., $A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$ in [54], or $\{b, R = p \times h(b \times PW_i), V = h_p(h(b \times PW_i))\}$ in [20]) stored in the card memory, and these password-

related verifiers can just be used to check whether a guessed password is right or not once an attack has obtained the card and extracted these password-related verifiers.

6) DA2-Local-Insecure⊗DA10. This can be obtained directly from the relationship that DA2-Local-Secure∞DA10.

7) DA2-Local-Insecure⊗SR6. A scheme supports DA2-Local-Insecure means a user can locally change her password while there is no password-related verifier stored in the card memory. Apparently, this scheme can not prevent an attacker from easily changing the password, as shown in Section 3.2. On the other hand, a scheme supports SR6 means an attacker can not easily change the password when obtaining the card, indicating DA2-local-insecure is not supported in the scheme.

8) DA2-Interactive⊗DA10. A scheme supports DA2-Interactive (e.g., the schemes in [51,55]) means a user is required to change her password by interacting with the server. This implies that there is no password-related verifier stored on the card. Without such data, there is no information that the card can use to check whether or not the user has accidently input a wrong password when login (i.e. DA10), and and vice versa.

9) DA3▷SR2, SR3, SR5 and SR8. A scheme supports DA3 means an attacker can not impersonate either the user or the server. This excludes the possibility of impersonation attack, parallel session attack, replay attack and reflection attack. Otherwise, mutually authentication (i.e., DA3) can not be assured. On the other hand, neither the achievement of SR2, SR3, SR5 or SR8 indicates the achievement of DA3. For example, a scheme (e.g., [99]) achieves SR8 may still be prone to replay attack, which invalidates DA3.

10) DA10⊗SR6. A scheme supports DA10 (e.g., [47, 54]) means the smart card can timely detect whether the user has accidently input a wrong password when login. To this end, there should be some password-related verifier stored on the card. In this case, an attacker can perform smart card lost attack just by exploiting this data.

11) SR6▷DA4. According to the definitions given in [59], DA4 is entirely incorporated into SR6: both SR6 and DA4 concern the case in which a user has lost her card, yet SR6 requires that $\mathcal{A}$ shall not be able to perform impersonation attack, offline guessing attack or easily change the password (see Section 3.2), while what a scheme supporting DA4 can only guarantee is that $\mathcal{A}$ needs to know the password to impersonate the user (but $\mathcal{A}$ may perform other malicious operations such as easily changing the password). For example, the scheme in [86] is a typical one that fails to achieve DA4, while the scheme in [42] achieves DA4 but fails to provide SR6.

From Table 7, one can see that there is always a mutually exclusive relationship (denoted by ⊗) among "DA2-*" and some other criteria. More specifically, both DA2-Local-Secure and DA2-Local-Insecure are mutually exclusive with SR6, while DA2-Interactive is mutually exclusive with DA10. This means no matter how the user change her password (i.e., either locally or interactively), at least one of SR6 and DA10 can not be achieved, which suggests the impossibility of constructing an "ideal" scheme that satisfies all the criteria in Madhusudhan-Mittal's evaluation set.

6 Conclusion

In this paper, we have investigated the question "whether an 'ideal' scheme that satisfies all the criteria in Madhusudhan-Mittal's evaluation set can be built"? By cryptanalyzing two representative dynamic ID-based two-factor authentication schemes as case studies, we uncover some subtleties and challenges in designing this type of schemes, and explore the relationships among the criteria in Madhusudhan-Mittal's evaluation set. Our results highly indicate a negative answer to the examined question. We believe this work provides a better understanding of the underlying evaluation metric for dynamic ID-based two-factor schemes, which is of fundamental importance for security engineers to make their choices correctly and for protocol designers to advance practical schemes with better usability-security tradeoffs. We leave for future work the question of evaluating practical effectiveness of the proposed "fuzzy-verifiers" by using large-scale real-life password data sets.

