Information Sciences 321 (2015) 162–178

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity

Ding Wang [a,c,*], Nan Wang [b], Ping Wang [b,c], Sihan Qing [b]

[a] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[b] School of Software and Microelectronics, Peking University, Beijing 100260, China
[c] National Engineering Research Center for Software Engineering, Beijing 100871, China

ARTICLE INFO

Article history:
Received 5 April 2014
Received in revised form 20 March 2015
Accepted 30 March 2015
Available online 3 April 2015

Keywords:
Password authentication
User anonymity
De-synchronization
Random oracle model

ABSTRACT

Due to its simplicity, portability and robustness, two-factor authentication has received much interest in the past two decades. While security-related issues have been well studied, how to preserve user privacy in this type of protocols still remains an open problem. In ICISC 2012, Kim–Kim presented an efficient two-factor authentication scheme that attempts to provide user anonymity and to guard against various known attacks, offering many merits over existing works.

However, in this paper we shall show that user privacy of Kim–Kim's scheme is achieved at the price of severe usability downgrade – a de-synchronization attack on user's pseudonym identities may render the scheme completely unusable unless the user re-registers. Besides this defect, it is also prone to known key attack and privileged insider attack. It is noted that our de-synchronization attack can also be applied to several latest schemes that strive to preserve user anonymity. As our main contribution, an enhanced scheme with provable security is suggested, and what we believe is most interesting is that superior security and privacy can be achieved at nearly *no* additional communication or computation cost. As far as we know, this work is the *first* one that defines a formal model to capture the feature of user un-traceability and that highlights the damaging threat of de-synchronization attack on privacy-preserving two-factor authentication schemes.

1. Introduction

With the rapid proliferation of wireless network technologies and micro-electromechanical systems, it is becoming more and more convenient for subscribers to enjoy desired services/resources from distributed service servers by using mobile devices (e.g., PDAs, PMPs, Smart phones) at any time and anywhere [78,85]. Meanwhile, it is of utmost importance to ensure that a system's services will not be consumed by unauthorized users in a fraudulent manner. Among numerous methods available for validating a remote user, password-based authentication is one of the most prevalent and effective approaches. Since the introduction of the seminal work (known as encrypted key exchange) of Bellovin–Merritt [5], there have been a

[*] Corresponding author at: Room 560, 2#, Chuangchunxinyuan, University, No. 5 Yiheyuan Road, Haidian District, Beijing 100871, China. Tel.: +86 185 1134 5776; fax: +86 010 6276 5808.

E-mail addresses: wangdingg@mail.nankai.edu.cn (D. Wang), wangnan@ss.pku.edu.cn (N. Wang), pwang@pku.edu.cn (P. Wang), qsihan@ss.pku.edu.cn (S. Qing).

http://dx.doi.org/10.1016/j.ins.2015.03.070
0020-0255/© 2015 Elsevier Inc. All rights reserved.

number of remarkable proposals (e.g., [1,12,48]) with various levels of security and complexity, catering for diverse application scenarios.

In password-based authentication schemes, the server has to *store a sensitive verifier table* that contains the passwords (or the salted passwords, e.g., by using Hash or symmetric encryption) of all the registered users. One prominent issue is that once this sensitive password-verifier table is leaked (hacked), all the users' passwords will be endangered. These days it is no surprise to see news about a catastrophic leakage of thousands of millions of passwords in the headlines [19,24], and the prevalence of zero-day attacks [6] will further aggravate the situation. Recently, two-server password-only schemes (e.g., [47,109]) are suggested to solve the server compromise problem, yet they are helpless to deal with another emerging problem – password leakage at the user side (e.g., malwares and social engineering [34,62]). As deeply investigated in [105], the latter security threat can only be well addressed (i.e., achieving both reasonable security and acceptable usability) by incorporating certain trusted devices.

Owing to its portability, cryptographic capacity and tamper resistance nature, smart cards are usually introduced to serve as the "second line of defense". In this new type of schemes (see Fig. 1), only the user with the valid smart card and the correct password can pass the verification of the remote server, while a compromise of either factor (but not both factors) would pose no danger to the system, which ensures the so-called '*two-factor security*' [40,87]. This sort of schemes has been widely adopted in various security-critical applications, such as e-commerce [27] and e-health [28].

In 1991, Chang and Wu [14] introduced the first smart-card-based password authentication scheme, yet it was not until 2005 that the first scheme that can achieve "truly two-factor security" was given in a seminal work by Fan et al. [30]. Despite the provision of two-factor security, Fan et al.'s scheme [30] fails to support some necessary properties like session key agreement and password update. To eliminate these weaknesses, a number of schemes were further developed [23,39,46,49,60,79,93,104]. Unfortunately, most of them have been demonstrated either insecure against some basic attacks or lack of some important properties (e.g., users cannot freely choose their own passwords). The past thirty years of research on password-based authentication scheme (i.e., single-factor) has proven to be not an easy task [48,74], the design of a practical smart-card-based password authentication scheme (i.e., two-factor) can only be harder, for the designers are faced with the challenging task of reconciling stringent usability, efficiency and security requirements [64,69,88]. To gain a better grasp of the difficulties and challenges in designing a secure and efficient two-factor scheme, readers are referred to the "break-fix-break-fix" history of this area in Fig. 1 of [88].

What further complicates matters is that, smart cards can no longer be deemed as fully tamper-proof devices. Recent research has reported that, the secret parameters stored in common commercial smart cards can be extracted by side-channel attacks such as power analysis [66,67], reverse engineering [3,70] and fault injection attacks (e.g., launched on software-supported Java Cards) [10,54]. In other words, the previous practice of resting complete trust in the physical security of smart cards (e.g., the schemes in [17,23,49,84,93]) is highly risky in the presence of state-of-the-art side-channel attacks. In addition, there is a constant arms race going on between attackers and security practitioners. Even though the physical security of smart cards may be evaluated by independent laboratories or certified by third-party certification authorities (e.g., FIPS-201 [68] and ETSI-TS-102 [29]) at the time of their production, how much confidence can we have that they are still tamper-proof after two years of circulation? Considering this, it is more prudent and desirable to assume that the smart cards can be somehow tampered when they are in the hands of the attacker.[1]

Here we give a concrete example to show that, under the new (but practical) assumption that smart cards can be somehow tampered when lost, two-factor schemes which were traditionally regarded secure may *no longer* be secure anymore. Consider a two-factor scheme in the client–server remote authentication environment, a user-chosen password is used to unlock the smart card which stores the user's private key, and a standard challenge response protocol (e.g., authenticated key exchanged protocol [51]) between the card and server (which keeps the user's corresponding public key) is used to prove user authenticity. This kind of design is quite intuitive and the resulting scheme is indeed secure if the smart cards are assumed to be tamper-proof. However, such an assumption about smart cards, as mentioned earlier, would not always be the case in reality. Once the sensitive data in the card is extracted, the proof-of-concept two-factor scheme will fall: an attacker with the victim's lost card can extract the private key stored in the card memory, and then use this key to impersonate the victim to the server. One may wonder what if the user's private key is not stored in plain-text but encrypted by user password? This case has been investigated in [94] and it falls short of two-factor security, too.

While security-related issues have long been the focus of the community, much less attention has been paid to how to design a privacy-preserving two-factor scheme. These years, with privacy concerns being raised rapidly among individuals and human rights organizations [2,7], user anonymity becomes an admired property of such schemes. Instead of a unique notion of what it means to be "user anonymity", there are a variety of flavors such as sender identity protection, sender un-traceability, sender k-anonymity, blender anonymity, controllable anonymity and so on [38,41,42], and quite varied (sometimes even contradictory) notions may be implemented in different application environments [53,81]. As for remote user authentication, the notion of user anonymity is defined against the public (eavesdropping attackers) rather than the server because the server has to obtain the user's real identity for revoking, accounting and/or billing purposes [65]. Basically, this notion means user identity-protection (i.e., "initiator anonymity" in [41] or "basic user anonymity" in [38]),

[1] As deeply investigated in [88], even if smart cards are assumed to be non-tamper-proof when they are in hands of the attacker for a relatively long period of time (e.g., a few hours), smart-cards-based schemes are still much more robust than common-memory-sticks-based ones in practice.

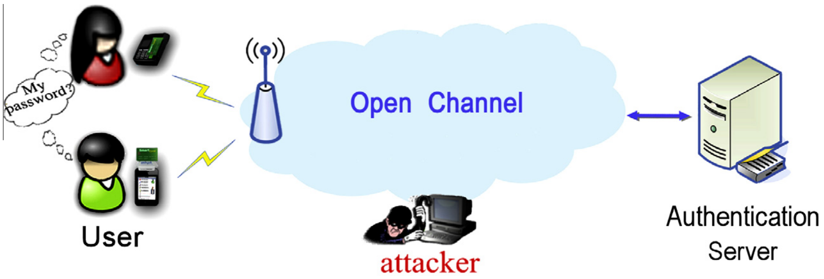

Fig. 1. Smart card based password authentication.

which guarantees that the adversary could not determine the real identity of the user. Comparatively, a more admired property of user anonymity is user un-traceability (i.e., “initiator un-traceability” in [41] or “non-linkability” in [38]), which means that the adversary can neither discern who the user is nor tell apart whether two conversations are originated from the same (unknown) user. This stronger notion of user anonymity has been adopted in most two-factor authentication schemes that endeavor to preserve user privacy [36,39,49,61,98,103]. Throughout this paper, unless otherwise specified, by “user anonymity” we will always mean “*user un-traceability*”.

1.1. Related work

In 2004, to provide user anonymity, Das et al. [22] proposed the first dynamic ID-based two-factor authentication scheme, in which the user’s real identity is concealed in session-variant pseudo-identities and also there is no other user-specific information leaked through the protocol transcripts. In this way, the user’s real identity is protected and an outsider cannot trace different sessions belonging to the same user. This scheme only involves lightweight one-way hash functions and thus is highly suitable for smart card environments.

Unfortunately, in 2009 Wang et al. [93] pointed out that Das et al.’s seminal scheme [22] is completely insecure for its independence of using passwords and fails to provide mutual authentication and user anonymity. Accordingly, they suggested an improved version, which was later found incapable of providing user anonymity and prone to impersonation attack by Yeh et al. [108] and Wen–Li [96], respectively. To the designers’ disappointment, both the enhancements of Yeh et al. and Wen–Li have been found vulnerable to the most damaging attack – offline dictionary attack [63]. In fact, attacks against most of previous protocols with only a heuristic analysis have been shown [63,77,87,88,99], suggesting the need for rigorous proofs of security in a formal, well-defined model.

In ICISC 2012, Kim–Kim [50] observed that previous dynamic ID-based schemes with each having its own merits and demerits, and developed a simple and elegant anonymous authentication scheme using smart cards. This protocol is efficient for resource-limited mobile devices (smart cards) in terms of computation, communication and storage overheads, and keeps the merits (e.g., user anonymity, sound repairability, half-forward secrecy and robust security) of the most known schemes [30,61,106,108]. Hence, it exhibits great application potentiality. In order to preserve user anonymity, this protocol adopts a synchronization mechanism to maintain consistency of the one-time pseudonym identities between the user and the server. As we shall show later, it is just this synchronization mechanism that makes Kim–Kim’s scheme vulnerable to de-synchronization attack, in which an attacker can easily break the consistency of pseudonym identities shared between the victim user and the server, and thereafter the victim user will always fail to login unless she re-registers.

