This paper is a full version of our work that has appeared at IEEE Transactions on Dependable and Secure Computing, 2014,
http://dx.doi.org/10.1109/TDSC.2014.2355850

Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment

Ding Wang *Student Member, IEEE*, Debiao He, Ping Wang, Chao-Hsien Chu *Senior Member, IEEE*

Abstract—Despite two decades of intensive research, it remains a challenge to design a practical anonymous two-factor authentication scheme, for the designers are confronted with an impressive list of security requirements (e.g., resistance to smart card loss attack) and desirable attributes (e.g., local password update). Numerous solutions have been proposed, yet most of them are shortly found either unable to satisfy some critical security requirements or short of a few important features. To overcome this unsatisfactory situation, researchers often work around it in hopes of a new proposal (but no one has succeeded so far), while paying little attention to the fundamental question: whether or not there are inherent limitations that prevent us from designing an "ideal" scheme that satisfies all the desirable goals?

In this work, we aim to provide a definite answer to this question. We first revisit two foremost proposals, i.e. Tsai et al.'s scheme and Li's scheme, revealing some subtleties and challenges in designing such schemes. Then, we systematically explore the inherent conflicts and unavoidable trade-offs among the design criteria. Our results indicate that, under the current widely accepted adversarial model, certain goals are beyond attainment. This also suggests a negative answer to the open problem left by Huang et al. in 2014. To the best of knowledge, the present study makes the first step towards understanding the underlying evaluation metric for anonymous two-factor authentication, which we believe will facilitate better design of anonymous two-factor protocols that offer acceptable trade-offs among usability, security and privacy.

Index Terms—Two-factor authentication, user anonymity, offline dictionary attack, de-synchronization attack, smart card loss attack.

1 Introduction

Password authentication with smart card is one of the most convenient and effective two-factor authentication mechanisms in distributed systems, and it assures 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 [1], [2], such as e-banking, e-government and e-health, there are severe challenges regarding security [3], privacy [4] and usability [5] due to the open and complex nature of distributed systems, as well as the resource-constrained characteristics of mobile devices.

In 1999, Yang and Shieh [6] introduced the first smart-card-based password authentication scheme without a sensitive verification table stored on the server, which is a key advantage of two-factor schemes over common password-only schemes, for the latter (e.g., [7], [8]) have to maintain a sensitive password (or salted password) table on the server. Once this table is leaked, the entire system collapses. The feature of no password-related table on the server is highly appealing when considering the unending catastrophic leakages of millions of user accounts in prominent service providers [9], [10] and the prevalence of zero-day attacks like the recent "Heartbleed" [11].

Since the seminal work of Yang and Shieh, there have been a great number of two-factor schemes suggested, and some notable ones include [12]–[15]. In most of the previous two-factor schemes, user's identity is transmitted in plaintext over public networks during the login process, which may leak the identity of the logging user once the login transcripts are 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 attacker 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, even age and gender, etc., can be learned and abused for marketing purposes, typically facilitating annoying advertizement flooding. 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 [16]. To address such static-user-ID-related issues, a feasible approach is to adopt the "dynamic ID technique" [17]: the user's real identity is concealed in session-variant pseudo-identities. And schemes employing this technique are known as "dynamic ID-based" or "anonymous" schemes.

In 2004, Das [17] introduced the first anonymous two-factor authentication scheme to preserve user privacy. Das's work has been followed by a number of proposals [18]–[20] 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 about smart cards, i.e., they simply assume that the security parameters stored in the smart card cannot be extracted. However, recent research results have demonstrated that common commercial smart cards shall no longer be considered to be fully tamper-proof: the secret information stored in the smart cards memory could be revealed by power analysis [21], [22], reverse engineering techniques [23], [24] or fault injection attacks [25]. As a consequence, such schemes

- *D. Wang and P. Wang are with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, P.R. China (Email: wangdingg@mail.nankai.edu.cn; pwang@pku.edu.cn).*
- *D. He is with the School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China. Email: hedebiao@163.com*
- *C.H. Chu is with the College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802 USA. Email: chu@ist.psu.edu*

Manuscript received 17 April 2014; Accepted 24 Aug. 2014.

based on the tamper-resistance assumption about the smart cards are susceptible to some types of attacks such as user impersonation attack and offline dictionary attack, once an adversary has breached the smart card. Therefore, it is more prudent and desirable 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 anonymous schemes [26]–[29] based on *non*-tamper-resistance assumption about the smart cards are put forward, and each is claimed to meet a self-imposed list of ambitious design goals.

1.1 Motivations

More often than not, the proponents assert the superiorities of their scheme, while (perhaps subconsciously) ignoring the features 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 imminent issue, in 2012, Madhusudhan and Mittal [30] 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., [13], [27], [31], [32]) and as far as we know, it is so far the most explicit, comprehensive and systematic criteria set for evaluating anonymous two-factor authentication schemes. In Madhusudhan-Mittal's work [30], it is concluded that all existing anonymous two-factor schemes are far from ideal and each has its own pros and cons, and 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., [20], [33]–[35]), followed by cryptanalysis reports (e.g., [36]–[38]), which, once again, falls into the unsatisfactory "attack-fix-attack-fix" cycle (see Fig.1 of [39]). In this vicious cycle, protocol designers work around the above problem by presenting "improved" schemes but with not much confidence, while cryptanalysts respond to the above problem with concrete rebuttals to new proposals, yet no one pays attention to the underlying question: Whether a particular scheme is flawed due to improper design or whether there are some inherent limitations of this type of schemes that prevent us from designing "an ideal scheme"? Or equally, this question can be expressed as: Whether is it *possible* to construct an ideal scheme which satisfies all the design goals listed in [30]?

As far as we know, Huang et al.'s work [40] may be the closest to what we will discuss in the current paper, however, it mainly deals with security threats and challenges in two-factor authentication and leaves over another interesting open problem as to "whether or not there exist secure smart-card-based password authentication protocols and the password-changing phase does not need any interaction with the server"?

Without these two fundamental questions addressed, 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.

1.2 Contributions

This study aims to provide definite answers to the above two questions. We first revisit the security and attribute provisions of two recent proposals, namely Tsai et al.'s scheme [36] and Li's scheme [41], and reveal some challenges and subtleties in designing anonymous two-factor schemes. These two schemes are among the foremost ones and claimed to be secure against various known attacks and to provide many admired features, yet as we will show, once again, they both fail to accommodate some important requirements. Remarkably, we figure out the fundamental flaw in their formal security proofs and highlight two practical threats, i.e. smart card loss attack and de-synchronization attack, the latter of which can be specially targeted at anonymous 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 [30], showing some inherent conflicts and unavoidable trade-offs in designing anonymous two-factor authentication schemes. Our results highly indicate that, under the current widely accepted adversarial model, certain goals are beyond attainment and therefore "an ideal scheme" is intrinsically out of reach. In particular, the revealed security-usability conflict also suggests a negative answer to the open problem left in [40]. To the best of knowledge, this study makes the first step toward exploring the inner relationships of evaluation criteria for anonymous two-factor authentication, which we believe will provide a much better understanding of how to design two-factor protocols that offer acceptable trade-offs among usability, security and privacy.

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

2 System Architecture, Adversarial Model and Evaluation Criteria

In this section, we elaborate on the system architecture, adversarial model and evaluation criteria. It is worth noting that, these three elements are key factors in determining whether a scheme has been evaluated systematically and fairly. There have been hundreds of papers dealing with smart-card-based password authentication quite recently (e.g., [15], [18], [19], [26], [42]), yet as far as we know, only a few ones [13], [15], [43] 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 architecture, define the currently widely accepted adversarial model and introduce Madhusudhan-Mittal's criteria set [30].

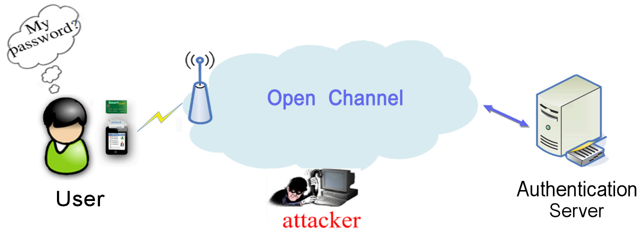

Fig. 1. Smart-card-based password authentication

2.1 System architecture

In this work, as with [39], [40], we mainly focus on the most general case of smart-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 consists of three basic phases, i.e. registration, authentication and password change, as well as some supplementary phases like eviction and revocation [29]. 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 ensure is that, only the user who possesses both a valid smart card and the corresponding password can be successfully verified by 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 revoke a lost card, admired schemes may also provide additional phases such as eviction phase and revocation phase, respectively.