References

1. Abdalla, M., Chevassut, O., Pointcheval, D.: One-time verifier-based encrypted key exchange. In: Vaudenay, S. (ed.) PKC 2005, LNCS, vol. 3386, pp. 47–64. Springer Berlin Heidelberg (2005)
2. Adham, M., Azodi, A., Desmedt, Y., Karaolis, I.: How to attack two-factor authentication internet banking. In: Sadeghi, A.R. (ed.) FC 2013, LNCS, vol. 7859, pp. 322–328. Springer-Verlag (2013)
3. Ahmadian, Z., Salmasizadeh, M., Aref, M.R.: Desynchronization attack on rapp ultralightweight authentication protocol. Information processing letters 113(7), 205–209 (2013)
4. Ahn, L., Blum, M., Hopper, N., Langford, J.: Captcha: Using hard ai problems for security. In: Biham, E. (ed.) EUROCRYPT 2003, LNCS, vol. 2656, pp. 294–311. Springer-Verlag (2003)
5. Alemán, J.L.F., Señor, I.C., Lozoya, P.Á.O., Toval, A.: Security and privacy in electronic health records: a systematic literature review. Journal of biomedical informatics 46(3), 541–562 (2013)
6. Alface, P.R., Macq, B.B.: Feature-based watermarking of 3d objects: toward robustness against remeshing and desynchronization. In: Proc. of SPIE. vol. 5681, pp. 400–408 (2005)
7. Bao, F.: Security analysis of a password authenticated key exchange protocol. In: Boyd, C., Mao, W. (eds.) ISC 2003, LNCS, vol. 2851, pp. 208–217. Springer/Berlin Heidelberg (2003)
8. Bao, F., Deng, R.: Privacy protection for transactions of digital goods. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001, LNCS, vol. 2229, pp. 202–213. Springer Berlin / Heidelberg (2001)
9. Barenghi, A., Breveglieri, L., Koren, I., Naccache, D.: Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. Proceedings of the IEEE 100(11), 3056–3076 (2012)
10. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proc. IEEE S&P 1992. pp. 72–84. IEEE (1992)
11. Blake-Wilson, S., Johnson, D., Menezes, A.: Key agreement protocols and their security analysis. In: Darnell, M. (ed.) Crytography and Coding, LNCS, vol. 1355, pp. 30–45. Springer–Verlag (1997)
12. Blythe, J., Koppel, R., Smith, S.W.: Circumvention of security: Good users do bad things. IEEE Security & Privacy 11(5), 80–83 (2013)
13. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proc. IEEE Security & Privacy 2012. pp. 538–552. IEEE Computer Society (2012)
14. Bonneau, J., Just, M., Matthews, G.: Whats in a name? In: Sion, R. (ed.) Proceedings of 14th International Conference on Financial Cryptography and Data Security (FC 2010), LNCS, vol. 6052, pp. 98–113. Springer Berlin/ Heidelberg (2010)
15. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM conference on Computer and communications security (CCS 2011). pp. 125–138. ACM (2011)

16. Certicom Research: Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0 (Jan 2010), available at http://www.secg.org/download/aid-784/sec2-v2.pdf
17. Chang, C.C., Wu, T.C.: Remote password authentication with smart cards. IEE Proceedings-Computers and Digital Techniques 138(3), 165–168 (1991)
18. Chang, Y.F., Tai, W.L., Chang, H.C.: Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. International Journal of Communication Systems (2013), doi: http://dx.doi.org/10.1002/dac.2368
19. Chen, B., Kuo, W., Wuu, L.: Robust smart-card-based remote user password authentication scheme. Int. J. Commun. Syst. (2012), doi:http://dx.doi.org/10.1002/dac.2368
20. Chen, T.H., Hsiang, H.C., Shih, W.K.: Security enhancement on an improvement on two remote user authentication schemes using smart cards. Future Gener. Comput. Syst. 27(4), 377–380 (2011)
21. Chien, H., Chen, C.: A remote authentication scheme preserving user anonymity. In: 19th International Conference on Advanced Information Networking and Applications (AINA'05). vol. 2, pp. 245–248. IEEE (2005)
22. Chung, H., Ku, W., Tsaur, M.: Weaknesses and improvement of wang et al.'s remote user password authentication scheme for resource-limited environments. Computer Standards & Interfaces 31(4), 863–868 (2009)
23. Das, A.K., Bruhadeshwar, B.: An improved and effective secure password-based authentication and key agreement scheme using smart cards for the telecare medicine information system. Journal of medical systems 37(5), 1–17 (2013)
24. Das, M., Saxena, A., Gulati, V.: A dynamic id-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 50(2), 629–631 (2004)
25. Dazzlepod Inc.: CSDN cleartext passwords (Mar 2013), online news, Available at http://dazzlepod.com/csdn/
26. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: Proc. INFOCOM 2010. pp. 1–9. IEEE (2010)
27. Deswarte, Y., Gambs, S.: The challenges raised by the privacy-preserving identity card. In: Naccache, D. (ed.) Cryptography and Security: From Theory to Applications: Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, LNCS, vol. 6805, pp. 383–404. Springer/Berlin Heidelberg (2012)
28. EMVCo Ltd.: About EMV (Europay, MasterCard, and Visa) (Sep 2013), available at http://www.emvco.com/approvals.aspx?id=91
29. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) Proc. CRYPTO 1999, LNCS, vol. 1666, pp. 537–554. Springer-Verlag (1999)
30. Gunson, N., Marshall, D., Morton, H., Jack, M.: User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. Computers & Security 30(4), 208–220 (2011)
31. Hamadouche, S., Lanet, J.L.: Virus in a smart card: Myth or reality? Elsevier Journal of Information Security and Applications 18(2), 130–137 (2013)
32. He, D., Chan, S., Chen, C., Bu, J., Fan, R.: Design and validation of an efficient authentication scheme with anonymity for roaming service in global mobility networks. Wireless Personal Communications 61(2), 465–476 (2011)
33. He, D., Ma, M., Zhang, Y., Chen, C., Bu, J.: A strong user authentication scheme with smart cards for wireless communications. Comput. Commun. 34(3), 367–374 (2011)
34. Hsieh, W., Leu, J.: Exploiting hash functions to intensify the remote user authentication scheme. Computers & Security 31(6), 791–798 (2012)
35. Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.: A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Transactions on Parallel and Distributed Systems 22(8), 1390–1397 (2011)
36. Huang, X., Chen, X., Li, J., Xiang, Y., Xu, L.: Further observations on smart-card-based password-authenticated key agreement in distributed systems. IEEE Transactions on Parallel and Distributed Systems (2013), Doi:http://dx.doi.org/10.1109/TPDS.2013.230