1.2. Motivations

To the best of our knowledge, though de-synchronization attack can hardly be seen as a new kind of threat, and actually it has been intensively discussed in the cryptographic protocol community (e.g., RFID authentication protocols [86,107] and multimedia watermarking schemes [71]), yet *little attention has been paid to this damaging threat in the domain of two-factor authentication*. Unsurprisingly, quite a number of newly proposed two-factor schemes [44,45,52,55,61,92,97,110], which strive to achieve user anonymity by employing a similar strategy to that of Kim–Kim’s scheme [50], invariably suffer from this vulnerability. In these schemes, an attacker who merely modifies or blocks a single message flow (e.g., the second flow of [44,52,61], third flow of [55], fourth flow of [97]), can render the user unable to be authenticated by the server in any of her following protocol runs. This once again underlines the importance of being fully aware of potential threats when designing a complex protocol. What is more, though various countermeasures to the problem of de-synchronization have been suggested in other protocol domains (see, e.g. [13,58,71]), as we will show in Section 3.1, they cannot be readily applied to the domain of two-factor authentication, and thus *how to remedy a two-factor scheme with the de-synchronization problem is left as a largely unexplored issue.*

There have been a number of privacy-preserving two-factor schemes [15,49,52,55,61,96,100] proposed, nevertheless, most of them are either inefficient to be implemented on smart cards (e.g., the scheme in [100]), or insecure against some serious threats like offline dictionary attack (e.g., the schemes in [15,49,96]) and de-synchronization attack (e.g., the schemes in [52,55,61]). It still remains an open problem as to *how to develop a privacy-preserving two-factor protocol that can guard against various known attacks while maintaining acceptable efficiency*. Most concretely, the current crux lies in simultaneously preserving user privacy without de-synchronization problem, achieving truly two-factor security and maintaining high efficiency [94,88], under the non-tamper resistance assumption of the smart cards.

1.3. Contributions

In a nutshell, the contributions of this paper are threefold:

- We use Kim–Kim's two-factor authentication scheme as a case study and demonstrate that quite a number of latest privacy-preserving two-factor schemes (e.g., [44,52,95,97,110]) are prone to de-synchronization attack. In other words, the user privacy of these schemes is preserved at the cost of significantly reduced usability, which indicates the infeasibility of their strategy for achieving user anonymity. Our results highlight the devastating threat of de-synchronization attack on two-factor schemes with user anonymity.
- As our main contribution, we put forward an improvement over Kim–Kim's scheme. Our new scheme makes up the missing security provisions necessary for real-life application environments while keeping the desirable features of the original protocol. Interestingly, the performance evaluation demonstrates that it attains the property of user anonymity "for free", i.e., without incurring additional cost (in terms of communication, computation and storage) as compared to related non-privacy-preserving schemes.
- We further prove the proposed scheme in the random oracle model and the ideal cipher model. Remarkably, for the first time the notion of user un-traceability is captured in a formal model for smart-card-based password authentication.

The rest of the paper is organized as follows: Section 2 briefly reviews Kim–Kim's scheme. After that, we describe its security pitfalls in Section 3. Our improved scheme is presented in Section 4. Section 5 and Section 6 provide the security analysis and performance evaluation of the proposed scheme, respectively. Section 7 concludes with some directions for future research.

2. Review of Kim–Kim's scheme

For a self-contained discussion, in this section we briefly review Kim–Kim's scheme [50]. Their scheme consists of three phases, namely, registration, pre-computation, authentication and key exchange. For ease of presentation, we employ some intuitive notations as listed in Table 1.

2.1. Registration phase

Before user registration, the server S selects a random number b as his private key and computes $y_s = g^b \bmod p$ as the corresponding public key. This phase involves the following operations:

(1) User A chooses her identity ID_A, password PW_A.
(2) $A \Rightarrow S : \{ID_A, PW_A\}$.
(3) On receiving the registration message from A, the server selects a random number t_A and computes $v_A = h(PW_A \oplus t_A), f_A = h(PW_A \| t_A \| ID_A)$. S chooses a pseudo-identity $PID_{A,0}$ and creates a new entry $\{PID_{A,0}, v_A, f_A\}$ for A in the backend database.
(4) $S \Rightarrow A$: A smart card containing parameters $\{PID_{A,0}, y_s, h(\cdot), t_A, g, p, q\}$.[2]

2.2. Pre-computation phase

To reduce the computational overhead in the authentication and key exchange phase, user A employs the pre-computation technique:

(1) The smart card selects $x \in_R \mathbb{Z}_q^*$, and computes $y_A = g^x \bmod p$ and $c_A = (y_s)^x = g^{bx} \bmod p$;
(2) The smart card stores $\{c_A, y_A\}$ in its memory.

2.3. Authentication and key exchange phase

When A wants to login to S (assume it is A's ith login), the following operations will be performed:

(1) A inserts her smart card into the card reader, and inputs ID_A and PW_A.

[2] In [50], only the parameters $\{PID_{A,0}, t_A\}$ are explicitly stated to be stored in the smart card, and we find the other parameters $\{y_s, h(\cdot), g, p, q\}$ are also necessary when the user computes her login requests.

Table 1
Notations and abbreviations.

Symbol	Description	Symbol	Description
A	A legitimate user	$\|$	The string concatenation operation
S	Remote server	p, q	Two large prime numbers such that $q \mid p-1$
$\mathcal{M}$	Malicious attacker	g	A primitive root of group $\mathbb{Z}_q^*$
ID_A	Identity of user A	y_s, b	Public-key and private-key of server S
PW_A	Password of user A	$\oplus$	The bitwise XOR operation
$\Rightarrow$	A secure channel	$E/D(\cdot)$	Symmetric encryption/decryption algorithm
$\rightarrow$	A common channel	$h(\cdot)$	Collision free one-way hash function

(2) The smart card computes $f_A = h(PW_A \| t_A \| ID_A)$ and $e_A = E_{f_A}(y_A)$.
(3) $A \rightarrow S : \{e_A, PID_{A,i}\}$.
(4) Upon receiving the login request, S retrieves v_A and f_A from the database by using $PID_{A,i}$.
(5) S obtains y_A by decrypting e_A using f_A, then computes $c_A = y_A^b = g^{bx} \texttt{mod}\ p$. S selects $r_s \in_R \mathbb{Z}_q^*$, computes $sk = h(c_A \| r_s)$ and $M_s = h(sk \| v_A)$.
(6) $S \rightarrow A : \{r_s, M_s\}$.
(7) Upon receiving the response, A computes $sk^* = h(c_A \| r_s)$ and $v_A = h(PW_A \oplus t_A)$, and checks $M_s \overset{?}{=} h(sk^* \| v_A)$. If the verification fails, the session is terminated. Otherwise, it implies that sk^* equals sk (which is computed by S), and A proceeds to the next step.
(8) A computes $PID_{A,i+1} = h(PID_{A,i} \oplus sk)$ and replaces $PID_{A,i}$ with $PID_{A,i+1}$ in the card memory, then computes $M_A = E_{sk}(PW_A \oplus t_A)$.
(9) $A \rightarrow S : \{M_A\}$.
(10) Upon receiving the response from A, S obtains $PW_A \oplus t_A$ by decrypting M_A using sk and computes $v_A^* = h(PW_A \oplus t_A)$.
(11) S checks whether the computed v_A^* equals the stored v_A. If they are not equal, the session is terminated; otherwise, it indicates that A is a valid user. Then S computes $PID_{A,i+1} = h(PID_{A,i} \oplus sk)$ and replaces $PID_{A,i}$ with $PID_{A,i+1}$ in the backend database.

3. Cryptanalysis of Kim–Kim's scheme

Although Kim–Kim's scheme [50] possesses many desirable features such as user anonymity, sound repairability, half-forward secrecy and high efficiency, it fails to withstand de-synchronization attack (a kind of denial of service attack), known key attack and privileged insider attack. Before these security flaws are fixed, this protocol is not suitable for the real world applications. The widely accepted adversary model [57,88,94] to analyze the security of authentication protocols based on smart cards assumes an attacker with the following capabilities:

(i) $\mathcal{M}$ can fully control the communication channel between the user and the server. In other words, she can intercept, inject, modify, block, and delete messages exchanged in the channel at will. This assumption is consistent with the Dolev–Yao model.
(ii) $\mathcal{M}$ is able to either (1) compromise the user's smart card through side-channel attacks when getting access to the smart card for a relatively long period of time (e.g., a few hours [4,94]) or (2) comprise the user's password (e.g., by malicious card reader [26] or shoulder-surfing [105]), but not both (1) and (2). Clearly, if $\mathcal{M}$ compromises both factors, there is no way to prevent $\mathcal{M}$ from impersonating the user. This is a trivial case.
(iii) In some situations, $\mathcal{M}$ may learn the server's long-term key, or previous session keys. This assumption enables us to deal with forward secrecy and known key attack.[3]

The above three assumptions (i)–(iii) are also implicitly made in Kim–Kim's scheme [50], and we will evaluate its security based on these three assumptions. It is worth noting that, *our attacks described below only needs an adversary with the capabilities of (i) and/or (iii)*, which means they do not exploit the physical properties of the card (i.e., without the involvement of Assumption (ii)), and in this light, they are rather practical and effective. Still, these three assumptions are taken into consideration when we design and analyze our new scheme.

3.1. De-synchronization attack

Kim–Kim's scheme employs the one-time identity $PID_{A,i}$ to provide user anonymity, and it needs an additional synchronization mechanism to maintain the consistency of the one-time identity between the user and the server: in Step 8 of the

[3] As pointed out to us by Prof. Michael Scott, any protocol (no matter password-only authentication or smart-card-based password authentication) in which the authentication server also acts as the registration center of clients cannot resist key compromise impersonation (KCI) attack, and this observation has been corroborated by a recent interesting work [101]. Hence, the resistance to KCI attack shall not be considered as a security requirement.

authentication phase (right after A has authenticated S), user A replaces $PID_{A,i}$ with $PID_{A,i+1}$ in the card memory; in Step 11 of the authentication phase (right after S has verified A), S replaces $PID_{A,i}$ with $PID_{A,i+1}$ in the database. In this way, both A and S keep the same one-time identity $PID_{A,i+1}$ that will be used in A's next login request.

We notice that, once this consistency is broken, the user will no longer be able to login the server. Actually, many factors can lead to inconsistency between the two parties. Let us give a concrete example. Assume the user A has performed Step 8 (it means A has replaced $PID_{A,i}$ with $PID_{A,i+1}$ in the card memory) and sends out M_A to S as specified in Step 9. Before M_A reaches S, the attacker $\mathcal{M}$ intercepts M_A and blocks it. As a result, S will reject A's ith login request, and of course, S will not update $PID_{A,i}$ with $PID_{A,i+1}$ in the database. In this way, the consistency of the one-time identity between A and S is eroded. From then on, A's next login request $\{PID_{A,i+1}, e_A\}$ will always be rejected by S, for there is no entry corresponding to $PID_{A,i+1}$ in S's database.

This attack is summarized in Fig. 2. It is worth noting that, there is no easy way to work out the problem on how to maintain the consistency of the one-time identities between the user and the server. Moreover, in the above attack, the attacker $\mathcal{M}$ only needs to block one login request sent by the victim, she does not need to perform any expensive operations such as searching through a password dictionary or extracting the security parameters stored in her own or the victim's smart card. In this regard, the above attack is rather effective and practical. In conclusion, Kim–Kim's scheme is prone to a kind of denial of service attack which (completely) prevents the user from accessing the service provided by the server.

Remark 1. One may think that, if user A defers replacing $PID_{A,i}$ with $PID_{A,i+1}$ and an 'acknowledgement' step is added to the authentication phase (i.e., the fourth message flow, which is from server to user, is added), then the issue will be addressed. Now our question is, what will happen if the attacker simply blocks the last (fourth) message flow? Obviously, this approach will not work. Further, it is not difficult to see that the above de-synchronization issue cannot be well tackled by just adding new message flow(s) without radical changes of the original protocol specifications (i.e., the first three flows). In addition, as communication is energy-intensive and time-consuming, increasing the number of protocol flows will greatly impair efficiency.

Another intuitive fix is to store both $PID_{A,i}$ and $PID_{A,i+1}$ on the card memory. If a login with $PID_{A,i+1}$ fails, this indicates S has not updated the user pseudo-identity $PID_{A,i}$ to $PID_{A,i+1}$, and therefore user A switches back to use $PID_{A,i}$ to login. However, this fix not only incurs more computation and communication overheads, but also violates the notion of user un-traceability, for $PID_{A,i}$ appears more than once in the protocol transcripts and any conversations that involve $PID_{A,i}$ can be easily linked together.