2.2 Adversarial model

In the conventional password authenticated key exchange (PAKE) protocols (e.g., [7], [8]), 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 [44]. 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 [21], [23]–[25]. In addition, 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 [15] and further investigated in [39], $\mathcal{A}$ 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 attacker can somehow know the victim's identity when having obtained the victim's card. Firstly, user's identity is static and generally confined to a predefined structure, and thus it is of little cryptographic strength [45] and can be easily guessed. Secondly, it probably can be harvested from popular forums and other open resources. Thirdly, users used to the idea of keeping passwords a secret would not normally be expecting to keep their identities a secret as well [46], e.g., writing their identities directly on the card. After all, $\mathcal{A}$ can learn more or less about the personal information of the card holder once she has gained access to 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 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 $\mathcal{A}$ endeavors to determine from the publicly available protocol transcripts.

TABLE 1
Capabilities of the adversary

C-01	The adversary $\mathcal{A}$ can enumerate offline all the items 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 protocol participants.
C-2	The adversary $\mathcal{A}$ may either (i) learn the password of a legitimate user via malicious card reader, or (ii) extract the sensitive parameters in the card memory 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 adversarial model presented here is based on the models introduced in [39], [40], [43]. The only (and key) difference is that in our model, we for the first time explicitly define the adversary $\mathcal{A}$'s capabilities related to user identity from both the security perspective and the privacy perspective. This separation enables us to specifically deal with *anonymous* two-factor schemes.[1] Otherwise, some effective attacks specifically aiming at anonymous two-factor schemes can never be captured, such as the smart card loss attack presented in [46] and the offline password guessing attack demonstrated in [47].

[1] In common two-factor schemes, user identities are transmitted in plain-text over the channel (i.e., without consideration of user privacy).

TABLE 2
Security requirements

SR1	Resistance to DoS attack
SR2	Resistance to impersonation attack
SR3	Resistance to parallel session attack
SR4	Resistance to password guessing attack
SR5	Resistance to replay attack
SR6	Resistance to smart card loss attack;
SR7	Resistance to stolen-verifier attack
SR8	Resistance to reflection attack
SR9	Resistance to 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

2.3 Evaluation criteria

In 2012, Madhusudhan and Mittal [30] pointed out that earlier criteria sets, e.g. [31], [32], have ambiguities and redundancies, and thus they developed a new criteria set of nine security requirements (see Table 2) and ten desirable attributes (see Table 3) to evaluate the goodness of anonymous schemes. This criteria set is a refinement of earlier criteria sets, and interested readers are referred to [30] for the specific definition of each criterion. Here we only point out some subtleties, and the inner relationships among the criteria will be investigated in Section 5.

To be called "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 [30] analyzed six recently proposed anonymous schemes and found none of existing ones can satisfy all the above nineteen criteria. Accordingly, they concluded that it remains an open problem to construct a scheme that can be considered ideal.

Madhusudhan-Mittal's criteria set is superior to other proposed criteria mainly for the following two reasons: (1) The security requirements of their criteria set are based on the non-tamper-resistance assumption about 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 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. [13], [27], [31], [32]) in this study.

For a better comprehension, we address two subtleties (which are not made clear in the original work [30]) 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 involves two aspects, i.e. identity protection and user un-traceability [26], [48]. Both are defined against the public (eavesdropping attackers) rather than the server [49], 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 CRYPTANALYSIS OF TSAI ET AL.'S SCHEME

In 2013, Tsai et al. [36] showed some severe security pitfalls in Li et al.'s scheme [26] and proposed a new anonymous two-factor protocol based on Elliptic Curve Cryptosystem (ECC). It is claimed that their new scheme achieves user untraceability while being secure against general threats. In this work, however, we will demonstrate that, under their own assumptions, Tsai et al.'s scheme [36] actually cannot provide truly two-factor security, which is the most critical goal that any two-factor scheme shall attain. In addition, as the result of overlooking an inherent usability-security trade-off, this scheme is subject to smart card loss attack.

3.1 Review of Tsai et al.'s scheme

In this section, we briefly describe the two-factor scheme proposed by Tsai et al. [36] in 2013. Their scheme has two versions: one with fingerprint information involved and the other not. The biometric-involved version is essentially a three-factor scheme, yet Tsai et al. never discussed the issues and challenges incurred by introducing the third authentication factor (i.e., the biometric). Based on Huang et al.'s work [14], one can easily find that the three-factor version of Tsai et al.'s scheme is likely to be prone to the issues of biometric error-tolerance and non-trusted devices. What's more, the non-biometric-involved version constitutes the basis of the biometric-involved version. Hence, we mainly focus on the two-factor version. Their scheme consists of five phases, namely, parameter generation, registration, pre-computation, login and password update. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 4.

TABLE 4
Notations and abbreviations

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
x	the secret key of remote server S
p	a large prime number
O	the point at infinity
P	base point of the elliptic curve E_p where $n \cdot P = O$
$h(\cdot), h_1(\cdot)$	common collision free one-way hash functions
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$\rightarrow$	a common communication channel
$\Rightarrow$	a secure communication channel

3.1.1 Parameter Generation Phase

First of all, the server S chooses an elliptic curve E_p over a finite filed $\mathbb{F}_p$, a base point G with order n and its private key x. Then, S computes $P_S = x \times P$ as its public key, selects two hash functions $h(\cdot) : \{0,1\}^* \to \{0,1\}^k$ and $h_1(\cdot) : \{0,1\}^* \to \{0,1\}^{k'}$, where $|k| = |x|$ and $|k'| \geq |k|$, e.g., $|k| = |x| = 128$ bits, $|k'| = 256$ bits. In the end, S publishes security parameters $\{p, E_p, G, P_S, n, h(\cdot), h_1(\cdot)\}$.

Remark 1. The original specification in [36] do not specify the length of k', which we think is a key factor in conducting formal security reductions, especially in the random oracle model. Hence, we assume that $|k'| \geq |k|$ according to the security reductions in [36]. Moreover, the parameter $h_1(\cdot)$ is not published in the original specification. As the user needs $h_1(\cdot)$ to compute her authenticator in the login phase, $h_1(\cdot)$ should have also been published. Consequently, we deduce a typo has occurred here.

3.1.2 Registration phase.

As mentioned earlier, we hereafter only focus on the version with no biometric factor involved. Whenever U_i wants to register in S, the following operations are performed:

1) User U_i chooses her identity ID_i, password PW_i and a random number b.

2) $U_i \Rightarrow S : \{ID_i, h(PW_i \| b)\}$.

3) On receiving the registration message from user U_i, the server S computes $V = h(ID_i \| x) \oplus h(PW_i \| b)$.

4) $S \Rightarrow U_i$: A security parameter V.

5) Upon receiving V, U_i keys b and V into the smart card.

3.1.3 Pre-Computation Phase

The smart card selects a random number N_{C1}, computes $e = N_{C1} \times P$ and $c = N_{C1} \times P_S$, and then stores $\{e, c, N_{C1}\}$ into its memory. This phase is executed only after the session key has been successfully established between U_i and S. In this way, the smart card is released from wasting time to do these computations in the next login.

3.1.4 Login phase

When U_i wants to access S, the following steps proceed:

1) U_i retrieves $h(ID_i \| x)$ by computing $h(ID_i \| x) = V \oplus h(PW_i \| b)$, selects a random number N_{C2}, then calculates $C_1 = (ID_i \| (h(ID_i \| x) \oplus N_{C2})) \oplus h_1(c)$.

2) $U_i \to S : \{C_1, e\}$.

3) S retrieves ID_i and $h(ID_i \| x) \oplus N_{C2}$ by computing $ID_i \| (h(ID_i \| x) \oplus N_{C2}) = C_1 \oplus h_1(x \times e)$. Next, S retrieves N_{C2} by computing $N_{C2} = h(ID_i \| x) \oplus N_{C2} \oplus h(ID_i \| x)$. Then, S selects a random number N_S, computes $C_2 = N_S \times P$, $SK = h(e \| C_2 \| N_S \times e) = h(e \| C_2 \| N_S \times N_{C1} \times P)$ and $C_3 = h(h(ID_i \| x) \| N_{C2} \| C_2 \| e \| SK)$.

4) $S \to U_i : \{C_2, C_3\}$.