37. Hwang, M.S., Li, L.H.: A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 46(1), 28–30 (2000)
38. Islam, S.H., Biswas, G.P.: Design of improved password authentication and update scheme based on elliptic curve cryptography. Mathematical and Computer Modelling 57(6), 2703–2717 (2013)
39. Jiang, Q., Ma, J., Li, G., Li, X.: Improvement of robust smart-card-based password authentication scheme. International Journal of Communication Systems (2013), doi:http://dx.doi.org/10.1002/dac.2644
40. Kasper, T., Oswald, D., Paar, C.: Side-channel analysis of cryptographic rfids with analog demodulation. In: Kwon, T., Lee, M.K., Kwon, D. (eds.) RFIDSec 2012, LNCS, vol. 7055, pp. 61–77. Springer Berlin / Heidelberg (2012)
41. Katz, J., Ostrovsky, R., Yung, M.: Efficient and secure authenticated key exchange using weak passwords. Journal of the ACM 57(1), 1–41 (2009)
42. Khan, M., Kim, S.: Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme'. Comput. Commun. 34(3), 305–309 (2011)
43. Khan, M.K., He, D.: A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography. Security and Communication Networks 5(11), 1260–1266 (2012)
44. Kim, K.K., Kim, M.H.: An enhanced anonymous authentication and key exchange scheme using smartcard. In: Juels, A., Paar, C. (eds.) Proceedings of the 16th Annual International Conference on Information Security and Cryptology (ICISC 2012), LNCS, vol. 7839, pp. 487–494,. Springer Berlin / Heidelberg (2012)
45. Kim, T.H., Kim, C., Park, I.: Side channel analysis attacks using am demodulation on commercial smart cards with seed. J. Syst. Soft. 85(12), 2899 – 2908 (2012)
46. Krawczyk, H.: HMQV: A high-performance secure diffie-hellman protocol. In: Shoup, V. (ed.) CRYPTO 2005, LNCS, vol. 3621, pp. 546–566. Springer-Verlag (2005)
47. Kumari, S., Khan, M.K.: Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme. International Journal of Communication Systems (2013), doi: http://dx.doi.org/10.1002/dac.2590
48. Kumari, S., Khan, M.K.: More secure smart card-based remote user password authentication scheme with user anonymity. Security and Communication Networks (2013), doi: http://dx.doi.org/10.1002/sec.916
49. Lee, Y., Yang, H., Won, D.: Attacking and improving on lee and chius authentication scheme using smart cards. In: Kwak, J., Deng, R., Won, Y., Wang, G. (eds.) ISPEC 2010, LNCS, vol. 6047, pp. 377–385. Springer Berlin Heidelberg (2010)
50. Li, C.T., Lee, C.C.: A robust remote user authentication scheme using smart card. Information Technology And Control 40(3), 236–245 (2011)
51. Li, C.T.: A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Information Security 7(1), 3–10 (2013)
52. Li, S., Shah, S., Khan, M., Khayam, S.A., Sadeghi, A.R., Schmitz, R.: Breaking e-banking captchas. In: Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC 2010). pp. 171–180. ACM (2010)
53. Li, T.Y., Wang, G.L.: Security analysis of two ultra-lightweight rfid authentication protocols. In: Venter, H., Eloff, M., Labuschagne, L., Eloff, J., Solms, R. (eds.) SEC 2007, IFIP AICT, vol. 232, pp. 109–120. Springer-Verlag (2007)
54. Li, X., Niu, J., Khan, M.K., Liao, J.: An enhanced smart card based remote user password authentication scheme. Journal of Network and Computer Applications (2013), doi: http://dx.doi.org/10.1016/j.jnca.2013.02.034
55. 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)
56. Li, X., Xiong, Y., Ma, J., Wang, W.: An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications 35(2), 763–769 (2012)