Similarly, the server may keep both $PID_{A,i}$ and $PID_{A,i+1}$ in its backend database to recover from a de-synchronization. This is possible, because after sending the second flow, the server can compute $PID_{A,i+1} = PID_{A,i} \oplus sk$, irrespective of whether it will receive the third flow. Unfortunately, there is no way for the user to know which pseudo-identity ($PID_{A,i}$ or $PID_{A,i+1}$) has been accepted by the server unless the server responds with some feedback in the second flow. Yet, how this feedback can be securely delivered to A poses a new challenging problem.

In a nutshell, though the above patches (which are inspired by countermeasures to de-synchronization in other protocol domains like [58,13,71]) alleviate the situation, they are far from satisfactory and only radical cryptographic changes might completely eliminate the problem.

Remark 2. As privacy-preserving two-factor authentication schemes have only recently attracted focused scientific interest, relatively little rationale is uncovered. Unsurprisingly, similar structural mistakes are repeated time and time again. As with Kim–Kim's scheme, a number of schemes attempt to employ some synchronization mechanism(s) to maintain the

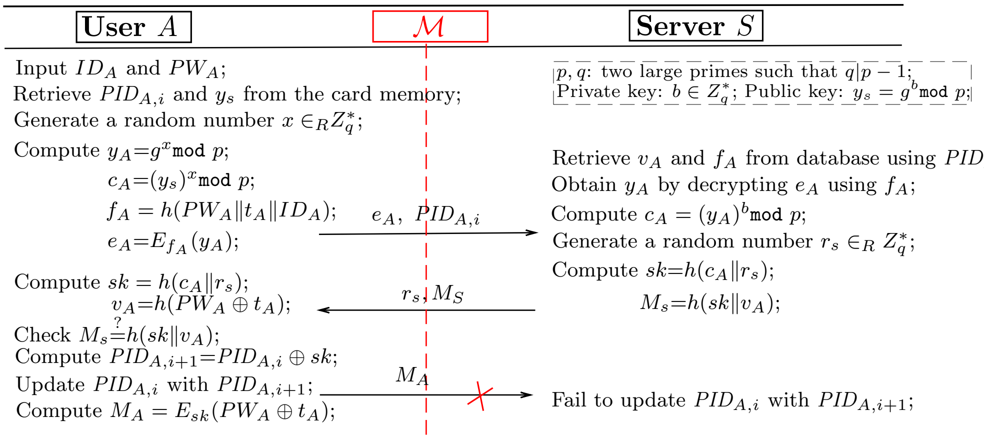
User A
$\mathcal{M}$
Server S
Input ID_A and PW_A;
Retrieve $PID_{A,i}$ and y_s from the card memory;
Generate a random number $x \in_R Z_q^*$;
Compute $y_A = g^x \bmod p$;
$c_A = (y_s)^x \bmod p$;
$f_A = h(PW_A \| t_A \| ID_A)$;
$e_A = E_{f_A}(y_A)$;
e_A, $PID_{A,i}$
p, q: two large primes such that $q|p-1$;
Private key: $b \in Z_q^*$; Public key: $y_s = g^b \bmod p$;
Retrieve v_A and f_A from database using $PID_{A,i}$;
Obtain y_A by decrypting e_A using f_A;
Compute $c_A = (y_A)^b \bmod p$;
Generate a random number $r_s \in_R Z_q^*$;
Compute $sk = h(c_A \| r_s)$;
$M_s = h(sk \| v_A)$;
r_s, M_S
Compute $sk = h(c_A \| r_s)$;
$v_A = h(PW_A \oplus t_A)$;
Check $M_s \overset{?}{=} h(sk \| v_A)$;
Compute $PID_{A,i+1} = PID_{A,i} \oplus sk$;
Update $PID_{A,i}$ with $PID_{A,i+1}$;
Compute $M_A = E_{sk}(PW_A \oplus t_A)$;
M_A
Fail to update $PID_{A,i}$ with $PID_{A,i+1}$;

Fig. 2. De-synchronization attack on Kim–Kim's scheme.

consistence of the one-time identity between the user and the server. We have reviewed more than one hundred and fifty recently proposed schemes (some of our recent cryptanalysis results include [63,87,88,90]), and observed that all these schemes using similar strategies to achieve user anonymity are prone to the above de-synchronization attack: the synchronization of one-time identities between the user and the server is broken by only blocking a single message flow. Quite recent ones include [44,52,95,97,110]. We do not detail the analysis to avoid repetition. We conjecture that the strategy of using synchronization mechanisms to achieve user anonymity is intrinsically infeasible, and promote the formal proof of this conjecture as one open problem.

Note that, recently this threat has also been mentioned in [87,89], which point out the de-synchronization vulnerability of the schemes in [55,83], yet no viable solution to cope with this kind of problem has been given. In this work, besides highlighting the necessity of special attention to this threat when designing schemes with user anonymity, we further propose a new scheme that can completely eliminate this threat while keeping high efficiency and achieving provable security. It is also worth noting that, de-synchronization attack can hardly be considered as a new kind of security threat in the broad spectrum of cryptographic protocols, and its practicality and seriousness have been intensively discussed in the cryptography community (e.g., RFID authentication protocols [86,107] and multimedia watermarking schemes [71]). Yet, as far as we know, little attention has been given to this destructive threat *in the domain of two-factor authentication.* This, once again, suggests the importance of being aware of potential attacks when designing a complex cryptographic protocol.

3.2. Privileged-insider attack

In 2007, Florencio and Herley [32] conducted a large scale study of password use and password re-use habits, obtained data from more than half a million users over a period of three months and showed that the average user has about 6.5 passwords and 25 accounts that require passwords, which means each password is shared across 3.9 different sites. This research well reveals the reality that, with an ever increasing number of password accounts and little improvement of human cognition and memory capacity, users tend to use the same password to access multiple servers for their convenience. In 2014, a work by Das et al. [21] further confirmed this bad practice of common users, reporting that 43.51% of users reuse the same password across multiple sites.

In this case, if a privileged insider of the server, e.g., the administrator, has obtained the user's password, she may abuse it to access other servers on behalf of the victim. In the registration phase of Kim–Kim's scheme [50], the user directly submits the password PW_A to S. With the knowledge of PW_A, the privileged insider of S can try to use it to impersonate A to access other service servers. Although it may be possible that A never uses the same password to access other servers and all the privileged insiders of S can be trusted, the implementers and the users of the scheme should be aware of such a potential risk.

3.3. Known key attack

As noted in [51,80], the resistance to known key attack is an important security requirement for authenticated key agreement schemes. Our concern is the realistic possibility that a session key may be leaked to an adversary. In reality, this may happen in various ways such as improper erasure of session keys after use, the malicious action of an insider or a temporary break-in. When any of these situations occurs, the exposed session is obviously insecure. But what a protocol with resistance to known key attack can guarantee is that, such an exposure will only affect the specific compromised session, other sessions established by the same or other parties will not be endangered by this leakage.

Without loss of generality, suppose a legitimate user A's jth session key, denoted by $sk^j = h(c_A \| r_s)$, is leaked. With $sk^j, \mathcal{M}$ can obtain $PW_A \oplus t_A = D_{sk}(M_A)$, where M_A is intercepted during A's jth authentication process. Further, $\mathcal{M}$ can compute the security parameter $v_A = h(PW_A \oplus t_A)$. Once the long-term security parameters v_A and $PW_A \oplus t_A$ stored in the server's database are leaked, the entire system is under great risk of failure. In conclusion, once a session key exchanged during one session is leaked to an adversary, the other unexposed sessions will be endangered. This is a serious vulnerability that violates the fundamental security principle [51] that the disclosure of session-specific information should not affect other sessions established by the same or other parties.

4. Our improved scheme

In this section, we first briefly sketch our design rationales, then present our improvement. In particular, we also reveal some subtleties and challenges in designing a privacy-preserving two-factor authentication protocol that aims to achieve "truly two-factor security".

Lessons learned from Kim–Kim's scheme motivate some of the elements in our new protocol design. Specifically, three implications for designing an efficient and secure authentication scheme with user anonymity are derived. Firstly, the strategy of sharing one-time pseudo-identities among the communicating parties to provide user anonymity needs additional synchronization mechanism to maintain consistency and this may bring new security problems, such as synchronization attack as described in Section 3.1. Fortunately, we find the dynamic-ID method introduced in [111][4] is simpler and more

[4] This scheme [111] is subject to replay attack, yet its novel method to achieve user anonymity is appealing.

robust: security and privacy are achieved at the same time by using a public-key encryption with proper padding mechanism. Secondly, to resist known key attacks, the session key shall be computed with the contribution of both session-specific secrets and long-term secrets, and the risk-taking encryption of long term security parameters using the session key should be eliminated. Finally, to withstand insider attacks, the user should not submit her password PW_A directly or in a form that is plaintext-equivalent to the password (e.g., $h(PW_A)$) to the server S. Instead, the user can submit $h(PW_A \oplus r)$ to S as the scheme in [49], where r is a random number chosen by user A herself. In practice, a user may write r on a piece of paper and can tear it up after registration. In this way, there is no need for her to remember r, and thus r can be selected as randomly as possible. With these principles in mind, we now put forward an improved scheme over Kim–Kim's scheme [50] as shown in Fig. 3.

To save space, here we only point out its key differences from the original scheme. To accommodate to the random oracle model, besides the hash function $h : \{0,1\}^* \rightarrow \{0,1\}^{l_0}$, we define three more hash functions $\mathcal{H}_i : \{0,1\}^* \rightarrow \{0,1\}^{l_i}, i = 1,2,3$, where l_i is the bit length of function output, e.g. $l_i = 160$. Moreover, the protocol involves a cipher $\mathcal{E} : \{0,1\}^{l_0} \times \mathbb{Z}_q^* \rightarrow \{0,1\}^{l_0}$. In the registration phase, the user additionally chooses a random number r_A, and submits $h(PW_A \oplus r_A)$ instead of $h(PW_A)$ to S. Note that, at the end of this phase, r_A is stored in the card and the user does not need to remember it any more.

As for the authentication phase, there are mainly three changes. Firstly, to provide user anonymity, we adopt the intrinsic *idea* of ElGamal encryption scheme but not the exact ElGamal encryption scheme itself. More specifically, in the ElGamal encryption scheme, message m is concealed in the multiplication $y_s^x \cdot m$, where y_s is the server's public key, x is a random number selected by the user (sender) and g is the generator, as illustrated in Table 1. As a result, only the server with the private key b corresponding to y_s can recover m by computing $m = (y_s^x \cdot m)/(g^x)^b$. Inspired by ElGamal's exploitation of the Diffie–Hellman one-way trapdoor function, in our scheme we conceal ID_A in the pseudo-identity $DID_A = ID_A \oplus \mathcal{H}_1(g^x \| g^{bx})$, and only the party with the private key b corresponding to y_s can obtain ID_A by "decrypting" $DID_A : ID_A = DID_A \oplus \mathcal{H}_1(g^x \| (g^x)^b)$. Secondly, in some similarity with ID_A, f_A is encrypted in e_A by using $c_A = g^{bx} \bmod p$ as the encryption key. As we shall see later, this instantiation is quite subtle. Thirdly, the authenticator M_A is computed with the help of a hash function but not an encryption algorithm, which ensures message integrity while eliminating the risk of exposure of the long-term security parameters (such as v_A and $PW_A \oplus t_A$ in [50]). Though the protocol itself seems simple and intuitive, assuring its security is quite tricky.

It is well known that the ElGamal encryption scheme coupled with a proper padding scheme (e.g., Fujisaki–Okamoto's generic padding framework [33]) can achieve provable security in the random oracle model while maintaining high efficiency. Since our above treatment of the Diffie–Hellman one-way trapdoor function is consistent with the ElGamal encryption scheme and our padding strategy is in line with the approach in [33], our scheme retains the potential to achieve provable security.

Some Notes. In the following, we highlight *some subtleties*in our new scheme, which demonstrates the error-prone nature of two-factor authentication schemes and particularly reveals the great challenges in designing a privacy-preserving protocol against offline dictionary attack.