5) On receiving the response from S, U_i computes $SK = h(e \| C_2 \| N_{C1} \times C_2) = h(e \| C_2 \| N_{C1} \times N_S \times P)$ and $C_3' = h(h(ID_i \| x) \| N_{C2} \| C_2 \| e \| SK)$. Then, U_i checks whether the received C_3 equals C_3'. If they are not equal, U_i rejects. Otherwise, U_i computes $C_4 = h(h(ID_i \| x) \| SK \| N_{C2} \| C_2 \| e)$.

6) $U_i \to S : \{C_4\}$.

7) S first computes $C_4' = h(h(ID_i \| x) \| SK \| N_{C2} \| C_2 \| e)$ and then compares whether C_4' equals the received C_4. If it holds, U_i is rejected. Otherwise, S grants U_i's request.

3.1.5 Password change phase

This phase is provided to allow users to change their passwords freely and locally. When U_i wants to change her password, she inserts her smart card into a terminal, keys her old password PW_i and the new one PW_i^{new}. Then, the card computes $V^{new} = V \oplus h(PW_i \| b) \oplus h(PW_i^{new} \| b) = h(ID_i \| x) \oplus h(PW_i^{new} \| b)$ and replaces V with V^{new}.

3.2 Cryptanalysis of Tsai et al.'s scheme

Tsai et al. present five kinds of adversarial models and provide formal security proofs for their scheme under each model. It is not difficult to see that, their five adversarial models have been included into our model as described in Section 2.2. Although their scheme exhibits many attractive properties over existing schemes, such as user un-traceability, high efficiency and formal security proofs, it is still far from an "ideal" anonymous two-factor protocol to be applicable for practical applications. In this section we will show that, it actually fails to resist smart card loss attack (i.e., SR6) under their own assumptions and is prone to smart card revocation problem (i.e., DA9). Now *a paradoxical question arises*: How can a protocol that was formally proven secure later be found insecure? We further explicate this paradox by showing that its security proofs are fallacious.

3.2.1 Smart card loss attack I

What a two-factor protocol with resistance to smart card loss attack (i.e., SR6) can guarantee is that when user' smart card is lost (or stolen), there is no way for an adversary $\mathcal{A}$ to easily change the user's password, guess the password of the user using password guessing attacks, or impersonate the user to freely enjoy the services. What concerns us is the realistic possibilities that users lose their smart cards. Recent studies [50], [51] on the usability of real-life two-factor systems have confirmed that, users do tend to leave their smart card unattended: 54% users have forgotten their smart cards in the card reader at least once during the study (i.e., a period of merely six weeks). Therefore, SR6 is a basic goal that any practical scheme shall attain. Unfortunately, the past research (e.g, [39], [40]) has proved that achieving this goal is notoriously difficult. In the following, we show that, once more, Tsai et al.'s attempt ends in vain — $\mathcal{A}$ can obtain the user's password by an offline password guessing attack once the user's smart card is in the possession of $\mathcal{A}$.

Suppose an adversary $\mathcal{A}$ has somehow obtained (stolen or picked up) user U_i's smart card for a relative long period of time (e.g., a few hours), and extracted the secret information $\{e, c, N_{C1}, V, b\}$ by using side-channel attacks [23]–[25] herself (or with recourse to professional labs). Then, $\mathcal{A}$ returns the breached card back to U_i without U_i's awareness. Once user U_i uses the breached smart card to login, the attacker can intercept U_i's login request $\{C_1, e, C_2, C_3\}$ and then obtains PW_i as follows:

Step 1. Guesses the value of PW_i to be PW_i^* from the dictionary space $\mathcal{D}_{pw}$.

Step 2. Computes $h(ID_i \| x)^* = V \oplus h(PW_i^* \| b)$, where V, b is extracted from U_i's smart card.

Step 3. Retrieves $h(ID_i\|x) \oplus N_{C2}$ by computing $ID_i\|(h(ID_i\|x)\oplus N_{C2}) = C_1 \oplus h(c)$, where c is extracted from U_i's smart card and C_1 is intercepted from the public channel.

Step 4. Computes $N^*_{C2} = (h(ID_i\|x) \oplus N_{C2}) \oplus h(ID_i\|x)^*$, $SK = h(e\|C_2\|N_{C1} \times C_2)$ and $C^*_3 = h(h(ID_i\|x)^*\| N^*_{C2}\| \ C_2\|e\|SK)$.

Step 5. Verifies the correctness of PW^*_i by checking if the computed C^*_3 is equal to the intercepted C_3.

Step 6. Repeats Step $1 \sim 5$ of this procedure until the correct value of PW_i is found.

In the above attack, 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., [13], [15], [52]) and it is also explicitly made in Tsai et al.'s work [36], while the latter has only recently raised attention [39], [40],[2] and it is indeed reasonable. For example, an employee accidentally leaves her bank card on her desk after work, $\mathcal{A}$ picks this card and executes the side-channel attacks herself (or asks help from professional labs) in the evening and sends it back before the victim comes to work the next morning. The victim will find no abnormality and use this card as usual. Unfortunately, once this breached card is put to use, the corresponding password might be disclosed as illustrated above. In a nutshell, these two assumptions are realistic, and the presented attack is indeed practical.

Let $|\mathcal{D}_{pw}|$ denote the number of passwords in $\mathcal{D}_{pw}$. The time complexity of the above attacking procedure is $\mathcal{O}(|\mathcal{D}_{pw}| * (T_P + 4T_H + T_X))$, where T_P is the running time for ECC point multiplication, T_H the running time for Hash function, T_X the running time for bitwise XOR operation. It is easy to see that, the time for $\mathcal{A}$ to recover U_i's password is a linear function of the password space size. And hence 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 operation timings (see Table 5), by using the publicly-available, rational arithmetic C/C++ library MIRACL [53]. In practice, due to the inherent limitations of human cognition, user passwords are often memorable short strings and hence the password space is very restricted, e.g., $|\mathcal{D}_{pw}| \leq 10^6$ [54], [55], and it follows that $\mathcal{A}$ can complete the above attacking procedure in seconds on a common PC.

Our attack shows that once the smart-card factor is compromised, the corresponding password factor can be offline guessed and hence the entire system collapses. This indicates that Tsai et al.'s scheme is not a truly two-factor scheme, while the original scheme (i.e., Li et al.'s scheme [26]) does not suffers from this pitfall. The failure of Tsai et al.'s scheme is mainly attributed to the pre-computation phase, which facilitates $\mathcal{A}$ to obtain not only the long-term parameters but also the session-specific values (e.g., the parameter c). To eliminate this pitfall, an intuitive solution is to remove the pre-computation phase and compute e and c in the login phase, however this is at the cost of efficiency. We also note that, Li et al.'s scheme [26] employs a similar pre-computation technique and is free from the above attack, but it is at the price of de-synchronization attack (a kind of denial of service attack, see Section 4.2.2). Naturally, one may ask, whether there exists a secure anonymous scheme that employs the pre-computation technique? This question is very interesting yet out of the scope of this work, and we prompt it as an open problem.

3.2.2 Smart card loss attack II

To support local user password update (i.e., DA2) like that of the schemes in [13], [19], [29], [33], the password change phase of Tsai et al.'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 as discussed in Section 3.2.1), she 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 $V^{new} = V \oplus h(R\|b) \oplus h(PW_i^{new}\|b)$ and replaces V with V^{new}.

Once the value of V is updated, legitimate user U_i cannot login successfully even after getting her smart card back because $V^{new} \oplus h(PW_i\|b) \neq h(ID_i\|x)$, and thus U_i's subsequent login request will be denied by the server S during all the following login phases. This means denial of service attack can be launched easily once the card is lost, and thus this scheme fails to fulfill SR6.

Actually, this design flaw may also give rise to *another* quite practical and troublesome problem: if a legitimate user accidently keys an incorrect value for the current (old) password in the password change process, the parameter V will be updated to an unpredictable (random) value. From now on, the smart card will become completely unusable unless the user re-registers with the server.

It is worth noting that, although these two vulnerabilities seem too basic to merit discussion, they are really practical and cannot be well conquered just with minor revisions. To eliminate these two vulnerabilities (i.e., achieving SR6) while preserving DA2, a verification of the authenticity of the original password before updating the value of V in the memory of smart card is essential. And thus, besides V, some additional parameter(s) should be stored in the smart card. Note that this may introduce new vulnerabilities, such as offline guessing attack and user impersonation attack. This subtlety has also been observed by Nam et al. [57] and Xiang et al. [58], but unfortunately, they left it as an open problem. Most subsequent works either just overlook this issue [13], [18], [19], [27], [29] or choose not to provide local user password change [26], [41], [52], while the few rest [33]–[35], [42] that are ambitious to both support local password change (i.e, DA2) and resist against the above

[2] In [40], 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 they have implicitly made the assumption that $\mathcal{A}$ can return the breached card without detection.