57. Liao, I., Lee, C., Hwang, M.: A password authentication scheme over insecure networks. Journal of Computer and System Sciences 72(4), 727–740 (2006)
58. Ma, C.G., Wang, D., Zhao, S.D.: Security flaws in two improved remote user authentication schemes using smart cards. Int. J. Commun. Syst. (2013), doi: http://dx.doi.org/10.1002/dac.2468
59. Madhusudhan, R., Mittal, R.: Dynamic id-based remote user password authentication schemes using smart cards: A review. Journal of Network and Computer Applications 35(4), 1235–1248 (2012)
60. Mangipudi, K., Katti, R.: A secure identification and key agreement protocol with user anonymity (SIKA). Computers & Security 25(6), 420–425 (2006)
61. 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)
62. Mishra, D.: Cryptanalysis of sun and cao's remote authentication scheme with user anonymity. CoRR abs/1310.6422:1-8 (2013), http://arxiv.org/abs/1310.6422
63. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx Virtex-II FPGAs. In: Proc. ACM CCS 2011. pp. 111–124. ACM, New York, NY, USA (2011)
64. Morse, E., Theofanos, M., Choong, Y., Paul, C., Zhang, A.: NIST-IR-7867 usability of piv smartcards for logical access. Tech. rep., National Institute of Standards and Technology, McLean, VA (2012), doi:http://dx.doi.org/10.6028/NIST.IR.7867
65. Murdoch, S.J., Drimer, S., Anderson, R., Bond, M.: Chip and pin is broken. In: Proc. IEEE Security & Privacy 2010. pp. 433–446. IEEE Computer Society (2010)
66. Nam, J., Kim, S., Won, D.: Security analysis of a nonce-based user authentication scheme using smart cards. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 90(1), 299–302 (2007)
67. Nguyen, V., Chow, Y.W., Susilo, W.: Breaking an animated captcha scheme. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012, LNCS, vol. 7341, pp. 12–29. Springer Berlin Heidelberg (2012)
68. Nohl, K., Evans, D., Starbug, S., Plötz, H.: Reverse-engineering a cryptographic rfid tag. In: Proc. USENIX Security 2008. pp. 185–193. USENIX Association (2008)
69. 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/Berlin Heidelberg (2011)
70. Paul, C., Morse, E., Zhang, A., Choong, Y.Y., Theofanos, M.: A field study of user behavior and perceptions in smartcard authentication. In: Campos, P., Graham, N., Winckler, M. (eds.) INTERACT 2011, LNCS, vol. 6949, pp. 1–17. Springer Berlin Heidelberg (2011)
71. Scott, M.: Cryptanalysis of a recent two factor authentication scheme. Cryptology ePrint Archive, Report 2012/527 (2012), http://eprint.iacr.org/2012/527.pdf
72. Sean, P.: LinkedIn Passwords Leaked Online: Hackers are beginning to decrypt 6.4 million passwords (June 6 2012), available at http://www.webpronews.com/linkedin-passwords-leaked-online-2012-06
73. Shamus Software Ltd.: Miracl library (May 2013), http://www.shamus.ie/index.php?page=home
74. Silva, S.S., Silva, R.M., Pinto, R.C., Salles, R.M.: Botnets: A survey. Computer Networks 57(2), 378–403 (2013)
75. Song, R.: Advanced smart card based password authentication protocol. Computer Standards & Interfaces 32(5), 321–325 (2010)
76. Sood, S., Sarje, A., Singh, K.: A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network and Computer Applications 34(2), 609–618 (2011)
77. Strouble, D.D., Schechtman, G., Alsop, A.S.: Productivity and usability effects of using a two-factor security system. Proceedings of SAIS, pp. 196–201 (2009)
78. Sun, D.Z., Cao, Z.F.: On the privacy of khan et al.'s dynamic id-based remote authentication scheme with user anonymity. Cryptologia 37(4), 345–355 (2013)
79. Tang, C., Wu, D.: Mobile privacy in wireless networks-revisited. IEEE Transactions on Wireless Communications 7(3), 1035 –1042 (march 2008)
80. Tsai, C., Lee, C., Hwang, M.: Password authentication schemes: current status and key issues. International Journal of Network Security 3(2), 101–115 (2006)