Regarding the authentication phase, one question is likely to arise: *why user A is verified by S twice? Is this necessary?* It is evident that S must check the legitimacy of A in the third protocol flow (see Fig. 3), otherwise a replay attack will occur: $\mathcal{M}$ simply replays the first message to impersonate A. Now it remains to show the necessity of checking the legitimacy of A before S sends back the second protocol flow. We justify it by showing that the protocol is weakened when certain minor modifications are made.

Note 1. Suppose S does not need to check the legitimacy of A upon receiving the first flow. Then, e_A is not necessary to be computed and sent to S. In this situation, once $\mathcal{M}$ has obtained A's smart card and the identity $ID_A, \mathcal{M}$ can launch an offline dictionary attack as follows:

Step 1. Selects a random value $x \in \mathbb{Z}_q^*$, and computes $y_A = g^x \bmod p, c_A = (y_s)^x = g^{bx} \bmod p$ and $DID_A = ID_A \oplus \mathcal{H}_1(y_A \| c_A)$;
Step 2. $\mathcal{A} \rightarrow S : \{DID_A, y_A\}$;
Step 3. Upon receiving the login request from A (actually from $\mathcal{M}$), S will find no abnormality and proceed as usual.
Step 4. Upon receiving the response $\{r_s, M_s\}$ from $S, \mathcal{M}$ proceeds to the next step;
Step 5. Randomly chooses a password PW_A^* from the password space $\mathcal{D}_{pw}$.
Step 6. Computes $v_A^* = h(h(PW_A^* \| r_A) \oplus t_A)$ and $sk^* = \mathcal{H}_2(c_A \| r_s \| v_A^*)$, where r_A and t_A are extracted from A's smart card, and c_A is computed by $\mathcal{M}$ herself in Step 1;
Step 7. Verifies the correctness of PW_A^* by checking $M_s \overset{?}{=} \mathcal{H}_3(sk^* \| v_A^*)$. If the equality does not hold, goes back to Step 1.
Step 8. Repeats Steps $5 \sim 7$ until the correct value of PW_A is found.

The above attack justifies the "seemingly" redundant (but essential) practice that user A is verified by S twice – otherwise, the proposed scheme cannot attain truly two-factor security: a compromise of the smart-card factor leads to the exposure of the remaining factor, i.e. the password factor. It is worth noting that our attack presumes that $\mathcal{M}$ can determine A's identity ID_A. This assumption is quite realistic. Firstly, user identity is static and often confined to a predefined format, therefore, it can be more easily guessed than the password, e.g. $|\mathcal{D}_{id}| \leqslant |\mathcal{D}_{pw}| \leqslant 10^6$ [8,9,25], where $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the number of identities in identity space $\mathcal{D}_{id}$ and the number of passwords in password space $\mathcal{D}_{pw}$, respectively. Secondly, user identity is usually displayed in plain on the screen and is susceptible to shoulder-surfing. What is more, in many practical scenarios an

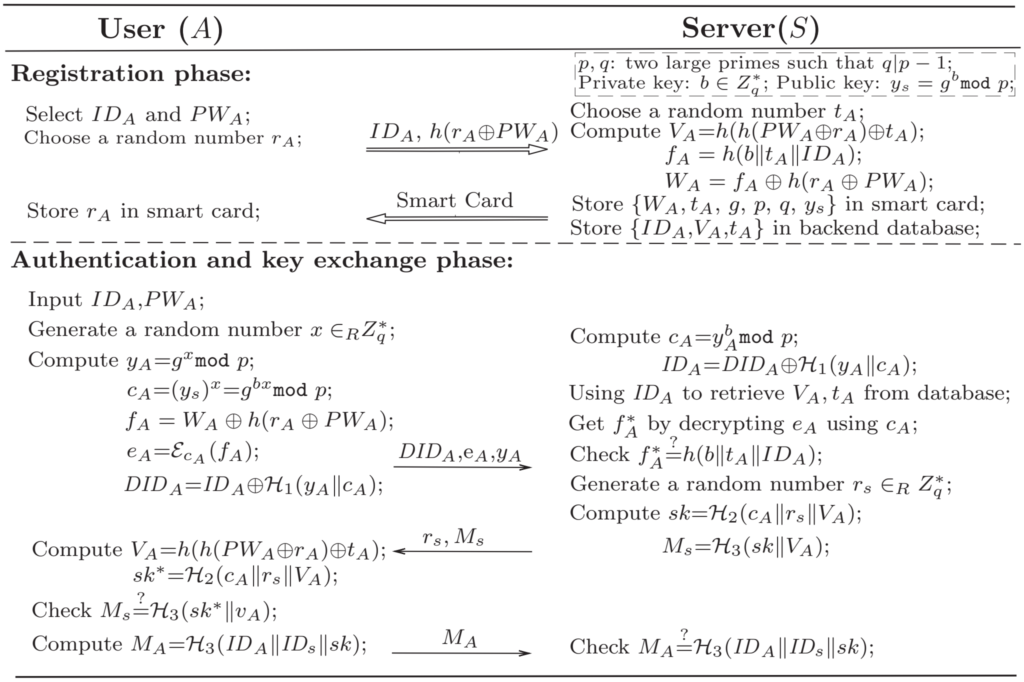
User (A)
Server(S)
Registration phase:
p, q: two large primes such that $q|p-1$;
Private key: $b \in Z_q^*$; Public key: $y_s = g^b \bmod p$;
Select ID_A and PW_A;
Choose a random number r_A;
$ID_A,\ h(r_A \oplus PW_A)$
Choose a random number t_A;
Compute $V_A = h(h(PW_A \oplus r_A) \oplus t_A)$;
$f_A = h(b \| t_A \| ID_A)$;
$W_A = f_A \oplus h(r_A \oplus PW_A)$;
Store r_A in smart card;
Smart Card
Store $\{W_A, t_A,\ g,\ p,\ q,\ y_s\}$ in smart card;
Store $\{ID_A, V_A, t_A\}$ in backend database;
Authentication and key exchange phase:
Input ID_A, PW_A;
Generate a random number $x \in_R Z_q^*$;
Compute $y_A = g^x \bmod p$;
$c_A = (y_s)^x = g^{bx} \bmod p$;
$f_A = W_A \oplus h(r_A \oplus PW_A)$;
$e_A = \mathcal{E}_{c_A}(f_A)$;
$DID_A = ID_A \oplus \mathcal{H}_1(y_A \| c_A)$;
DID_A, e_A, y_A
Compute $c_A = y_A^b \bmod p$;
$ID_A = DID_A \oplus \mathcal{H}_1(y_A \| c_A)$;
Using ID_A to retrieve V_A, t_A from database;
Get f_A^* by decrypting e_A using c_A;
Check $f_A^* \overset{?}{=} h(b \| t_A \| ID_A)$;
Generate a random number $r_s \in_R Z_q^*$;
Compute $sk = \mathcal{H}_2(c_A \| r_s \| V_A)$;
$M_s = \mathcal{H}_3(sk \| V_A)$;
r_s, M_s
Compute $V_A = h(h(PW_A \oplus r_A) \oplus t_A)$;
$sk^* = \mathcal{H}_2(c_A \| r_s \| V_A)$;
Check $M_s \overset{?}{=} \mathcal{H}_3(sk^* \| v_A)$;
Compute $M_A = \mathcal{H}_3(ID_A \| ID_s \| sk)$;
M_A
Check $M_A \overset{?}{=} \mathcal{H}_3(ID_A \| ID_s \| sk)$;

Fig. 3. A new privacy-preserving two-factor authentication scheme.

attacker can easily get hold of a large database of valid user names, e.g. email list, bank account list [72]. Last but not least, $\mathcal{M}$ can know more or less about the personal information of the victim when she has got the victim's smart card.

Regarding how to compute e_A, there are two subtleties to be noted, and any negligence may lead to serious security loopholes. Accordingly, two more notes regarding these two subtleties are detailed in Appendix A.

5. Formal security analysis

In this section, we show that our improved scheme is provably secure in both the random-oracle model and the ideal-cipher model.[5] In the random-oracle model, a hash function is modeled as an oracle which returns a random value for each new query. If the same query is asked twice, identical answers will be returned. In the ideal-cipher model, the encryption/decryption algorithm is assumed to have a permutation property. In the following analysis, we also assume the intractability of the computational Diffie–Hellman problem.

5.1. Security model

Our formal model of security for smart-card-based password authentication protocols is based on the reified BPR2000 security model proposed by Bresson et al. [11]. In addition, we adopt the technique developed by Wang et al. [91] so that we can define some special security requirements (e.g., resistance to smart card loss attack) for password authentication schemes using smart cards. In the following, we define the operations of the adversary and formulate the definition of security. Readers may refer to [11,91] for more details.

Players. We denote a server $S \in \mathsf{Server}$ and a user (client) $U \in \mathsf{User}$ that can participate in the two-factor authentication protocol $\mathcal{P}$. Each participant may have several instances called oracles involved in distinct, possibly concurrent, executions of $\mathcal{P}$. We denote client instances and server instances by U^i and $S^j, i, j \in \mathbb{Z}$, respectively. Further, we denote any kind of instance by I.

Initialization. According to our scheme, the server holds a long-term private/public key pair $(b, y = g^b)$. User U_A possesses a password PW_A uniformly drawn from a small dictionary $\mathcal{D}$ of size $|\mathcal{D}|$ and a smart card personalized with the parameters $\{t_A, g, p, q, y_s\}$. Additionally, when the user U_A enrolls in the server S, S stores V_A and f_A in the backend database, where V_A and f_A are (injective) transformations of PW_A and t_A. For simplicity, we assume the password is uniformly distributed. Note that everything would work with any other distribution, in which one can replace the probability $\frac{\lambda}{|\mathcal{D}|}$ by the sum of the probabilities of the λ most probable passwords.

[5] These two models have recently been shown to be equivalent [20].

Queries. The interaction between an adversary $\mathcal{A}$ and the protocol participants is modeled via access to oracles whose inputs may range over $I \in \mathsf{User} \cup \mathsf{Server}$ and $i \in \mathbb{Z}$. In this way, the adversary's capabilities are captured. The session identifier sid_I^i is defined to keep track of the different executions of a particular participant I. Without loss of generality, sid_I^i is assigned the value of the (ordered) concatenation of all messages sent and received by instance I^i as it does in [48]. The query types available to $\mathcal{A}$ are defined as follows:

- $\mathsf{Execute}(U^i, S^j)$: This oracle call represents passive eavesdropping of a protocol run, and its output consists of the messages that were exchanged during the honest protocol execution.
- $\mathsf{Send}(I^i, m)$: This query models an active attack, in which the adversary constructs a message m by intercepting a message and then falsifying it, or creating it by herself. The message m is delivered to instance I^i and $\mathcal{A}$ gets back the response that I^i generates in processing message m according to the protocol specification. A query Send(U^i, Start) initializes the protocol $\mathcal{P}$.
- $\mathsf{Test}(I^i)$: This oracle query is not used to simulate the adversary's capability, but to define session key's semantic security. This query can be only called once, at any time during the adversary's execution. If no session key for instance I^i is defined, then the symbol $\perp$ is returned. Otherwise, a private coin c is flipped. If $c = 1$, the session key sk is returned to $\mathcal{A}$, otherwise $\mathcal{A}$ is given a random key of the same size.
- $\mathsf{Reveal}(I^i)$: This query models the misuse of session keys. It outputs I^i's session key sk to the adversary if the targeted instance actually "holds" a session key. Otherwise the $\perp$ is returned.
- $\mathsf{Corrupt}(I^i, a)$: This query models the corruption capabilities of the adversary. $\mathcal{A}$ can *break either the smart card factor* of the user by side-channel attack *or the password factor* of the user by using a malicious card reader, shoulder surfing or social engineering, but is restricted from breaking both factors to avoid the trivial case:
  - If $I = U, a = 1$, it outputs the password PW_A of U_A; If $I = U, a = 2$, it outputs parameters $\{t_A, g, p, q, y_s\}$ that are stored in the smart card.
  - If $I = S, a = 1$, it outputs the private key b of S; If $I = S, a = 2$, it outputs the entry $\{ID_A, V_A, f_A\}$ stored in the backend database.