TABLE 5
Computation evaluation of related operations on common PCs

Experimental Platform (common PCs)	ECC Point Multiplication T_P(ECC sect163r1 [56])	Modular Exponentiation T_E($\|n\| = 512$)	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	2.573 ms	2.049 µs	2.580 µs	0.011 µs
Intel E5500 2.80 GHz	0.617 ms	1.348 ms	0.572 µs	0.753 µs	0.009 µs
Intel i3-530 2.93 GHz	0.508 ms	1.169 ms	0.541 µs	0.693 µs	0.008 µs

smart card loss attack II (i.e, SR6) are all found prone to offline password guessing attack [37], [59].

To gain more insights into this problem, here we give a concrete example. Suppose an additional parameter $A_i = h(ID_i \parallel h(PW_i))$ is stored in the smart card. Whenever U_i wants to update her password, first she must input her identity ID_i^* and password PW_i^*, then the smart card verifies whether $h(ID_i^* \parallel h(PW_i^*))$ is equal to the stored A_i. It is not difficult to see that $\mathcal{A}$ could exhaustively enumerate all the (ID_i, PW_i) pairs and determine the correct one in an offline manner once the parameter A_i has been 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 exist $\frac{|\mathcal{D}_{id}| * |\mathcal{D}_{pw}|}{2^8} \approx 2^{32}$ candidates of (ID, PW) pair to thwart $\mathcal{A}$ when $|\mathcal{D}_{id}| = |\mathcal{D}_{pw}| = 10^6$ [54], [55], where $\hbar(\cdot)$ is a special one-way hash function $\{0,1\}^* \rightarrow \{0,1,\ldots,255\}$, $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the size of the identity space and password space, respectively. Even if ID_i is also leaked to $\mathcal{A}$, there still exist $\frac{|\mathcal{D}_{pw}|}{2^8} \approx 2^{12}$ candidates of PW_i, each of which can only be excluded by an online guessing attack, while online guessing can be effectively thwarted [60]. In this way, $\mathcal{A}$ is prevented from obtaining the exactly correct PW_i and we call A_i computed through this new approach "a fuzzy verifier", which achieves the same effect with that of [43]. One may wonder what if U_i happens to input 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 will be reduced to $\frac{1}{256^2}$ if we further require the user types her old/new passwords twice whenever changing password and if $\hbar(\cdot)$ behaves like a random oracle. An obvious "by-product" of this "fuzzy verifier" is that it can be used to provide timely wrong password detection when login (i.e., DA10).

Therefore, only radical changes in Tsai et al.'s scheme can completely eliminate the above pitfalls, 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 5.3. 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., Evernote [9] and LinkedIn [10]) have provided wonderful materials for this use, and it constitutes one of our future work.

3.2.3 Card revocation problem

In Tsai et al.'s scheme [36], to support the criterion DA10 (i.e., no password-related verification table stored on the server), there is no user-related (or card-related) data kept on the server at all. Of course, DA10 can be provided. But going too far is absolutely undesirable. As no card-related information stored in the server, there is no way for the server to tell apart a valid card from an invalid card, which means invalid (or expired) cards cannot be revoked.

What's more, Tsai et al.'s scheme is not easily repairable when the user re-registers with the server after the user finds that her smart card has been lost and/or the critical parameter $h(ID_i \| x) = V \oplus h(PW_i \| b)$ is somehow obtained by an adversary. As described in Section 3.2.1, there are non-negligeable possibilities that $h(ID_i \| x)$ may be leaked to $\mathcal{A}$ once the user's card is lost. In this case, impersonation attacks cannot be prevented even if U_i finds that her card has been out of control and then re-registers with S. As the value of $h(ID_i \| x)$ is computed only with the contribution of U_i's identifier ID_i and S's permanent secret key x, S cannot update $h(ID_i \| x)$ for U_i unless ID_i or x can be modified to a new one. However, since x is generally related to all registered users rather than U_i only, it is inefficient and virtually unrealistic if x is changed to restore the security of U_i only. On the other hand, it is impractical to change ID_i, which is unalterable in most scenarios and may be bound to U_i in many application systems.

3.2.4 Flaws in Tsai et al.'s formal security proof

Generally speaking, a two-factor protocol achieving semantic security or the so-called "AKE security" [43], [61] under the non-tamper resistance assumption about the smart card (i.e., C-2i in Table 1 or the smart-card-lose case in [36]) can provide a basic level of security, such as resistance to impersonation attack and offline password guessing attack, even if the card has been lost and breached. Tsai et al.'s protocol is armed with a claimed proof of semantic security, yet as shown earlier, it cannot withstand offline password guessing attack once the smart-card factor is compromised.

Now a paradox arises: *How can a protocol that was proved secure later turn out to be insecure?* To address this interesting question, we scrutinize the reductionist security arguments of Tsai et al.'s protocol and manage to uncover the fundamental flaw in their reasoning of the proof.

The security model adopted by Tsai et al. [36] is based on the work of Xu et al. [52], while the latter is essentially a variant of the random oracle model introduced by Bellare et al. [61]. The general rationale that lies behind a proof of semantic security in the random oracle model is: (1) Modelling any hash function as an oracle which outputs a random value for each new query and the same value for every identical query; (2) Supposing $\mathcal{A}$ can break the semantic security of the target protocol $\mathcal{P}$; (3) Exploiting $\mathcal{A}$ to build algorithms for each of the underlying cryptographic primitives (e.g., computational Diffie-Hellman assumption and decisional Diffie-Hellman assumption) in such a way that if $\mathcal{A}$ manages to break protocol $\mathcal{P}$, then *at least* one of these algorithms succeeds in solving an underlying primitive. Since these primitives are widely deemed intractable, no probabilistic polynomial time (PPT) algorithm can succeed in breaking any one of them and hence protocol $\mathcal{P}$ remains secure (i.e., semantic security is preserved).

The tactic adopted by Tsai et al. to prove semantic security of the session key is similar to that of [43], [52]. They construct a series of attacking games G_n ($n = 0, 1, \cdots, 5$), starting with the real attack G_0 and ending in a game G_5 where $\mathcal{A}$'s advantage is 0, and for which they bound the difference in $\mathcal{A}$'s advantage between any two consecutive games. This yields a bound on $\mathcal{A}$'s advantage in attacking the original protocol $\mathcal{P}$. Though Tsai et al. incrementally define a series of games, yet, *their security reductions are far from rigorous and they have placed a substitute by subterfuge*! More specifically, they carried out a security proof for their *three-factor* version of the scheme with a security model which is only suitable for *two-factor* authentication. Without an appropriate security model (which is the cornerstone of a security reduction) employed, the proof is destined to fail.

Tsai et al. divides their security model into five cases (see Section III of [36]), yet no case is related to the biometric factor. As is well known, biometrics (e.g., fingerprint and iris) are prone to various sophisticated attacks and shall never be deemed as a secure "black box" that is free from threats [14], [62], [63]. This means their security model are unfit for analyzing three-factor (i.e., smart card, password and biometric) schemes, yet Tsai et al. just make an attempt to accomplish this task, while leaving the two-factor version of their scheme un-analyzed. The second one of their five cases is the smart-card-loss case. In this case, $\mathcal{A}$ is essentially equipped with the ability C-1&C-2ii (see Table 1), i.e., $\mathcal{A}$ is assumed to be able to control the communication channel and has breached the user's card. Just under this case, we have shown their two-factor version cannot achieve truly two-factor security, which implies, at least, that their three-factor version cannot achieve truly three-factor security — Once the smart card factor and the biometric factor are compromised, the password factor can be offline guessed. This indicates neither the two-factor version nor three-factor version of their scheme are sound.

4 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 adopts a completely difficult strategy: each party updates the user's session-variant pseudonym identity after having authenticated its counterpart. Though this strategy can indeed support user un-traceability, as we shall show in the following, it is highly impractical, for it introduces a serious vulnerability that greatly downgrades the usability of the scheme. Moreover, this scheme also fails to provide truly two-factor security which is the most essential goal a two-factor shall achieve.

4.1 Review of Li's scheme

In 2013, Li [41] proposed two ECC-based two-factor authentication schemes, one with user anonymity and the other without user anonymity. Here we are only interested in the one with user anonymity. This scheme consists of four phases: registration, authentication, password update and user eviction. For clarity, the intuitive abbreviations are listed in Table 4 and some additional ones in Table 6.