81. Tsai, J., Wu, T., Tsai, K.: New dynamic id authentication scheme using smart cards. International Journal of Communication Systems 23(12), 1449–1462 (2010)
82. Wang, D., Ma, C.G., Zhao, S., Zhou, C.: Cryptanalysis of two dynamic id-based remote user authentication schemes for multi-server architecture. In: Xu, L., Bertino, E., Mu, Y. (eds.) Proceedingg of the 6th International Conference on Network and System Security (NSS 2012), LNCS, vol. 7645, pp. 462–475. Springer Berlin / Heidelberg (2012)
83. Wang, D., Ma, C., Wang, P., Chen, Z.: Robust smart card based password authentication scheme against smart card security breach. Cryptology ePrint Archive, Report 2012/439 (2012), http://eprint.iacr.org/2012/439.pdf
84. Wang, D., Wang, P.: Offline dictionary attack on password authentication schemes using smart cards. In: Desmedt, Y., Thuraisingham, B., Hamlen, K. (eds.) ISC 2013, pp. 1–16. LNCS, Springer-Verlag (2013), http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/isc2013.pdf
85. Wang, R.C., Juang, W.S., Lei, C.L.: Robust authentication and key agreement scheme preserving the privacy of secret key. Computer Communications 34(3), 274–280 (2011)
86. Wang, Y., Liu, J., Xiao, F., Dan, J.: A more efficient and secure dynamic id-based remote user authentication scheme. Computer communications 32(4), 583–585 (2009)
87. Wang, Y.G.: Password protected smart card and memory stick authentication against off-line dictionary attacks. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012, IFIP AICT, vol. 376, pp. 489–500. Springer Boston (2012)
88. Wen, F., Li, X.: An improved dynamic id-based remote user authentication with key agreement scheme. Computers & Electrical Engineering 38(2), 381–387 (2012)
89. Wen, F., Susilo, W., Yang, G.: A secure and effective anonymous user authentication scheme for roaming service in global mobility networks. Wireless Pers. Commun. (2013), doi: http://dx.doi.org/10.1007/s11277-013-1243-4
90. Wu, S.H., Zhu, Y.F., Pu, Q.: Robust smart-cards-based user authentication scheme with user anonymity. Security and Communication Networks 5(2), 236–248 (2012)
91. Wu, T.: A real-world analysis of kerberos password security. In: Proc. NDSS 1999. pp. 13–22. Internet Society (1999)
92. Xiang, T., Wong, K., Liao, X.: Cryptanalysis of a password authentication scheme over insecure networks. Journal of Computer and System Sciences 74(5), 657–661 (2008)
93. Xie, Q.: Dynamic id-based password authentication protocol with strong security against smart card lost attacks. In: Snac, P., Ott, M., Seneviratne, Aruna, e.a. (eds.) Proceedings of IET International Conference on Wireless Communications and Applications, LNICST, vol. 72, pp. 412–418. Springer Berlin / Heidelberg (2012)
94. Xu, J., Zhu, W., Feng, D.: An improved smart card based password authentication scheme with provable security. Comput. Stand. & Inter. 31(4), 723–728 (2009)
95. Xue, K., Hong, P., Ma, C.: A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J. Comput. System Sci. (2013), doi: http://dx.doi.org/10.1016/j.jcss.2013.07.004
96. 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)
97. Yeh, H., Sun, H., Yang, C., Chen, B., Tseng, S.: Improvement of password authenticated key exchange based on rsa for imbalanced wireless networks (fundamental theories). IEICE transactions on communications 86(11), 3278–3282 (2003)
98. Yeh, K.H., Su, C., Lo, N.W., Li, Y., Hung, Y.X.: Two robust remote user authentication protocols using smart cards. Journal of Systems and Software 83(12), 2556–2565 (2010)
99. Zhou, T., Xu, J.: Provable secure authentication protocol with anonymity for roaming service in global mobility networks. Computer Networks 55(1), 205–213 (2011)