It is easy to see that, the above oracle queries indeed can model all the adversary capabilities listed at the beginning of Section 3.

Partnering. The notion of partnering is fundamental in defining both correctness and security. We say that the instances U^i and S^j are partnered if the following conditions are satisfied: ① Both instances have accepted. ② Both instances shared the same session identifier (sid), i.e. sid_U^i=sid_S^j. ③ The partner identifier (pid) of U^i is S and vice versa.

Freshness. The freshness notion captures the intuitive fact that a session key cannot be trivially known to the adversary. We say that an instance I is fresh if: ① I has accepted and computed a session key. ② Neither I nor its partner have been asked for a Reveal-query. ③ At most one kind of Corrupt-query is made to I *or* its partner from the beginning of the game.[6] Note that, without these italicized words, the adversary will be given more power than it shall be allowed (see the assumptions listed in Section 3). For example, if the queries $\mathsf{Corrupt}(U, 2)$ and $\mathsf{Corrupt}(S, 1)$ can be made simultaneously by an adversary, no existing two-factor scheme, as far as we know, can resist offline dictionary attack.

Correctness. If U^i and S^j are partnered and they are accepted, then they end up with the same session key $sk_U^i = sk_S^j$.

Advantage of the adversary. The major concern of authentication schemes with key agreement is to protect the privacy of the session key. In a protocol execution of $\mathcal{P}$, an adversary $\mathcal{A}$ can ask a polynomial number of $\mathsf{Execute}$-query, Reveal-query, $\mathsf{Corrupt}$-query and Send-query. We say an adversary $\mathcal{A}$ succeeds if it makes a single Test-query to a fresh instance I^k and outputs a guessed bit c' for the bit c involved in the Test-query where $c' = c$. We denote this event by Succ. Accordingly, the advantage of $\mathcal{A}$ in attacking protocol $\mathcal{P}$ is defined to be

$$\mathsf{Adv}_{\mathcal{A},\mathcal{P}}^{ake}(\ell) = 2Pr[Succ(\mathcal{A})] - 1 = 2Pr[c' = c] - 1$$

where the probability space is taken over all the random coins of $\mathcal{A}$ and all the oracles (including the initialization phase), and ℓ is the security parameter.

It remains to define what we mean by a secure protocol. It is clear that a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ can always gain an advantage close to 1 by trying all passwords one-by-one in an on-line impersonation attack, for the size of the password dictionary is constant (and small). In such an attack, $\mathcal{A}$ can test the correctness of the guessed password PW^* by observing the decision (i.e., acceptance or rejection) of the corresponding party. For a secure protocol, we expect that $\mathcal{A}$ can only test a single password in each on-line attack. In particular, instances with which the adversary interacts via $\mathsf{Execute}$ queries are not counted as on-line attacks. As suggested in [48], we use the Send-query type to count the number of on-line guesses performed by $\mathcal{A}$. Based on this idea, we have the following definition of secure smart-card-based password authentication protocol:

[6] These italicized word 'or' is a substitution of the word 'and' in response to a insightful suggestion by Dr. Debiao He from Wuhan University, and we are grateful to him for helping us to observe this key point.

Definition 1 ($\mathtt{Semantic\ Security}$). Protocol $\mathcal{P}$ is a secure smart-card-based password authentication protocol if, for every PPT adversary $\mathcal{A}$ making at most q_{send} on-line attacks, there exists a negligible function $\epsilon(\cdot)$ such that:

$$\mathsf{Adv}^{ake}_{\mathcal{A},calP}(\ell) \leqslant q_{send}/|\mathcal{D}| + \epsilon(\ell)$$

This definition ensures that polynomially-many calls to the Execute and Reveal queries (and one additional call to the Corrupt query) are of no help to an adversary, only online impersonation attack (which is harder to mount and easier to detect) is the adversary's best strategy.

The property of user anonymity (un-traceability) states that besides user U herself and the server S, no one else can, based *only* on the protocol transcripts, tell apart whether two conservations involve the same user [61,90]. To model this property, besides the queries (i.e., Execute, Send, Reveal and Corrupt) defined above, the following two additional queries are needed:

- RevealAnonymity(sid^i_U): This query returns to the adversary some static information (e.g., user identity or other unique user-related parameters) that is specific to user participant U^i of session instance sid^i_U. It models the possible violation of user un-traceability.
- TestAnonymity(sid^i_U, sid^j_U): This oracle query is not used to capture the adversary $\mathcal{A}$'s capability, but to define user anonymity. After querying the oracle, a binary status bit 1 indicating "the same user" or 0 "not the same user" will be returned according to a predefined random bit c. If $c = 1, \mathcal{A}$ will receive the bit 1 indicating "the same user"; Otherwise, $\mathcal{A}$ will receive the bit 0 indicating "not the same user". This query can be called only once.

Anonymity-fresh. We say that a session instance sid^i_U is anonymity-fresh if and only if the following two conditions are met: ① No RevealAnonymity query has been made to sid^i_U. ② At most one kind of Corrupt-query is made to U^i and no Corrupt-query is made to S from the beginning of the game.

Definition 2 ($\mathtt{User\ Anonymity/Un\text{-}traceability}$). Protocol $\mathcal{P}$ is a privacy-preserving two-factor protocol if, for every PPT adversary $\mathcal{A}$ making at most q_{send} active attacks and q_{exe} passive attacks. let $\mathsf{Succ}^{anon}(\mathcal{A})$ be the event that $\mathcal{A}$ asks a single TestAnonymity query directed to two anonymity-fresh session instances at the end of protocol $\mathcal{P}$ and manages to output a bit c' equal to the bit c that is predefined in the TestAnonymity query. We say that the authentication protocol $\mathcal{P}$ achieves user anonymity if there exists a negligible function $\epsilon(\cdot)$ such that:

$$\mathsf{Adv}^{anon}_{\mathcal{A},\mathcal{P}}(\ell) = 2\Pr[\mathsf{Succ}^{anon}(\mathcal{A})] - 1 = 2\Pr[c' = c] - 1 \leqslant \epsilon(\ell)$$

There do have been a few works [91,98,102,111] that adopt a formal method to analyze the security and privacy provisions of two-factor schemes, yet to the best of our knowledge, the models introduced in previous works can only deal with the basic notion of user anonymity (i.e., user identity-protection). In this work, we take the first step towards capturing the more advanced notion of user anonymity (i.e., user un-traceability) in the ROM model with practical assumptions.

Before presenting the security results, we briefly review the computational Diffie–Hellman (CDH) assumption and decisional Diffie–Hellman (DDH) assumption, both of which constitute the foundations of the security of our scheme.

CDH Assumption. Let $\mathbb{G}$ be a finite cyclic group of prime order q generated by an element g, where the operation is denoted multiplicatively and $|q| = k$, e.g., $k = 1024$. Let $\mathcal{A}$ be a CDH-adversary with running time at most t. The probability that $\mathcal{A}$ succeeds in computing g^{xy} from (g^x, g^y) is denoted by $\mathsf{Adv}^{\mathsf{CDH}}_{g,\mathbb{G}}(\mathcal{A})$. We further define $\mathsf{Adv}^{\mathsf{CDH}}_{g,\mathbb{G}}(t) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{CDH}}_{g,\mathbb{G}}(\mathcal{A})\}$, where the probability is taken over the random values x and y. The CDH-Assumption states that $\mathsf{Adv}^{\mathsf{CDH}}_{g,\mathbb{G}}(t) \leqslant \epsilon(\ell)$ for any $t/\epsilon(\ell)$ not too large.

DDH Assumption. Given group $\mathbb{G}$ as defined in CDH assumption, for any PPT algorithm D, the probability that D succeeds in distinguishing between (g, g^x, g^y, g^{xy}) and (g, g^x, g^y, r) is denoted by $\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathbb{G}}(D) = |\Pr[D(g, g^x, g^y, g^{xy})] - \Pr[D(g, g^x, g^y, r)]|$, where $r \in_R \mathbb{Z}^*_q$. We further define $\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathbb{G}}(t) = \max_D\{\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathbb{G}}(D)\}$, where the maximum is taken over all Ds with running time at most t. The DDH-Assumption states that $\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathrm{G}}(t) \leqslant \epsilon(\ell)$ for any $t/\epsilon(\ell)$ not too large.

5.2. Security and privacy results

Theorem 1. *Let $\mathbb{G}$ be a representative group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let $\mathcal{P}$ be the improved proposed authentication scheme stated in Section 3. Let $\mathcal{A}$ be an adversary against the semantic security within a time bound t, making less than q_{send} Send-queries, q_{exe} Execution-queries and q_h random oracle queries. Then we have*

$$\mathsf{Adv}^{ake}_{\mathcal{A},\mathcal{P}}(\ell) \leqslant \frac{q_{send}}{|\mathcal{D}|} + (8q_h + 3q_{send})\mathsf{Adv}^{\mathsf{CDH}}_{g,\mathcal{G}}(t') + \frac{q_h^2 + 6q_{send}}{2^\ell} + \frac{(q_{send} + q_{exe})^2}{p},$$

where $t' \leqslant t + (q_{send} + q_{exe} + 1) \cdot \tau_e, \tau_e$ is the computation time for an exponentiation in $\mathbb{G}$.

Proof. Let $\mathcal{A}$ be an adversary attempting to break the security of protocol $\mathcal{P}$. The idea is to employ $\mathcal{A}$ to construct adversaries for each of the underlying primitives in such a way that if $\mathcal{A}$ manages to break $\mathcal{P}$, then at least one of these adversaries succeeds in solving an underlying primitive. The detailed security proof is similar to that of one of our earlier works [91], in which the goal is accomplished through a series of hybrid games, starting with the real attack and ending in a game where $\mathcal{A}$'s advantage is 0, and for which we can bound the difference in $\mathcal{A}$'s advantage between any two consecutive games. Once $\mathcal{A}$'s capabilities has been properly modeled (captured) in the formal model, the proof itself has no much trick. Hence, we omit it for simplicity. □

Theorem 2. *Let $\mathbb{G}$ be a representative group and $\mathcal{P}$ the improved proposed authentication scheme stated in Section 3. Let $\mathcal{A}$ be an adversary against the user anonymity (more specifically, user un-traceability) property within a time bound t, making less than q_{send} Send-queries and q_{exe} Execution-queries and q_h random oracle queries. Then we have*

$$\mathrm{Adv}^{anon}_{\mathcal{A},\mathcal{P}}(\ell) \leqslant \mathcal{O}(\mathrm{Adv}^{\mathrm{DDH}}_{g,\mathbb{G}}(t))$$

Sketch of Proof. Let $\mathcal{A}$ be an adversary attempting to break the anonymity of protocol $\mathcal{P}$ with a non-negligible probability. Then we can employ $\mathcal{A}$ to construct an algorithm D with the same non-negligible probability to break the DDH assumption as follows:

D = "Given (g, g^x, g^b, g^{xb}) and (g, g^x, g^b, r) as input, where $x, r \in_R \mathbb{Z}_q^*$ and b is S's private key:

(1) Suppose D is a legitimate but curious user with her own card and password.
(2) Set $y_D = g^x, c_D = g^{xb}$ and perform an honest run (as usual) with S, we denote the session identifier of this protocol run by sid_D^i (see the definition of *sid* in Section 5.1).
(3) Set $y_D = g^x, c_D = r$ and perform an honest run (as usual) with S, we denote the session identifier of this protocol run by sid_D^j. In this run, upon receiving the first flow from user D, server S may reject and refuse to respond. In this case, D can set $r_s \in_R \mathbb{Z}_q^*, M_s \in_R \{0,1\}^{|\mathcal{H}_3(\cdot)|}$ and $M_A \in_R \{0,1\}^{|\mathcal{H}_3(\cdot)|}$ to make sid_D^j have the same structure with sid_D^i.
(4) Choose $x' \in_R \mathbb{Z}_q^*$, compute $y_D = g^{x'} \bmod p$ and $c_D = (y_s)^{x'} = g^{bx'} \bmod p$; Use y_D, c_D to perform an honest run with S, and denote the session identifier of this run by sid_D^k.
(5) Run $\mathcal{A}$ to make the queries $\mathsf{TestAnonymity}(sid_D^i, sid_D^j)$ and $\mathsf{TestAnonymity}(sid_D^i, sid_D^k)$, and denote the bits that $\mathcal{A}$ returns by b_1 and b_2, respectively.
(6) If $b_1 = b_2 = 1$, output "Both are Diffie–Hellman tuples"; if $b_1 = 1, b_2 = 0$, output "(g, g^x, g^b, r) is a Diffie–Hellman tuple", the other is not; if $b_1 = 0, b_2 = 1$, output "(g, g^x, g^b, g^{xb}) is a Diffie–Hellman tuple", the other is not; if $b_1 = b_2 = 0$, output "None is a Diffie–Hellman tuple".