TABLE 6
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
U_S	public key of remote server S, where $U_S = d_S \cdot G$
K_x	secret key where $K = PW_A \cdot r_A \cdot U_s = (K_x, K_y)$
$E_{K_x}(\cdot)/D_{K_x}(\cdot)$	symmetric encryption/decryption with K_x

4.1.1 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, server S selects a pseudo-identity IND_A for A, and creates an entry (IND_A, U_A, $status\text{-}bit$) in its database, where $status\text{-}bit$ indicates the status of A. More specifically, when A has logged-in to S, the $status\text{-}bit$ is set to 1, otherwise it is set to 0.

4) $S \Rightarrow A$: 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 r_A into the card, which means the card henceforth stores $\{IND_A, G, h(\cdot), E_{K_x}(\cdot)/D_{K_x}(\cdot), r_A\}$.

4.1.2 Authentication phase

When A logins to S, the following steps are involved:

1) A inserts her smart card into the card reader, and inputs her password 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. A checks whether the hashed result of $\{W_S \| U'_A \| sk \| IND'_A\}$ equals the received

$H(W_S \| U'_A \| sk \| IND'_A)$. If they are equal, A is assured that S is authentic 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 $\{IND'_A, U'_A\}$.

4.1.3 Password change phase

The password change phase is provided to allow users to change their passwords freely (but not locally). When A wants to change her password, she first needs to go through the above authentication phase to make sure that the input password is valid and then can change it to a new one.

4.1.4 User eviction phase

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

4.2 Cryptanalysis of Li's scheme

In [41], Li demonstrated that Islam-Biswas's scheme [64] 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 presented. Besides its high efficiency due 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 — it is of poor usability and fails to achieve two-factor security under their non-tamper resistance assumption of the smart cards.

4.2.1 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 [39], [40], to date few schemes have achieved this "precious" goal. Once more, Li's attempt [41] ends in vain, as we will show how an attacker in possession of a user's smart card can recover the 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 and extracted the secret information $\{IND_A, r_A, U_S\}$ by using side-channel attacks [23]–[25] herself (or with recourse to professional labs), and then $\mathcal{A}$ returns the breached card back to A without A's awareness. Note that these assumptions are quite practical as discussed in Section 2.1 and Section 3.2.1. 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 $\mathcal{D}_{pw}$.

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^* 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 $|\mathcal{D}_{pw}|$ denote the size of password space $\mathcal{D}_{pw}$. The time complexity of the above attacking procedure is $\mathcal{O}(|\mathcal{D}_{pw}| * (2T_P + T_S))$, 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 $|\mathcal{D}_{pw}|$, and thus our attack is quite effective. Furthermore, in practice users generally choose common and relatively weak passwords, and thus $\mathcal{D}_{pw}$ is very restricted , e.g., $|\mathcal{D}_{pw}| \leq 10^6$ [54], [55]. According to the operation timings reported in Table 5, $\mathcal{A}$ can accomplish the above procedure in seconds on a common PC.

Our attack implies 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 [39]. Since then, there is no way to prevent $\mathcal{A}$ from impersonating A to enjoy the system's services/resources, unless A re-registers with the server. This suggests that Li's scheme is essentially not a truly two-factor scheme and provides no better security than the original scheme (i.e., Islam-Bswas's scheme [64]).

4.2.2 De-synchronization attack

To provide user un-traceability, a number of dynamic-ID based two-factor protocols (e.g., [36], [43], [47], [65]) 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., [26], [27], [66]) adopt a quite 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 somewhere on the user side, and to recognize this user in the following protocol run, the sever also needs to maintain a copy of the user's new pseudo-identity after the current protocol run. So the synchronization of this new 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 Section 4.1.2) 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 her card memory. Before $\{IND_A, h(sk \| IND'_A)\}$ reaches S, $\mathcal{A}$ 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 $\{U'_A, IND'_A\}$ in its backend database. Consequently, the consistency of the user pseudo-identity between A and S is broken. Thereafter, A will send IND'_A to S in her login requests, yet S stores the old IND_A and will always reject A due to $IND'_A \neq IND_A$.

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. In other words, there are no other expensive operations involved such as reverse engineering and power analysis. Moreover, it is not difficult to see that, instead of altering this protocol transcript, $\mathcal{A}$ might equally attain her end by simply blocking this protocol transcript.

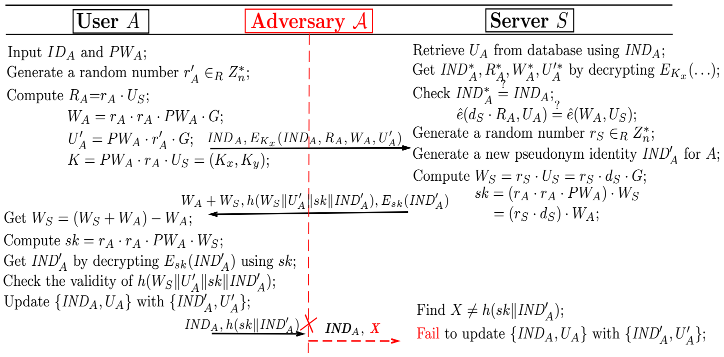
User A
Adversary $\mathcal{A}$
Server S
Input ID_A and PW_A;
Generate a random number $r'_A \in_R Z_n^*$;
Compute $R_A = r_A \cdot U_S$;
$W_A = r_A \cdot r_A \cdot PW_A \cdot G$;
$U'_A = PW_A \cdot r'_A \cdot G$;
$K = PW_A \cdot r_A \cdot U_S = (K_x, K_y)$;
$IND_A, E_{K_x}(IND_A, R_A, W_A, U'_A)$
Retrieve U_A from database using IND_A;
Get $IND^*_A, R^*_A, W^*_A, U'^*_A$ by decrypting $E_{K_x}(\ldots)$;
Check $IND^*_A \overset{?}{=} IND_A$;
$\hat{e}(d_S \cdot R_A, U_A) \overset{?}{=} \hat{e}(W_A, U_S)$;
Generate a random number $r_S \in_R Z_n^*$;
Generate a new pseudonym identity IND'_A for A;
Compute $W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G$;
$sk = (r_A \cdot r_A \cdot PW_A) \cdot W_S$
$= (r_S \cdot d_S) \cdot W_A$;
$W_A + W_S, h(W_S \| U'_A \| sk \| IND'_A), E_{sk}(IND'_A)$
Get $W_S = (W_S + W_A) - W_A$;
Compute $sk = r_A \cdot r_A \cdot PW_A \cdot W_S$;
Get IND'_A by decrypting $E_{sk}(IND'_A)$ using sk;
Check the validity of $h(W_S \| U'_A \| sk \| IND'_A)$;
Update $\{IND_A, U_A\}$ with $\{IND'_A, U'_A\}$;
$IND_A, h(sk \| IND'_A)$
IND_A, X
Find $X \neq h(sk \| IND'_A)$;
Fail to update $\{IND_A, U_A\}$ with $\{IND'_A, U'_A\}$;

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

It also should be noted that, although this vulnerability appears rather simple, it cannot be well addressed just with minor revisions. One may think that, if an additional "ack" message is sent back to user A and only when A has received this "ack" should 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 wait for an 'ack" message which never comes, failing to replace $\{r_A, IND_A\}$ with $\{r'_A, IND'_A\}$. On the other hand, the server S has already updated $\{U_A, IND_A\}$ to $\{U'_A, IND'_A\}$ before sending this "ack". Similarly, any attempt to overcome this vulnerability by adding new protocol flow(s) will be doomed to failure.

Another possible (defensive) solution one may think is to store both the old and new pseudo-identities on the smart card and/or the server side, and the old pseudo-identity is put to use whenever the new one fails to work (i.e., a de-synchronization has occurred). Regrettably, such a solution may bring other issues. One prominent problem is that user un-traceability will be violated once the adversary blindly blocks a new conservation initiated by the same user.

Remark 2. We have analyzed more than one hundred and twenty recently proposed two-factor schemes and more than fifty anonymous two-factor schemes (some of our recent cryptanalysis results include [39], [43], [47], [48], [59], [67]), and observed that all these schemes that employ a similar synchronization mechanism to that of Li's scheme [41] to maintain the consistence of user pseudo-identities are subject to de-synchronization problem without no exception. Some other problematic ones include [27], [66], [68]. The above attack reveals the unsoundness of these schemes. Unfortunately, as stated in [69], [70], the widely used formal methods (e.g., random oracle model, BAN logic) can not capture such structural mistakes, and assuring soundness of authentication protocols still remains an open issue.

5 Exploration of the Relationships among Evaluation Criteria

In this section, we first summarize and investigate the available ways to achieve user anonymity. Then, we explicate the definitions of some criteria in Madhusudhan-Mittal's set [30] to provide a clear basis for investigating their inner relationships. Further, using these two schemes examined earlier in this paper as case studies and building on the previous cryptanalysis results [35], [37], [39], [40], [46], [59], [67], we for the first time show that it is unlikely to construct an "ideal" dynamic ID-based two-factor authentication scheme that satisfies all the criteria in Madhusudhan-Mittal's set [30], and provide a negative answer to the question raised by Huang et al. [40]. Lastly, considering the promising applications of formal methods, we discuss what the role of "provable security" is in breaking the vicious circle of "attack-fix-attack-fix".

5.1 Ways to achieve user anonymity

Intuitively, there are two broad approaches to implement the "dynamic ID technique": (1) Making use of cryptographic primitives (e.g., symmetric-key operations like that of [18], [19], CCA-2 public key encryption [71] like that of [36], [43], ring/group signatures [72], [73] or attribute-based signatures [74]); and (2) Exploiting some non-cryptographic mechanisms (e.g., pre-loading a pseudo-IDs pool like that of [75], or synchronization mechanism like that of [26], [41]). However, it is not difficult to see that, the idea of pre-loading a large pseudo-IDs pool on a resource-limited smart card is practically infeasible, while most anonymous two-factor schemes that employ some types of synchronization mechanism, which has been discussed in Section 4, are prone to a fatal usability problem.

As non-cryptographic mechanisms do not seem to work, the cryptographic approach remains the only alternative to preserve user anonymity in two-factor authentication. Since ring/group signatures often need the support of public key infrastructure (PKI) and their computation overhead grows linearly with the ring/group size, neither of them could readily be used in two-factor authentication schemes. Attribute-based signatures are free from the burden of PKI, yet there are generally a number of expensive bilinear pairing operations involved, which may be unsuitable for implementation on low-power smart cards. What's more, as pointed out in [15], how to keep secret the signing key on user side is a non-trivial issue when smart cards are assumed that they can be tampered when lost.

To the best of knowledge, all this for the first time well explains why most anonymous two-factor schemes [17]–[20], [33], [35], [36], [41], [43] prefer to only employ some symmetric cryptographic primitives (e.g., hash functions, XOR operations, symmetric encryption) or a few comparatively lightweight public-key operations (e.g., modular exponentiations and elliptic curve point multiplications). Accordingly, we further classify these anonymous schemes into two categories: (1) Symmetric-key-based ones; and (2) Public-key based ones. We say the former kind of schemes are with the property "DA5-SymmetricKey" and the latter kind of schemes provide the property "DA5-PublicKey". In a recent work [59], we manage to show that these two properties do have intrinsic (but subtle) relationships with the security requirement SR6 (i.e., Resistance to smart card loss attack), which will be elaborated later.

5.2 Further explications of some criteria

A scheme supporting property 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 Yang and Shieh [6] in 1999, DA1 has become one of the most basic design goals of two-factor schemes [31], [32]. Like the scheme investigated in Section 3, a number of schemes (e.g., [17], [18], [20], [68]) attempting to achieve DA1 advocate that the server shall only keep some secret key(s) for verifying the users and there be no other user-specific data stored on the server. On the contrary, the other schemes (e.g., the scheme examined in Section 4 and [19], [27], [43]) with DA1 store some non-security-critical user-specific information such as $\{ID_i, T_{reg}\}$ on the server side, where ID_i is user's identity and T_{reg} user's registration time. We say the former kind of schemes provides the property "DA1-Strong" and the latter kind of schemes supports the property "DA1-Weak".

In password-based authentication, besides free user password choice (DA2), it is a universally accepted practice that users shall regularly change their passwords [76]. Accordingly, as stated in Section 2.1, the password change phase has been a basic phase in any two-factor scheme. Obviously, this phase may either enable a user to locally change her password or require a user to interact with the server in order to change her password. As investigated in Section 3.2.2, a scheme that facilitates the 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 the user to locally change her password as well as supporting secure password change is prone to the same threat. For ease of presentation, we say the former scheme is with attribute "DA2-Local-Secure", the latter one with attribute "DA2-Local-Insecure". In addition, for a scheme that does not support local password change, we say this scheme provides attribute "DA2-Interactive".

5.3 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 $\rhd$, respectively. More specifically, DAi and SRj being of a symbiotic relation (i.e., DAi∞SRj) means if either one is held by the scheme, both will be held; DAi and SRj are mutually exclusive (i.e., DAi$\otimes$SRj) if and only if at any time, at most one of them is held by the scheme; DAi and SRj being of an implicative relation means that either (1) whenever DAi is held by the scheme, SRj is held by the scheme (i.e., DAi$\rhd$SRj) or (2) whenever SRj is held by the scheme, DAi is held by the scheme (i.e., SRj$\rhd$DAi). Our observations are summarized in Table 7 and detailed as follows:

1) DA1-Strong$\otimes$DA9. A scheme that 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., Tsai et al.'s scheme [36] and the ones in [18], [68]) tell apart a valid card from an expired card? As far as we know, there is no way for the server to accomplish this aim. One may think that, for those expired cards, the server S adds a piece of valid time information as well as a digital signature for this data on the card, then by verifying the signature, S can tell apart valid cards from expired ones; For those revoked cards, S can maintain a revocation list, as it does in a traditional certificate authority. However, one can see that, in this way, it will defeat the purpose of storing no user-specific data on S (i.e., DA1-Strong).

2) DA1-Strong$\rhd$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$\rhd$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 that 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 [34], or $\{r, N = h(r\|x) \times h(PW_i), Y = h(ID_i\|h(r\|x))\}$ in [42]) 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$\otimes$SR6. A scheme that 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 [34], or $\{r, N = h(r\|x) \times h(PW_i), Y = h(ID_i\|h(r\|x))\}$ in [42]) 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$\otimes$DA10. This can be obtained directly from the relationship that DA2-Local-Secure∞DA10.

7) DA2-Local-Insecure$\otimes$SR6. A two-factor scheme that 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 3.2.2. On the other hand, a scheme supports SR6 means an attacker shall not easily change the password when obtaining the card, indicating DA2-local-insecure is not supported in the scheme.

8) DA2-Interactive$\otimes$DA10. A scheme attains the feature DA2-Interactive (e.g., the schemes in [26], [41]) means that the user is required to change her password by interacting

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-SymmetricKey	∗	∗	∗	∗	∗	∗	⊗	⊗	∗	∗	∗	▷	▷	∗	▷	⊗	∗	▷	∗
DA5-PublicKey	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	▷	▷	∗	▷	∗	∗	▷	∗
DA6	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗
DA7	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗
DA8	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗
DA9	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗
DA10	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	∗	⊗	∗	∗	∗

Note: 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.

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. This indicates that a scheme supporting DA2-Interactive cannot provide DA10, and vice versa.

9) DA5▷SR2, SR3, SR5 and SR8. A scheme that supports DA5 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., DA5) can not be assured. On the other hand, neither the achievement of SR2 (or, SR3, SR5 SR8) indicates the achievement of DA5. For example, a scheme (e.g., [65]) achieves SR8 may still be prone to replay attack, which invalidates DA5.

10) DA5-SymmetricKey⊗DA7, DA8 and SR6. A scheme that supports DA5-SymmetricKey means mutual authentication is achieved by only using symmetric-key techniques. According to [59], under the assumption C-2 as listed in Table 1, schemes that do not employ public-key primitives are intrinsically unable to provide DA7, DA8 or SR6.

11) DA10⊗SR6. A scheme that supports DA10 (e.g., [34], [68]) 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(s) stored on the card. In this case, an attacker can perform smart card lost attack just by exploiting this data.

12) SR6▷DA4. According to the definitions given in [30], 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 3.2.2), 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 [18] is a typical one that fails to achieve both DA4 and SR6, while the schemes in [19], [20], [33] achieve DA4 but fail 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 changes her password (i.e., locally or interactively), either SR6 or DA10 definitely can not be achieved, which indicates it is unlikely to construct an "ideal" scheme that satisfies all the criteria in Madhusudhan-Mittal's evaluation set [30]. In particular, the relationships among DA2-Local-Secure, DA2-Local-Insecure and SR6 suggest a negative answer to the question left by Huang et al. [40] — "whether or not there exist secure smart-card-based password authentication protocols that the password-changing phase does not need any interaction with the server"?