It is not difficult to see that, the above algorithm D can be completed in polynomial time. While the rationalities of Steps 1–5 are quite intuitive, Step 6 needs some remarks. It is critical to note that in the protocol transcripts, the only two (implicit) static elements corresponding to user D, which can be exploited by $\mathcal{A}$ to identify this specific user, are ID_D and f_D. In any two distinct sessions that D is involved, say sid_D^m and sid_D^n, ID_D and f_D remain the same (unchanged) *if and only if* $c_D^m = CDH(y_D^m, y_s)$ and $c_D^n = CDH(y_D^n, y_s)$, due to the collision-resistant nature of random-oracle model and ideal-cipher model. This indicates $|\Pr[\mathsf{TestAnonymity}(sid_D^i, sid_D^j) = 1] - \Pr[\mathsf{TestAnonymity}(sid_D^i, sid_D^k) = 1]| = |\Pr[D(g, g^x, g^b, g^{xb}) = 1] - \Pr[D(g, g^x, g^b, r) = 1]|$, which justifies Step 6.

In the above argument, $g^y (= g^b)$ is fixed. However, g^x is dynamically changed, according to the "random self-reducible" of the DDH problem [73], we have $\mathrm{Adv}^{\mathrm{DDH}}_{g,\mathbb{G}}(D) = |\Pr[D(g, g^x, g^y, g^{xy}) = 1] - \Pr[D(g, g^x, g^y, r) = 1]| = |\Pr[D(g, g^x, g^b, g^{xb}) = 1] - \Pr[D(g, g^x, g^b, r) = 1]|$, where the exponents b is a fixed value and $x, y \in_R \mathbb{Z}_q^*$. Consequently, we obtain $\mathrm{Adv}^{\mathrm{DDH}}_{g,\mathbb{G}}(D) \geqslant |\Pr[\mathsf{TestAnonymity}(sid_D^i, sid_D^j) = 1] - \Pr[\mathsf{TestAnonymity}(sid_D^i, sid_D^k) = 1]|$. This means if $\mathcal{A}$ can violate user un-traceability, D will, with no less probability, solve the DDH problem which is believed to be hard in $\mathbb{G}$. Now a contradiction occurs, which completes the proof. □

Theorem 1 indicates that our scheme can achieve the security requirements for user authentication and key agreement with great confidence, and the well-known attacks, such as offline dictionary attack, impersonation and man-in-the-middle cannot succeed with a non-negligible probability. Other attacks (which generally are not captured by the notion of semantic security) like replay, parallel session, reflection and stolen verifier, can as well be guarded against by our scheme, which is trivial to confirm. Also worth noting is that the ideal (i.e. random-oracle and ideal-cipher) models in Theorems 1 and 2 are less desirable than a standard cryptographic assumption. To avoid using these ideal models, we could use the technique introduced in [48], which requires "only (roughly) 4 times more computation than standard Diffie–Hellman key exchange." Considering the resource-constrained nature of smart cards, provably secure schemes in the standard model are from a theoretical point of view very interesting but far from practical. For example, a 512-bit modular exponentiation will take 141 ms [76] on the popular 36 MHz MIPS-32 based smart cards, it is unacceptable to conduct several such expensive operations in

Table 2
Performance comparison among relevant authentication schemes.

	Protocol rounds	Computation overhead		Resistance to known attacks	User anonymity	Provable security	Sound repairability	Forward secrecy
		User side	Server side					
Song [79]	2^a	$T_S + 3T_H$	$T_E + T_S + 3T_H$	× [16]	×	×	×	×
Li et al. [61]	3	$2T_E + T_S + 4T_H$	$T_E + +3T_S + 8T_H$	×	✓	×	✓	×
Kim–Kim [50]	3	$2T_E + 2T_S + 5T_H$	$T_E + 2T_S + 3T_H$	×	✓	×	×	×
Chen et al. [16]	3	$2T_E + 4T_H$	$T_E + 3T_H$	× [63]	×	×	×	×
Wang et al. [91]	3	$3T_E + 8T_H$	$3T_E + 5T_H$	✓	✓	✓	✓	✓
Kumari–Khan [52]	2^a	$2T_E + T_S + 4T_H$	$T_E + T_S + 3T_H$	×	✓	×	✓	×
Jiang et al. [43]	2^a	$4T_E + 4T_H$	$2T_E + 4T_H$	✓	×	×	×	×
Our scheme	3	$2T_E + T_S + 8T_H$	$T_E + T_S + 4T_H$	✓	✓	✓	✓	×

×[*x*]: The corresponding scheme fails to meet some security requirement(s), and the cryptanalysis is carried out by the reference [*x*]. The schemes in [50,52,61] are prone to de-synchronization attack as described in Section 3.1. According to [63], all the schemes in [16,50,52,61,79] cannot provide forward secrecy due to the fact that there are less than two exponential operations conducted on the server, while Jiang et al.'s scheme [43] fail to attain this feature due to an improper formation of the session key.

[a] Schemes employing the timestamp mechanism to achieve message freshness are subject to clock synchronization problem.

one login session. While not based on standard assumptions, ideal models do provide reasonable security results and supply strong evidence that our scheme is not flawed, while offering superior computational overheads as will be discussed below.

6. Performance evaluation

To evaluate our scheme, we compare the security features and performance of our scheme with other relevant two-factor authentication schemes in this section. The reason why the schemes presented in [16,43,50,52,79,82], instead of other schemes mentioned earlier in this paper, are selected to compare with is that, these six schemes are the few ones that are based on CDH-assumption and do not (or choose not to) provide perfect forward secrecy (PFS). It is well known that PFS can be easily remedied through the standard Deffie–Hellman key exchange (but at the cost of four modular exponentiations, which is expensive). In addition, since the security proofs of our scheme are quite similar to that in [91], it is necessary to perform a comparison and show the differences between these two schemes. The results are depicted in Table 2.

Since smart cards are of scarce energy resources and limited computing capability, computation cost at user side is of major concern, and thus it deserves special attention. Generally, the authentication and key exchange phase is executed much more frequently than the registration phase, therefore only the computation cost during the former is taken into consideration. Let T_H, T_E and T_S denote the time complexity for hash function, exponential operation and symmetric cryptographic operation, respectively. Other operations, like XOR and concatenation, are negligible as compared to them. Generally, the time complexity associated with these operations can be expressed as $T_E \gg T_H \approx T_S$ [31,56]. For a better view of the computational efficiency of our scheme, in Table 3 we list the computation time for related cryptographic operations on a 32-bit RISC MIPS-based 33 MHz processor and a 64-bit Pentium(R) Dual-Core 2.80 GHz processor, respectively. Note that both our implementation and that of [76] take advantage of the publicly available cryptographic library MIRACL [75], which is a multi-precision integer, rational arithmetic C/C++ library.

As illustrated in Table 2, the proposed scheme can eliminate all the security loopholes and preserve user privacy while enjoying nearly the same performance with the other seven schemes. Remarkably, in our scheme, only two modular exponentiations plus some lightweight symmetric operations need to be conducted on the user side, and the efficiency can be further improved by using pre-computing techniques: these two exponentiations can be offline pre-computed and stored in the card memory, and thus they will be readily available as and when required. It also should be noted that, since our scheme as well as other five schemes only conducts one modular exponentiation on the server side, they all are unable to provide the property of forward secrecy, according to "the forward secrecy principle" advanced by one of our earlier works [63]. As compared to the scheme proposed by Wang et al. [91], our scheme is more efficient yet short of the feature of perfect forward secrecy. In particular, our scheme for the first time shows that provable security can be achieved by conducting only one modular exponentiation on the server side.

Remark 3. Of particular interest might be our observation that the property of user un-traceability can be achieved "*for free*", i.e. without involving additional cost in terms of communication, computation and storage. We have analyzed more than one hundred two-factor schemes and as far as we know, the schemes in [16,18,37,59] are the most efficient non-privacy-preserving ones that can resist against offline dictionary attack [94].[7] As compared to these schemes, our proposed scheme meets the same security requirements plus the goal of user anonymity without sacrificing efficiency, which highly indicates that a non-privacy-preserving scheme can be updated to a privacy-preserving one (while maintaining the same level of security) with almost no additional computation and communication cost. Yet there arises a fundamental question: Do

[7] Note that, in this paper we mainly focus on schemes that are based on the intractability of the discrete logarithm problem (and its variants like CDHP and ECDLP), which along with the integer factoring problem (IFP) are the two most widely used NP-hard problems upon which public-key cryptosystems are built.

Table 3
Computation overhead of related cryptographic operations.

Experimental platform	Exponentiation T_E ($\|n\| = 1024$)	Symmetric encryption T_S (AES-128)	Hash operation T_H (SHA-1)
Philips HiPerSmart™ 36 MHz	140 ms	4.972 ms	12.261 ms
Pentium(R) DualCore E5500 2.80 GHz	7.637 ms	0.531 µs	0.756 µs

privacy requirements and security goals rely on the same cryptographic tools? For example, Halevi and Krawczyk [35] have proved that in password based protocols, security (i.e., guarding against offline dictionary attack) can only be maintained by employing asymmetric-key techniques, what about user privacy?

7. Conclusion

In this paper, we have revisited a series of privacy-preserving two-factor authentication schemes for various settings and used such one scheme presented at ICISC 2012 as a case study, and shown that they all suffer from a serious security flaw – de-synchronization attack, serving to highlight the need for special attention to this damaging threat in the area of anonymous two-factor authentication. As our main contribution, a robust scheme is suggested to cope with the aforementioned defects and analyzed in a formal model based on reified BPR2000. The comparison results show that, our new scheme eliminates several hard security threats that are difficult to be tackled at the same time in previous scholarship, while preserving user privacy for "free" –at nearly no additional computation or communication cost. When pre-computing technique is further employed, only some lightweight symmetric-key operations are involved on the client side during the whole login process. This is appealing in mobile environments where user terminals (smart cards) have limited resources.

Our protocol is proved secure in the random oracle model and, for the first time, the notion of user un-traceability is formally defined and captured. While this formal model has been widely used in security proofs, a provably secure protocol without random oracles is certainly more desirable, which suggests a natural direction for future research. It has been conjectured that the strategy of using synchronization mechanism to achieve user un-traceability is intrinsically infeasible, but we promote the formal proof of this conjecture as one open problem. Another interesting and fundamental question about two-factor authentication protocols that remains is, whether privacy requirements and security goals rely on the same cryptographic tools (primitives)?

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive suggestions and comments that highly improve the readability and completeness of the paper, and Dr. Zhe Liu at University of Luxembourg for helpful discussions. This research was partially supported by the National Natural Science Foundation of China under Grants Nos. 61472016 and 61170282.

Appendix A. Two notes on the computation of the parameter e_A

In Section 4, we have shown that it is critical for user A to be verified by S twice. Otherwise, the proposed scheme would suffer from an offline dictionary attack and thus cannot attain truly two-factor security. In the following, we demonstrate that there are also two subtleties to be noted in the computation of the parameter e_A, overlooking either of which would render the scheme insecure against offline dictionary attacks.