5.4 The role of "provable security"

For several decades, protocol designers worked by trials and errors. A protocol was proposed, and then the community tried to break and fix it. It turned out that many protocols were compromised a number of years after they were first suggested. This unsatisfactory situation was not ameliorated until the early 1980s by the seminal work of Goldwasser and Micali [77], followed by a series of influential works [44], [61], [78]. These works suggest that security could be "proved" under well known complexity-theoretic assumptions (e.g., the intractability of CDH problem), and the methodology they identified is thereafter called "provable security". This methodology has been proved especially useful in eliminating redundancies in the list of attacks on a protocol. For example, most of the attacking-oriented goals listed in Table 2 may be reduced to only two formal ones, namely, semantic security and mutual authentication [43]. Furthermore, security goals defined in a formal model are more precise in capturing the security provisions offered by a protocol than that of a heuristic model. Consequently, provable security has become an indispensable tool in analyzing and evaluating new cryptographic schemes.

However, provable security has its limitations. Generally, the process of providing "provable security" for a protocol entails five stages: (1) Definition of adversarial model; (2) Statement of security goals; (3) Specification of cryptographic assumptions; (4) Description of protocol; and (5) Reductionist proof. It follows that any provably secure protocol meets its goals within some security model under some cryptographic assumption(s), rather than the mere claim that such-and-such a protocol achieves provable security. Past research over the last thirty decades has told us that, a security proof is highly prone to be fallacious due to the adoption of an insufficient security model which fails to capture all the realistic capabilities of the adversary or due to a flawed/non-tight security reduction, and "the field of provable security is as much an art as a science" [79], [80]. Our past experience of cryptanalysis of two-factor schemes over the last two decades has revealed that, most of the two-factor schemes (e..g., the ones in [27], [29], [42], [52]) that are equipped with a formal proof have been found severely problematic shortly after they were presented. Our "smart card loss attack I" (see Section 3.2.1) on Tsai et al.'s protocol perfectly demonstrates that, having a formal (but insufficient) security model and designing a "proven secure" protocol in that model are no panacea for assuring actual security. While formal methods are often misused and reductionist security proofs are usually very intricate, turgid and prone to errors, particular care shall be given when conducting a proof for a two-factor protocol.

It is also worth noting that, many attacking scenarios are difficult to be captured in a formal adversarial model. For example, the "smart card loss attack II" (see Section 3.2.2) on Tsai et al.'s protocol and the "de-synchronization attack" (see Section 4.2.2) on Li's protocol, as far as we know, can not be captured in any existing model. While these practical attacks cannot be modelled in current models, it is crucial that protocol designers are fully aware of such damaging threats. What's more, as shown in [70], [81], even if the security model employed is the right one at the present, the correctness of a security proof largely depends on the prover's attacking experience. Last but not the least, even if the security model is accurate and the security proof is correct, the features (functionalities) of a protocol can hardly be analyzed or assured by the methodology of provable security. All this highlights the critical role that old-fashioned cryptanalysis continues to play in establishing confidence in the security and versatility of a protocol, suggesting the importance and necessity of this work.

6 Conclusion

In this paper, we have investigated the question of *whether is it possible* to build an "ideal" anonymous two-factor authentication scheme that satisfies all the criteria listed in Madhusudhan-Mittal's evaluation set? By cryptanalyzing two foremost anonymous two-factor schemes as case studies, we uncover several subtleties and challenges in designing this type of schemes, and explore the relationships among the criteria. Our results highly indicate a negative answer to the examined question. Most essentially, we find that, a scheme supporting local user password change is unlikely to achieve "SR6: resistance to smart card loss attack", while a scheme not supporting local user password change is unlikely to provide the property of "DA10: timely typo detection". This presents an unavoidable usability-security tradeoff, thereby also suggesting a negative answer to the open question raised by Huang et al. [40].

We believe this work provides a better understanding of the underlying evaluation metric for anonymous two-factor schemes, which is of fundamental importance for security engineers to make their choices correctly and for protocol designers to develop 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 recently disclosed large-scale real-life password data-sets like the 50 million "Evernote" dataset and the 6.4 million "LinkedIn" dataset.

Acknowledgment

The authors are grateful to the anonymous reviewers for their valuable comments that highly improve the readability and completeness of the paper. This research was supported by the National Natural Science Foundation of China (NSFC) under Grants No.61170263 and No.6137024,

References

[1] *About EMV (Europay, MasterCard, and Visa)*, EMVCo Ltd., Sep. 2013, available at http://www.emvco.com/approvals.aspx?id=91.

[2] L.-Y. Yeh and W. Tsaur, "A secure and efficient authentication scheme for access control in mobile pay-tv systems," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1690–1693, 2012.

[3] M. Bond, O. Choudary, and S. Murdoch, "Chip and skim: cloning EMV cards with the pre-play attack," in *Proc. IEEE S&P 2014*. IEEE Computer Society, 2014, pp. 1–15.

[4] Y. Deswarte and S. Gambs, "The challenges raised by the privacy-preserving identity card," in *Cryptography and Security: From Theory to Applications: Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, ser. LNCS, D. Naccache, Ed. Springer/Berlin Heidelberg, 2012, vol. 6805, pp. 383–404.

[5] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Comput. Secur.*, vol. 30, no. 4, pp. 208–220, 2011.

[6] W.-H. Yang and S.-P. Shieh, "Password authentication schemes with smart cards," *Comput. Secur.*, vol. 18, no. 8, pp. 727–733, 1999.

[7] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE S&P 1992*. IEEE, 1992, pp. 72–84.

[8] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *J. ACM*, vol. 57, no. 1, pp. 1–41, 2009.

[9] "50 million compromised in evernote hack," Mar. 2013, http://www.cnn.com/2013/03/04/tech/web/evernote-hacked/.

[10] P. Sean, *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.

[11] "Heartbleed – openssl zero-day bug leaves millions of websites vulnerable," April 2014, http://www.tuicool.com/articles/yyUfUz.

[12] C. Fan, Y. Chan, and Z. Zhang, "Robust remote authentication scheme with smart cards," *Comput. Secur.*, vol. 24, no. 8, pp. 619–628, 2005.

[13] G. Yang, D. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, 2008.

[14] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1390–1397, 2011.

[15] Y. G. Wang, "Password protected smart card and memory stick authentication against off-line dictionary attacks," in *Proc. SEC 2012*, D. Gritzalis, S. Furnell, and M. Theoharidou, Eds. Springer Boston, 2012, vol. 376, pp. 489–500.

[16] F. Zhu, S. Carpenter, and A. Kulkarni, "Understanding identity exposure in pervasive computing environments," *Pervasive Mob. Comput.*, vol. 8, no. 5, pp. 777–794, 2012.

[17] M. Das, A. Saxena, and V. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 629–631, 2004.

[18] Y. Wang, J. Liu, F. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Comput. Commun.*, vol. 32, no. 4, pp. 583–585, 2009.

[19] M. Khan and S. Kim, "Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme'," *Comput. Commun.*, vol. 34, no. 3, pp. 305–309, 2011.

[20] Y.-F. Chang, W.-L. Tai, and H.-C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *Int. J. Commun. Syst.*, 2013, doi: http://dx.doi.org/10.1002/dac.2368.

[21] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, 2002.

[22] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, "On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx Virtex-II FPGAs," in *Proc. ACM CCS 2011*. New York, NY, USA: ACM, 2011, pp. 111–124.

[23] K. Nohl, D. Evans, S. Starbug, and H. Plötz, "Reverse-engineering a cryptographic rfid tag," in *Proc. USENIX Security 2008*. USENIX Association, 2008, pp. 185–193.

[24] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using am demodulation on commercial smart cards with seed," *J. Syst. Soft.*, vol. 85, no. 12, pp. 2899 – 2908, 2012.

[25] A. Barenghi, L. Breveglieri, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proc. of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[26] X. Li, W. Qiu, D. Zheng, K. F. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 793–800, 2010.

[27] R. C. Wang, W. S. Juang, and C. L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key," *Comput. Commun.*, vol. 34, no. 3, pp. 274–280, 2011.

[28] F. Wen and X. Li, "An improved dynamic ID-based remote user authentication with key agreement scheme," *Comput. Electr. Eng.*, vol. 38, no. 2, pp. 381–387, 2012.

[29] S. H. Wu, Y. F. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Secur. Commun.Netw.*, vol. 5, no. 2, pp. 236–248, 2012.

[30] R. Madhusudhan and R. Mittal, "Dynamic ID-based remote user password authentication schemes using smart cards: A review," *J. Netw. Comput. Appl.*, vol. 35, no. 4, pp. 1235–1248, 2012.