Note 1. Suppose we compute $e_A = \mathcal{E}_{f_A}(y_A)$ as in Kim–Kim's scheme [50], then one can find without much difficulty that the resulting scheme is prone to offline dictionary attack. $\mathcal{M}$ can acquire A's password PW_A by performing the following attack procedure:

Step 1. Intercepts a login message $\{DID_A, e_A, y_A\}$ sent by user A.
Step 2. Computes $f_A^* = W_A \oplus h(r_A \oplus PW_A^*)$, where $\{W_A, r_A\}$ are extracted from A's card, ID_A^* and PW_A^* stand for the guessed identity and password of A.
Step 3. Verifies the correctness of ID_A^* and PW_A^* by checking $e_A \overset{?}{=} \mathcal{E}_{f_A^*}(y_A)$.
Step 4. Repeats the Steps $2 \sim 3$ until the correct value of (ID_A, PW_A) is found.

Note 2. Suppose we compute $e_A = \mathcal{E}_{f_A}(c_A)$ instead of $e_A = \mathcal{E}_{c_A}(f_A)$, then the resulting scheme is still susceptible to offline dictionary attack. $\mathcal{M}$ proceeds as follows:

Step 1. Intercepts a login message $\{DID_A, e_A, y_A\}$ sent by user A.
Step 2. Computes $f_A^* = W_A \oplus h(r_A \oplus PW_A^*)$, where $\{W_A, r_A\}$ are extracted from A's card, ID_A^* and PW_A^* stand for the guessed identity and password of A.

Step 3. Computes $c_A^* = \mathcal{D}_{f_A^*}(e_A)$, where $\mathcal{D}(\cdot)$ is the decryption algorithm corresponding to $\mathcal{E}(\cdot)$.
Step 4. Verifies the correctness of ID_A^* and PW_A^* by checking $DID_A \overset{?}{=} ID_A^* \oplus \mathcal{H}_1(y_A \| c_A^*)$.
Step 5. Repeats the Steps 2–4 until the correct value of (ID_A, PW_A) is found.

The above modifications well serve to illustrate that designing secure two-factor protocols is a delicate and challenging task, and that subtle change(s) to a protocol can render it highly vulnerable. This once again suggests the necessity of employing some kind of formal methods to establish confidence in the security of such sort of complicated cryptosystems.

References

[1] M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, in: S. Vaudenay (Ed.), PKC 2005, LNCS, vol. 3386, Springer, Berlin Heidelberg, 2005, pp. 65–84.
[2] A. Acquisti, L. Brandimarte, G. Loewenstein, Privacy and human behavior in the age of information, Science 347 (6221) (2015) 509–514.
[3] J. Balasch, B. Gierlichs, R. Verdult, L. Batina, I. Verbauwhede, Power analysis of Atmel CryptoMemory–recovering keys from secure EEPROMs, in: O. Dunkelman (Ed.), CT-RSA 2012, LNCS, vol. 7178, Springer, Berlin/Heidelberg, 2012, pp. 19–34.
[4] A. Barenghi, L. Breveglieri, I. Koren, D. Naccache, Fault injection attacks on cryptographic devices: theory, practice, and countermeasures, Proc. IEEE 100 (11) (2012) 3056–3076.
[5] S.M. Bellovin, M. Merritt, Encrypted key exchange: password-based protocols secure against dictionary attacks, in: Proceedings of IEEE S&P 1992, IEEE, 1992, pp. 72–84.
[6] L. Bilge, T. Dumitras, Before we knew it: an empirical study of zero-day attacks in the real world, in: Proceedings of ACM CCS 2012, ACM, 2012, pp. 833–844.
[7] J. Bohannon, Credit card study blows holes in anonymity, Science 347 (6221) (2015). 468–468.
[8] J. Bonneau, The science of guessing: analyzing an anonymized corpus of 70 million passwords, in: Proceedings of IEEE S&P 2012, IEEE Computer Society, 2012, pp. 538–552.
[9] J. Bonneau, M. Just, G. Matthews, Whats in a name?, in: R. Sion (Ed.), FC 2010, LNCS, vol. 6052, Springer, Berlin/Heidelberg, 2010, pp. 98–113.
[10] G. Bouffard, J.-L. Lanet, The next smart card nightmare, in: D. Naccache (Ed.), Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday-Cryptography and Security: From Theory to Applications, LNCS, vol. 6805, Springer, Berlin Heidelberg, 2012, pp. 405–424.
[11] E. Bresson, O. Chevassut, D. Pointcheval, Security proofs for an efficient password-based key exchange, in: Proceedings of ACM CCS 2003, ACM, New York, NY, USA, 2003, pp. 241–250.
[12] E. Bresson, O. Chevassut, D. Pointcheval, New security results on encrypted key exchange, in: F. Bao, R. Deng, J. Zhou (Eds.), PKC 2004, LNCS, vol. 2947, Springer-Verlag, 2004, pp. 145–158.
[13] S. Cai, Y. Li, T. Li, R.H. Deng, Attacks and improvements to an RIFD mutual authentication protocol and its extensions, in: Proceedings of ACM WiSec 2009, ACM, New York, NY, USA, 2009, pp. 51–58.
[14] C.C. Chang, T.C. Wu, Remote password authentication with smart cards, IEE Proc.-Comp. Dig. Techniq. 138 (3) (1991) 165–168.
[15] Y.-F. Chang, W.-L. Tai, H.-C. Chang, Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update, Int. J. Commun. Syst. 27 (11) (2014) 3430–3440.
[16] B. Chen, W. Kuo, L. Wuu, Robust smart-card-based remote user password authentication scheme, Int. J. Commun. Syst. 27 (2) (2014) 377–389.
[17] T.H. Chen, H.C. Hsiang, W.K. Shih, Security enhancement on an improvement on two remote user authentication schemes using smart cards, Fut. Gener. Comp. Syst. 27 (4) (2011) 377–380.
[18] H. Chung, W. Ku, M. Tsaur, Weaknesses and improvement of Wang et al.'s remote user password authentication scheme for resource-limited environments, Comput. Stand. Inter. 31 (4) (2009) 863–868.
[19] L. Constantin, Sony Stresses that PSN Passwords were Hashed <http://news.softpedia.com/news/Sony-Stresses-PSN-Passwords-Were-Hashed-198218.shtml> (May 2011).
[20] J.-S. Coron, J. Patarin, Y. Seurin, The random oracle model and the ideal cipher model are equivalent, in: D. Wagner (Ed.), CRYPTO 2008, LNCS, vol. 5157, Springer, Berlin/Heidelberg, 2008, pp. 1–20.
[21] A. Das, J. Bonneau, M. Caesar, N. Borisov, X. Wang, The tangled web of password reuse, in: Proceedings of NDSS 2014, The Internet Society, 2014, pp. 1–15.
[22] M. Das, A. Saxena, V. Gulati, A dynamic id-based remote user authentication scheme, IEEE Trans. Consum. Electron. 50 (2) (2004) 629–631.
[23] M.L. Das, Two-factor user authentication in wireless sensor networks, IEEE Trans. Wirel. Commun. 8 (3) (2009) 1086–1090.
[24] Dazzlepod Inc., CSDN Cleartext Passwords, Online News <http://dazzlepod.com/csdn/> (March 2013).
[25] M. Dell'Amico, P. Michiardi, Y. Roudier, Password strength: an empirical analysis, in: Proceedings of INFOCOM 2010, IEEE, 2010, pp. 1–9.
[26] S. Drimer, S.J. Murdoch, R. Anderson, Thinking inside the box: system-level failures of tamper proofing, in: Proceedings of 2008 IEEE Security & Privacy, IEEE, 2008, pp. 281–295.
[27] EMVCo, LLC., Integrated Circuit Card Specifications for Payment Systems: Europay, Mastercard and Visa <http://www.emvco.com/specifications.aspx?id=223> (November 2011).
[28] N. Ernstmann, O. Ommen, M. Neumann, A. Hammer, R. Voltz, H. Pfaff, Primary care physician's attitude towards the german e-health card projectłdeterminants and implications, J. Med. Syst. 33 (3) (2009) 181–188.
[29] ETSI-TS-102: Smart Cards; UICC-Terminal Interface; Physical and Logical Characteristics <http://www.etsi.org/standards> (February 2010).
[30] C. Fan, Y. Chan, Z. Zhang, Robust remote authentication scheme with smart cards, Comp. Sec. 24 (8) (2005) 619–628.
[31] N. Ferguson, B. Schneier, T. Kohno, Cryptography Engineering: Design Principles and Practical Applications, John Wiley & Sons, 2010.
[32] D. Florencio, C. Herley, A large-scale study of web password habits, in: Proceedings of WWW 2007, ACM, New York, NY, USA, 2007, pp. 657–666.
[33] E. Fujisaki, T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, in: M. Wiener (Ed.), Proceedings of CRYPTO 1999, LNCS, vol. 1666, Springer, Verlag, 1999, pp. 537–554.
[34] A.R.A. Grégio, D.S. Fernandes, V.M. Afonso, P.L. de Geus, V.F. Martins, M. Jino, An empirical analysis of malicious internet banking software behavior, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC 2013), ACM, 2013, pp. 1830–1835.
[35] S. Halevi, H. Krawczyk, Public-key cryptography and password protocols, ACM Trans. Inf. Syst. Sec. 2 (3) (1999) 230–268.
[36] D. He, N. Kumar, N. Chilamkurti, A secure temporal–credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks, Inf. Sci. 321 (2015) 263–277.
[37] D.B. He, J.H. Chen, J. Hu, Improvement on a smart card based password authentication scheme, J. Internet Technol. 13 (3) (2012) 38–42.
[38] D.J. He, C. Chen, J. Bu, S. Chan, Y. Zhang, Security and efficiency in roaming services for wireless networks: challenges, approaches, and prospects, IEEE Commun. Magaz. 51 (2) (2013) 142–150.
[39] D.J. He, M.D. Ma, Y. Zhang, C. Chen, J.J. Bu, A strong user authentication scheme with smart cards for wireless communications, Comput. Commun. 34 (3) (2011) 367–374.
[40] X. Huang, X. Chen, J. Li, Y. Xiang, L. Xu, Further observations on smart-card-based password-authenticated key agreement in distributed systems, IEEE Trans. Parallel Distrib. Syst. 25 (7) (2014) 1767–1775.
[41] D. Hughes, V. Shmatikov, Information hiding, anonymity and privacy: a modular approach, J. Comp. Sec. 12 (1) (2004) 3–36.

[42] J.Y. Hwang, S. Lee, B.-H. Chung, H.S. Cho, D. Nyang, Group signatures with controllable linkability for dynamic membership, Inf. Sci. 222 (2013) 761–778.

[43] Q. Jiang, J. Ma, G. Li, X. Li, Improvement of robust smart-card-based password authentication scheme, Int. J. Commun. Syst. 28 (2) (2015) 383–393.

[44] Q. Jiang, J. Ma, Z. Ma, G. Li, A privacy enhanced authentication scheme for telecare medical information systems, J. Med. Syst. 37 (1) (2013) 1–8.

[45] Q. Jiang, Z. Ma, J.F. Ma, G. Li, Security enhancement of robust user authentication framework for wireless sensor networks, China Commun. 9 (10) (2012) 103–111.

[46] W.-S. Juang, S.-T. Chen, H.-T. Liaw, Robust and efficient password-authenticated key agreement using smart cards, IEEE Trans. Ind. Electron. 55 (6) (2008) 2551–2556.

[47] J. Katz, P. MacKenzie, G. Taban, V. Gligor, Two-server password-only authenticated key exchange, J. Comp. Syst. Sci. 78 (2) (2012) 651–669.

[48] J. Katz, R. Ostrovsky, M. Yung, Efficient and secure authenticated key exchange using weak passwords, J. ACM 57 (1) (2009) 1–41.

[49] M. Khan, S. Kim, K. Alghathbar, Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme', Comp. Commun. 34 (3) (2011) 305–309.

[50] K.-K. Kim, M.-H. Kim, An enhanced anonymous authentication and key exchange scheme using smartcard, in: A. Juels, C. Pa ar (Eds.), Proceedings of the 15th International Conference on Information Security and Cryptology (ICISC 2012), LNCS, vol. 7839, Springer, Berlin/Heidelberg, 2012, pp. 487–494.

[51] H. Krawczyk, HMQV: a high-performance secure Diffie–Hellman protocol, in: V. Shoup (Ed.), CRYPTO 2005, LNCS, vol. 3621, Springer, Berlin/Heidelberg, 2005, pp. 546–566.

[52] S. Kumari, M.K. Khan, Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme, Int. J. Commun. Syst. 27 (12) (2014) 3939–3955.

[53] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, X. Shen, CPAL: a conditional privacy-preserving authentication with access linkability for roaming service, IEEE Internet Things J. 1 (1) (2014) 46–57.

[54] X. Leroy, Smart Card Security From a Programming Language and Static Analysis Perspective, INRIA Rocquencourt & Trusted Logic, Tecnical Report, 2013 <http://pauillac.inria.fr/~xleroy/talks/language-security-etaps03.pdf.

[55] C.-T. Li, A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card, IET Inf. Sec. 7 (1) (2013) 3–10.

[56] C.-T. Li, M.-S. Hwang, A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks, Inf. Sci. 181 (23) (2011) 5333–5347.

[57] C.T. Li, C.C. Lee, A robust remote user authentication scheme using smart card, Inf. Technol. Control 40 (3) (2011) 236–245.

[58] T.-Y. Li, G.-L. Wang, Security analysis of two ultra-lightweight RFID authentication protocols, in: H. Venter, M. Eloff, L. Labuschagne, J. Eloff, R. Solms (Eds.), Proceedings of SEC 2007, IFIP AICT, vol. 232, Springer-Verlag, 2007, pp. 109–120.

[59] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, J. Netw. Comput. Appl. 36 (5) (2013) 1365–1371.

[60] X. Li, Y. Xiong, J. Ma, W. Wang, An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards, J. Netw. Comput. Appl. 35 (2) (2012) 763–769.

[61] X.X. Li, W.D. Qiu, D. Zheng, K.F. Chen, J. Li, Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards, IEEE Trans. Ind. Electron. 57 (2) (2010) 793–800.

[62] J. Long, No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing, Syngress, 2011.

[63] C.-G. Ma, D. Wang, S. Zhao, Security flaws in two improved remote user authentication schemes using smart cards, Int. J. Commun. Syst. 27 (10) (2014) 2215–2227.

[64] R. Madhusudhan, R. Mittal, Dynamic id-based remote user password authentication schemes using smart cards: a review, J. Netw. Comput. Appl. 35 (4) (2012) 1235–1248.

[65] K. Mangipudi, R. Katti, A secure identification and key agreement protocol with user anonymity (sika), Comp. Sec. 25 (6) (2006) 420–425.

[66] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comp. 51 (5) (2002) 541–552.

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

[68] E. Morse, M. Theofanos, Y. Choong, C. Paul, A. Zhang, FIPS 201: Personal Identity Verification of Federal Employees and Contractors, Tech. rep., National Institute of Standards and Technology, McLean, VA, August 2013, http://dx.doi.org/10.6028/NIST/.FIPS.201-2.

[69] S.J. Murdoch, S. Drimer, R. Anderson, M. Bond, Chip and pin is broken, in: Proceedings of IEEE S&P 2010, IEEE Computer Society, 2010, pp. 433–446.

[70] K. Nohl, D. Evans, S. Starbug, H. Plötz, Reverse-engineering a cryptographic RFID tag, in: USENIX Security 2008, USENIX Association, 2008, pp. 185–193.

[71] A. Patrice, B.B. Macq, Feature-based watermarking of 3d objects: toward robustness against remeshing and desynchronization, in: Proceedings of SPIE, vol. 5681, 2005, pp. 400–408.

[72] B. Pinkas, T. Sander, Securing passwords against dictionary attacks, in: Proceedings of ACM CCS 2002, ACM, 2002, pp. 161–170.

[73] D. Pointcheval, Provable security for public key schemes, in: Contemporary Cryptology, Springer, 2005, pp. 133–190.

[74] D. Pointcheval, Password-based authenticated key exchange, in: M. Fischlin, J. Buchmann, M. Manulis (Eds.), PKC 2012, LNCS, vol. 7293, Springer, Berlin/Heidelberg, 2012, pp. 390–397.

[75] M. Scott, Miracl Library, CertiVox UK Ltd., 2011 <https://www.certivox.com/miracl>.

[76] M. Scott, N. Costigan, W. Abdulwahab, Implementing cryptographic pairings on smartcards, in: L. Goubin, M. Matsui (Eds.), CHES 2006, LNCS, vol. 4249, Springer, Berlin Heidelberg, 2006, pp. 134–147.

[77] K. Shim, Security flaws in three password-based remote user authentication schemes with smart cards, Cryptologia 36 (1) (2012) 62–69.

[78] H. Shirazi, J. Cosmas, D. Cutts, A cooperative cellular and broadcast conditional access system for pay-tv systems, IEEE Trans. Multim. 56 (1) (2010) 44–57.

[79] R. Song, Advanced smart card based password authentication protocol, Comp. Stand. Interf. 32 (5) (2010) 321–325.

[80] H.-M. Sun, B.-Z. He, C.-M. Chen, T.-Y. Wu, C.-H. Lin, H. Wang, A provable authenticated group key agreement protocol for mobile environment, Inf. Sci. 321 (2015) 224–237.

[81] J. Sun, C. Zhang, Y. Zhang, Y. Fang, Sat: a security architecture achieving anonymity and traceability in wireless mesh networks, IEEE Trans. Depend. Secur. Comput. 8 (2) (2011) 295–307.

[82] J. Tsai, T. Wu, K. Tsai, New dynamic id authentication scheme using smart cards, Int. J. Commun. Syst. 23 (12) (2010) 1449–1462.

[83] J.-L. Tsai, N.-W. Lo, T.-C. Wu, Novel anonymous authentication scheme using smart cards, IEEE Trans. Ind. Inform. 9 (4) (2013) 2004–2013.

[84] W.-J. Tsaur, J.-H. Li, W.-B. Lee, An efficient and secure multi-server authentication scheme with key agreement, J. Syst. Softw. 85 (4) (2012) 876–882.

[85] A.G. Vicente, I.B. Munoz, J.L.L. Galilea, P.A.R. del Toro, Remote automation laboratory using a cluster of virtual machines, IEEE Trans. Ind. Electron. 57 (10) (2010) 3276–3283.

[86] C.-H. Wang, S. Chin, A new RFID authentication protocol with ownership transfer in an insecure communication environment, Proceedings of the Ninth International Conference on Hybrid Intelligent Systems (HIS 2009), vol. 1, IEEE, 2009, pp. 486–491.

[87] D. Wang, D. He, P. Wang, C.-H. Chu, Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment, IEEE Trans. Depend. Secur. Comput. (2014). http://dx.doi.org/10.1109/TDSC.2014.2355850.

[88] D. Wang, P. Wang, Offline dictionary attack on password authentication schemes using smart cards, in: Y. Desmedt, B. Thuraisingham, K. Hamlen (Eds.), ISC 2013, LNCS, Springer, Berlin/Heidelberg, 2013, pp. 1–16. <http://eprint.iacr.org/2014/208.pdf>.

[89] D. Wang, P. Wang, On the usability of two-factor authentication, in: Proceedings of SecureComm 2014, Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, 2015, pp. 1–9.

[90] D. Wang, P. Wang, J. Liu, Improved privacy-preserving authentication scheme for roaming service in mobile networks, in: Proceedings of 15th IEEE Wireless Communications and Networking Conference (WCNC 2014), IEE, 2014, pp. 3178–3183.

[91] D. Wang, P. Wang, C.G. Ma, Z. Chen, Robust Smart-Card-Based Password Authentication Scheme Against Smart Card Security Breach, Cryptology ePrint Archive, Report 2012/439, 2012 <http://eprint.iacr.org/2012/439.pdf>.

[92] R.C. Wang, W.S. Juang, C.L. Lei, Robust authentication and key agreement scheme preserving the privacy of secret key, Comp. Commun. 34 (3) (2011) 274–280.

[93] Y. Wang, J. Liu, F. Xiao, J. Dan, A more efficient and secure dynamic id-based remote user authentication scheme, Comp. Commun. 32 (4) (2009) 583–585.

[94] Y.G. Wang, Password protected smart card and memory stick authentication against off-line dictionary attacks, in: D. Gritzalis, S. Furnell, M. Theoharidou (Eds.), SEC 2012, IFIP AICT, vol. 376, Springer, Boston, 2012, pp. 489–500.

[95] F. Wen, A more secure anonymous user authentication scheme for the integrated epr information system, J. Med. Syst. 38 (5) (2014) 1–7.

[96] F. Wen, X. Li, An improved dynamic id-based remote user authentication with key agreement scheme, Comp. Electr. Eng. 38 (2) (2012) 381–387.

[97] F. Wen, W. Susilo, G. Yang, A secure and effective anonymous user authentication scheme for roaming service in global mobility networks, Wirel. Pers. Commun. 73 (3) (2013) 993–1004.

[98] S.H. Wu, Y.F. Zhu, Q. Pu, Robust smart-cards-based user authentication scheme with user anonymity, Secur. Commun. Netw. 5 (2) (2012) 236–248.

[99] T. Xiang, K. Wong, X. Liao, Cryptanalysis of a password authentication scheme over insecure networks, J. Comp. Syst. Sci. 74 (5) (2008) 657–661.

[100] Q. Xie, Dynamic id-based password authentication protocol with strong security against smart card lost attacks, in: P. Snac, M. Ott, A. Seneviratne, O. Akan (Eds.), Proceedings of ICWCA 2012, LNICST, vol. 72, Springer, Berlin/Heidelberg, 2012, pp. 412–418.

[101] H. Xiong, Y. Chen, Z. Guan, Z. Chen, Finding and fixing vulnerabilities in several three-party password authenticated key exchange protocols without server public keys, Inf. Sci. 235 (2013) 329–340.

[102] J. Xu, W.-T. Zhu, A generic framework for anonymous authentication in mobile networks, J. Comp. Sci. Technol. 28 (4) (2013) 732–742.

[103] J. Xu, W.-T. Zhu, D.-G. Feng, An efficient mutual authentication and key agreement protocol preserving user anonymity in mobile networks, Comp. Commun. 34 (3) (2011) 319–325.

[104] K. Xue, P. Hong, C. Ma, A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture, J. Comp. Syst. Sci. 80 (1) (2014) 195–206.

[105] Q. Yan, J. Han, Y. Li, R.H. Deng, On limitations of designing leakage-resilient password systems: attacks, principles and usability, in: Proceedings of NDSS 2012, February 5–8, The Internet Society, San Diego, USA, 2012, pp. 1–16.

[106] G. Yang, D. Wong, H. Wang, X. Deng, Two-factor mutual authentication based on smart cards and passwords, J. Comp. Syst. Sci. 74 (7) (2008) 1160–1172.

[107] M.H. Yang, Across-authority lightweight ownership transfer protocol, Electron. Comm. Res. Appl. 10 (4) (2011) 375–383.

[108] K.H. Yeh, C. Su, N.W. Lo, Y. Li, Y.X. Hung, Two robust remote user authentication protocols using smart cards, J. Syst. Softw. 83 (12) (2010) 2556–2565.

[109] X. Yi, S. Ling, H.-X. Wang, Efficient two-server password-only authenticated key exchange, IEEE Trans. Paral. Distrib. Syst. 24 (9) (2013) 1773–1782.

[110] Y. Zhang, J. Chen, B. Huang, C. Peng, An efficient password authentication scheme using smart card based on elliptic curve cryptography, Inf. Technol. Control 43 (4) (2014) 390–401.

[111] T. Zhou, J. Xu, Provable secure authentication protocol with anonymity for roaming service in global mobility networks, Comp. Netw. 55 (1) (2011) 205–213.