[31] I. Liao, C. Lee, and M. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, 2006.

[32] C. Tsai, C. Lee, and M. Hwang, "Password authentication schemes: current status and key issues," *Int. J. Netw. Secur.*, vol. 3, no. 2, pp. 101–115, 2006.

[33] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 195–206, 2014.

[34] X. Li, J. Niu, M. Khurram Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1365–1371, 2013.

[35] S. Kumari and M. K. Khan, "More secure smart card-based remote user password authentication scheme with user anonymity," *Secur. Comm. Netw.*, 2013, doi: http://dx.doi.org/10.1002/sec.916.

[36] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, "Novel anonymous authentication scheme using smart cards," *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2004–2013, 2013.

[37] D. Wang, C. G. Ma, S. Zhao, and C. Zhou, "Cryptanalysis of two dynamic ID-based remote user authentication schemes for multi-server architecture," in *Proc. NSS 2012*, ser. LNCS, L. Xu, E. Bertino, and Y. Mu, Eds. Springer-Verlag, 2012, vol. 7645, pp. 462–475.

[38] S. Kumari and M. K. Khan, "More secure smart card-based remote user password authentication scheme with user anonymity," *Secur. Comm. Netw.*, 2014, doi: http://dx.doi.org/10.1002/sec.916.

[39] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proc. ISC 2013*, ser. LNCS, Y. Desmedt and K. Hamlen, Eds. Springer-Verlag, 2013, pp. 1–16, full version at http://eprint.iacr.org/2014/208.pdf.

[40] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1767–1775, 2014.

[41] C.-T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," *IET Inform. Secur.*, vol. 7, no. 1, pp. 3–10, 2013.

[42] K. H. Yeh, C. Su, N. W. Lo, Y. Li, and Y. X. Hung, "Two robust remote user authentication protocols using smart cards," *J. Syst. Soft.*, vol. 83, no. 12, pp. 2556–2565, 2010.

[43] D. Wang, C. G. Ma, P. Wang, and Z. Chen, "iPass: Robust smart card based password authentication scheme against smart card loss problem," J. Comput. Syst. Sci., Accepted, 2014, full version available at http://eprint.iacr.org/2012/439.pdf.

[44] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. EUROCRYPT 2001*, ser. LNCS, B. Pfitzmann, Ed. Springer, 2001, vol. 2045, pp. 453–474.

[45] J. Bonneau, M. Just, and G. Matthews, "What's in a name?" in *Proc. FC 2010*, ser. LNCS, R. Sion, Ed. Springer Berlin/ Heidelberg, 2010, vol. 6052, pp. 98–113.

[46] M. Scott, "Cryptanalysis of a recent two factor authentication scheme," Cryptology ePrint Archive, Report 2012/527, 2012, http://eprint.iacr.org/2012/527.pdf.

[47] D. Wang, P. Wang, and J. Liu, "Improved privacy-preserving authentication scheme for roaming service in mobile networks," in *Proc. WCNC 2014*, April 2014, pp. 3178–3183.

[48] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," Comput. Netw., 2014, doi: http://dx.doi.org/10.1016/j.comnet.2014.07.010.

[49] K. Mangipudi and R. Katti, "A secure identification and key agreement protocol with user anonymity (SIKA)," *Comput. Secur.*, vol. 25, no. 6, pp. 420–425, 2006.

[50] E. Morse, M. Theofanos, Y. Choong, C. Paul, and A. Zhang, "NIST-IR-7867 usability of piv smartcards for logical access," National Institute of Standards and Technology, McLean, VA, Tech. Rep., 2012, doi:http://dx.doi.org/10.6028/NIST.IR.7867.

[51] C. Paul, E. Morse, A. Zhang, Y.-Y. Choong, and M. Theofanos, "A field study of user behavior and perceptions in smartcard authentication," in *Proc. INTERACT 2011*, ser. LNCS, P. Campos, N. Graham, and M. Winckler, Eds. Springer-Verlag, 2011, vol. 6949, pp. 1–17.

[52] J. Xu, W. Zhu, and D. Feng, "An improved smart card based password authentication scheme with provable security," *Comput. Stand. & Inter.*, vol. 31, no. 4, pp. 723–728, 2009.

[53] *Miracl library*, Shamus Software Ltd., May 2013, http://www.shamus.ie/index.php?page=home.

[54] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password strength: an empirical analysis," in *Proc. INFOCOM 2010*. IEEE, 2010, pp. 1–9.

[55] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE S&P 2012*. IEEE Computer Society, 2012, pp. 538–552.

[56] *Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0*, Certicom Research, Jan. 2010, available at http://www.secg.org/download/aid-784/sec2-v2.pdf.

[57] J. Nam, S. Kim, and D. Won, "Security analysis of a nonce-based user authentication scheme using smart cards," *IEICE Trans. Fund. Electron. Comm. Comput. Sci.*, vol. 90, no. 1, pp. 299–302, 2007.

[58] T. Xiang, K. Wong, and X. Liao, "Cryptanalysis of a password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 74, no. 5, pp. 657–661, 2008.

[59] C. Ma, D. Wang, and S. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, 2012, doi: http://dx.doi.org/10.1002/dac.2468.

[60] M. Alsaleh, M. Mannan, and P. Van Oorschot, "Revisiting defenses against large-scale online password guessing attacks," *IEEE Trans. Depend. Secur. Comput.*, vol. 9, no. 1, pp. 128–141, 2012.

[61] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT 2000*, ser. LNCS, B. Preneel, Ed. Springer/Berlin Heidelberg, 2000, vol. 1807, pp. 139–155.

[62] *Thinking Putty defeats Fingerprint Scanners!*, Dan's Data, Sep. 2013, available at http://www.puttyworld.com/thinputdeffi.html.

[63] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Trans. Depend. Secur. Comput.*, 2013, doi: http://dx.doi.org/10.1109/TDSC.2013.2297110.

[64] S. H. Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Math. Comput. Model.*, vol. 57, no. 6, pp. 2703–2717, 2013.

[65] T. Zhou and J. Xu, "Provable secure authentication protocol with anonymity for roaming service in global mobility networks," *Comput. Netw.*, vol. 55, no. 1, pp. 205–213, 2011.

[66] F. Wen, W. Susilo, and G. Yang, "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.

[67] D. Wang and P. Wang, "On the usability of two-factor authentication," in *Proc. SecureComm 2014.* Springer-Verlag, Sep. 2014, pp. 1–9.

[68] S. Kumari and M. K. Khan, "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.

[69] G.-L. Wang, J.-S. Yu, and Q. Xie, "Security analysis of a single sign-on mechanism for distributed computer networks," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 294–302, 2013.

[70] F. Bao, "Security can only be measured by attacks," 2008, http://wenku.baidu.com/view/ff14a31ea300a6c30c229f7a.html.

[71] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proc. CRYPTO 1999*, ser. LNCS, M. Wiener, Ed. Springer-Verlag, 1999, vol. 1666, pp. 537–554.

[72] D. Chaum and E. Heyst, "Group signatures," in *Proc. EUROCRYPT 1991*, ser. LNCS, D. Davies, Ed. Springer-Verlag, 1991, vol. 547, pp. 257–265.

[73] J. Ren and L. Harn, "An efficient threshold anonymous authentication scheme for privacy-preserving communications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 3, pp. 1018–1025, 2013.

[74] H. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proc. CT-RSA 2011*. Springer, 2011, pp. 376–392.

[75] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 48–53, 2012.

[76] J. Blythe, R. Koppel, and S. W. Smith, "Circumvention of security: Good users do bad things," *IEEE Secur. & Priv.*, vol. 11, no. 5, pp. 80–83, 2013.

[77] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[78] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. CRYPTO 1993*, ser. LNCS, D. Stinson, Ed. Springer Berlin Heidelberg, 1994, vol. 773, pp. 232–249.

[79] N. Koblitz and A. J. Menezes, "Another look at "provable security"," *J. Cryptology*, vol. 20, no. 1, pp. 3–37, 2007.

[80] T.-Y. Li and G.-L. Wang, "Analyzing a family of key protection schemes against modification attacks," *IEEE Trans. Depend. Secur. Comput.*, vol. 8, no. 5, pp. 770–776, 2011.

[81] A. Menezes, "Another look at provable security," in *Proc. EUROCRYPT 2012*, ser. LNCS, D. Pointcheval and T. Johansson, Eds. Springer Berlin / Heidelberg, 2012, vol. 7237, pp. 8–8, availe at http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf.