Robust Smart Card based Password Authentication Scheme against Smart Card Security Breach*

Ding Wang[1,2**], Ping Wang[2], Chun-guang Ma[3], and Zhong Chen[1]

[1] College of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[2] National Engineering Research Center for Software Engineering, Beijing 100871, China
[3] College of Computer Science, Harbin Engineering University, Harbin 150001, China
wangdingg@mail.nankai.edu.cn

Abstract. As the most prevailing two-factor authentication mechanism, smart card based password authentication has been a subject of intensive research in the past decade and hundreds of this type of schemes have been proposed. However, most of them were found severely flawed, especially prone to the smart card security breach problem, shortly after they were first put forward, no matter the security is heuristically analyzed or formally proved. In SEC'12, Wang pointed out that, the main cause of this issue is attributed to the lack of an appropriate security model to fully identify the practical threats. To address the issue, Wang presented three kinds of security models, namely Type I, II and III, and further proposed four concrete schemes, only two of which, i.e. PSCAV and PSCAb, are claimed to be secure under the Type III model, i.e. the harshest security model. However, in this paper, we demonstrate that PSCAV still cannot achieve the claimed security goals and is vulnerable to an offline password guessing attack and other attacks in the Type III security mode, while PSCAb has several practical pitfalls. As our main contribution, a robust scheme is presented to cope with the aforementioned defects and it is proven to be secure in the random oracle model. Moreover, the analysis demonstrates that our scheme meets all the proposed criteria and eliminates several hard security threats that are difficult to be tackled at the same time in previous scholarship, which highly indicates the settlement of an open problem raised by Madhusudhan and Mittal in 2012. Beyond our cryptanalysis of current schemes and our proposal of the new scheme, the proposed adversary model and criteria set provide a benchmark for the systematic evaluation of future two-factor authentication proposals.

Keywords: Cryptanalysis, Authentication protocol, Smart card, Non-tamper resistant, Dynamic ID, Offline password guessing attack.

Revision. In http://eprint.iacr.org/2012/527 [66], Prof. M. Scott conducted a cryptanalysis of the original scheme and mainly four points are presented: 1) a determined adversary should be allowed to somehow learn a client's identity, which is not explicitly stated in our original adversary model; 2) an offline password guessing attack can be successfully launched under the assumption that a "powerful" attacker is equipped with the ability of breaching a legitimate user's smart card and the ability of determining this user's identity; 3) "If password derived data should not be stored on the server, and if identities are to be used as a kind of extra

* This is a revised version in response to [66], Sep 25, 2012. In this version, we correct the flaw in our scheme and have sent the paper to Prof. Michael Scott for a review before making it public, and we sincerely thank him for his insightful observations and constructive comments on improving this study. Since then, only slightly modifications are made to Section 2: Adversary model and evaluation criteria, while the other parts remain unchanged.

** Part of this work was done while the first author was in Harbin Engineering University.

password, then identity-derived information should also not be stored by the server, and this should form one of the desirable attributes of such a scheme."; 4) "the use of identities as a kind of surrogate password is not a viable strategy for the development of such schemes. "

We are grateful to Scott for his insightful observations, and acknowledge that the first two points are correct and invaluable. As we shall see later, in the extended adversary model, a determined adversary is explicitly allowed to learn a client's identity and the enhanced scheme is still provably secure in the random oracle model provided that the CDH problem is intractable. The remedy is rather simple: to thwart Scott's attack, the key parameter $k = \mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ is included into the calculation of CID_i in the login phase, which is the only essential difference as compared to the original scheme.

However, the latter two points are not so reasonable yet, maybe due to the ambiguities in our original paper. We think a misunderstanding occurs here. In the original paper, we did agree with the idea that the security of a scheme shall only rely on the secrecy of the password and the possession of the smart card, which can be confirmed by our affirmation that:

> *" Yang et al.'s formal adversary model does capture the exact two-factor authentication: only with both the smart card and the correct password can a user carry out the smart-card-based password authentication scheme successfully with the remote authentication server."*

In the original paper, we never stated that user's identity should be "considered as a kind of surrogate extra password",[0] and actually we favored the opposite. The point here is, it is our implicit assumption that an attacker who is equipped with the capability of C-2(ii) shall not be able to learn the victim user's identity that leads to Scott's successful cryptanalysis, but not the strategy that "user's identity could be considered as a kind of surrogate extra password" that does. In this regard, the preconditions for Point 3 and Point 4 don't hold up and thus these two points may not be necessary, although they are no doubt correct. Nevertheless, they do underline the following consensus: it is more reasonable and practical to not consider user identity as a kind of surrogate extra password. In a word, the attribute that identity-derived information should also not be stored by the server is meaningless and thus it is unreasonable to incorporate it into the criteria set.

It is also worth noting that: 1) user's identity directly relates to user privacy and it is desirable to have it well protected from eavesdropping attackers (passive attackers), i.e. to provide user anonymity, which means the provision of identity protection and user un-traceability against an eavesdropping attacker; 2) although the security of a protocol should not rely on the secrecy of user's identity, preserving user anonymity does help to safeguard protocol security in reality. Notion 1 is obvious, while notion 2 is seemingly quite paradoxical. When we evaluate the security of a protocol, we always assume an extremely powerful adversary with all the reasonable capabilities allowed to her (except for the ones or the combination of the ones that enable her to trivially break any of this type of protocols); but in specific scenarios, if user anonymity is preserved, an eavesdropping attacker is kept away from user's identity and an active attacker still needs to make some efforts (e.g., by shoulder-surfing or gaining temporary access to the smart card) to learn user's identity, because it can no longer be trivially learnt by wiretapping, while the efforts to learn a user's identity is often more costly (difficult) than by wiretapping. From this point of view, it is appropriate to say that preserving user anonymity does safeguard (increase) protocol security. This explains the paradox.

[0] Note that, in our earlier works, e.g. [51, 53, 76], the security of the protocols does rely on the secrecy of user's identity.

1 Introduction

Password authentication with smart card is one of the most convenient and effective two-factor authentication mechanisms for remote systems to assure one communicating party of the legitimacy of the corresponding party by acquisition of corroborative evidence. This technique has been widely deployed for various kinds of authentication applications, such as remote host login, online banking, e-commerence and e-health [14]. In addition, it constitutes the basis of three-factor authentication [28]. However, there still exists challenges in both security and performance aspects due to the stringent security requirements and resource-strained characteristics of the clients.

Since Chang and Wu [10] introduced the first remote user authentication scheme using smart cards in 1993, there have been many of such schemes proposed [15, 17, 33, 44, 46, 51, 69, 81, 83, 84, 87, 92]. One prominent issue in this type of schemes is security against offline guessing attack, which is the severest threat that a sound and practical scheme must be able to thwart. Traditionally, to prevent an adversary from launching offline guessing attack, one need to make sure that the scheme is not going to leak any information useful about the client's password to the adversary in the protocol run, even though the password is considered to be weak and low-entropy. By observing this, many schemes employed some techniques similar to Bellovin and Merritt's Encrypted Key Exchange protocol. A common feature of such schemes is that the smart card is assumed to be tamper-resistant, i.e., the secret parameters stored in the smart card cannot be revealed. However, recent research results have demonstrated that the secret data stored in the smart card could be extracted by some means, such as monitoring the power consumption [5, 37, 39, 58] or analyzing the leaked information [32, 55]. Therefore, such schemes [15,33,49,83] based on the tamper resistance assumption of the smart card are vulnerable to offline password guessing attacks, user impersonation attack, etc, once an adversary has obtained the secret data stored in a user's smart card and/or just some intermediate computational results in the smart card [13,44,75,85,89]. Consequently, a stronger notion of security against offline guessing attack is developed to require that compromising a client's smart card should do not help the adversary launch offline guessing attack against the client's password.

In 2008, seeing that most of the previous schemes have been found inconsistent with this strong notion of security due to the lack of formal security analysis, Yang et al. [93] presented a formal adversarial model for analyzing the security of this type of schemes and a generic construction framework with a new scheme to demonstrate its effectiveness. Although Yang *et al.*'s scheme suffers from some security vulnerabilities like smart card loss attack (as shown in Appendix A) and privileged insider attack, and fails to provide many desirable features such as repairability and user anonymity, Yang et al.'s formal adversary model does capture the exact two-factor authentication of smart-card-based password authentication schemes: only with both the smart card and the correct password can a user carry out the smart-card-based password authentication scheme successfully with the remote authentication server.

Following Yang et al.'s seminal work [93], many enhanced schemes [25,44,46,62,68,69,73,80, 87,90,91,94] have been proposed to address the smart card security breach problem, however, most of them were shortly found having various security weaknesses being overlooked [24, 51, 64,67,76,77,87]. Remarkably, some of them, like [62,73,80,81,91,94], even have been provided with a formal proof. The past thirty years of research in the area of password-authenticated key exchange (PAKE) has proved that it is incredibly difficult to get even a single factor based authentication scheme right [8, 96], while the past decade of research in smart card based password authentication has proved that designing a secure and practical two-factor authentication protocol can only be harder. For a typical example, in 2012, Hsieh and Leu

[27] demonstrated several attacks against Hsiang-Shih's scheme [26] and further proposed an enhanced version by "exploiting hash functions". They claimed that the improved scheme can withstand offline password guessing attack even if the sensitive parameters are extracted by the adversary. However, we found Hsieh-Leu's scheme still cannot achieve its claimed main security goal by demonstrating an offline password guessing attack in Appendix B. Through the security analysis of Hsieh-Leu's scheme and Yang *et al.*'s scheme, some subtleties and challenges in designing this type of schemes, different from the traditional password-based (i.e., one-factor) authentication, are uncovered.

In SEC'12, Wang [84] observed that the previous papers in this area present attacks on protocols in previous papers and propose new protocols without proper security justification (or even a security model to fully identify the practical threats), which contributes to the main cause of the above failure. Accordingly, Wang presented three kinds of security models, namely Type I, II and III, and further proposed four concrete schemes, only two of which, i.e. PSCAb and PSCAV, are claimed to be secure under the harshest model, i.e. Type III security model. The type III model will be reviewed later in Section 2. However, PSCAb requires Weil or Tate pairing operations to defend against offline guessing attack and may not be suitable for systems where pairing operations are considered to be too expensive or infeasible to implement. Moreover, PSCAb suffers from the well-known key escrow problem and lacks some desirable features such as local password update, repairability and user anonymity. As for PSCAV, in Appendix C, we will demonstrate that it still cannot achieve the claimed security goals and is vulnerable to an offline password guessing attack and other attacks under the Type III security model. As our main contribution, a robust and efficient scheme is presented to cope with the identified defects and it is formally proven to be secure in the Type III model.

The remainder of this paper is organized as follows: in Section 2, we elaborate on the adversary model and evaluation criteria. Our proposed scheme is presented in Section 3, and its security analysis is given in Section 4. The comparison of the performance of our scheme with the other related schemes is shown in Section 5. Section 6 concludes the paper. Yang *et al.*'s scheme [93] is investigated in Appendix A. An offline password guessing attack against the scheme proposed by Hsieh-Leu [27] is uncovered in Appendix B. The demonstration of offline password guessing attack on Wang's PSCAV protocol [84] is given in Appendix C.

2 Adversary model and evaluation criteria

There have been several papers dealing with smart-card-based password authentication schemes in recent years (see, e.g., [17, 25, 33, 47, 48, 73, 83, 90]). However, in most of these studies, the authors present attacks on previous schemes and propose new protocols with assertions of the superior aspects of their schemes, while ignoring benefits that their scheme doesn't attempt (or fail) to provide, thus overlooking dimensions on which it fares poorly. Despite the lack of evaluation criteria, another common feature of these studies is that, there is no proper security justification (or even an explicit security model) presented, which explains why these protocols previously claimed to be secure turn out to be vulnerable. The research history of this area can be summarized in the following diagram (For a more telling picture, see Fig.1):

$$\text{New protocol} \rightarrow \text{broken} \rightarrow \text{improved protocol} \rightarrow \text{broken again} \rightarrow \text{further improved protocol} \rightarrow \cdots$$

which generates a lot of literature, yet as far as we know, little attention has been paid to the systematic design and analysis of this sort of schemes. Accordingly, in the following, an adversary model consistent with the reality is explicitly defined and a comprehensive criteria set is proposed.

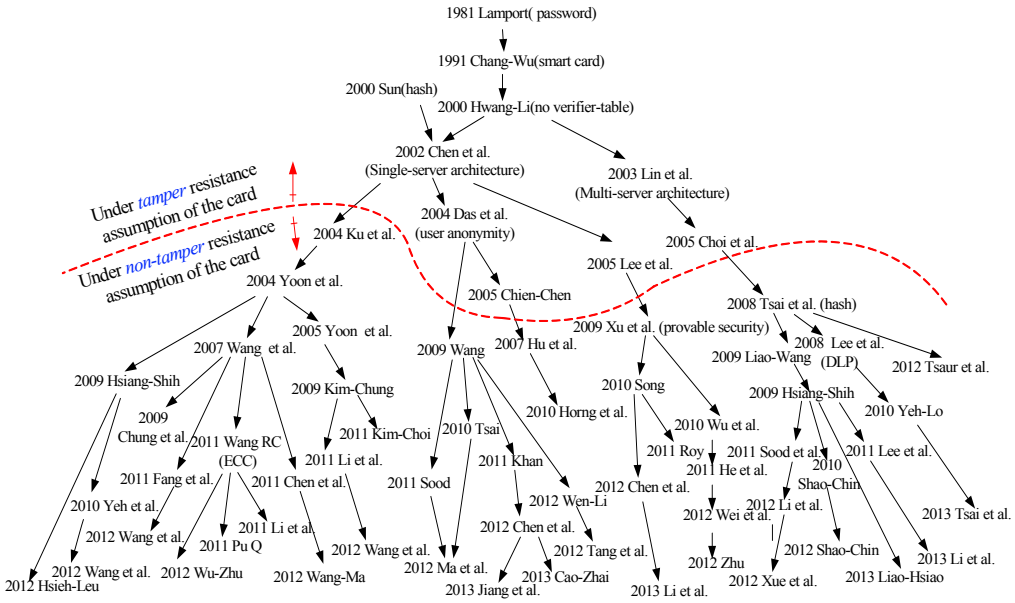

Fig. 1. The history of "break-fix-break-fix"

2.1 Adversary model

In the conventional password authenticated key exchange protocols, the attacker is modeled to have full control of the communication channel between the communicating parties [3], such as eavesdropping, intercepting, inserting, deleting, and modifying any transmitted messages over the public channel. Though this assumption is reasonable for password-based authentication situations, it is not sufficient for password-based remote authentication using smart cards.

Recent studies have reported that, the secret parameters stored in common smart cards could be extracted (or partially extracted) by power analysis attacks [37, 58, 74], the software loophole exploiting attacks (launched on software-supported card, e.g., Java Card) [43] or reverse engineering techniques [60]. Consequently, the leakage of sensitive parameters stored in the smart card may lead the original secure schemes vulnerable to the smart card loss problem, such as offline password guessing attack and impersonation attack [41,52,54,68,77,91]. Accordingly, it is more prudent and desirable to design password authentication schemes using smart cards under the assumption that the secret keys stored in the smart card could be revealed by some means. What's more, as observed and in-deep investigated by Wang [84] quite recently, malicious card readers also contribute to the security failures of such schemes. Once the card reader is under the control of the attacker, e.g. the card reader is infected with viruses and/or Trojans, the card owner's input password may be intercepted.

However, we restrict the attacker from first intercepting the password via the card reader and then reading the information stored in the card via the stolen (or lost) smart card, otherwise this combination will enable the attacker to trivially break any two-factor authentication protocols. This treatment adheres to "the extreme-adversary principle" [23]: Robust security is to protect against an extremely powerful adversary, of whom the only restricted powers are those that would allow her to trivially break any this type of schemes. Moreover, this treatment is reasonable in reality: (1) the user is at the scene when she inserts her card into a malicious terminal, and there is little chance for the attacker to launch side-channel attacks (which needs

special instruments and attack platforms); (2) the attacker is unlikely to succeed in revealing the sensitive data on the card within a short period of time. However, if a memory USB stick is used in such an un-trusted terminal, both the parameters stored in the memory and the user's password will be exposed easily and without any abnormality. This well explains the essential advantage of using smart cards over employing common memory sticks, even if non-tamper resistance assumption of the smart cards are made.

In reality, previous session key(s) or materials used to construct the session key may be lost for a variety of reasons [40], ranging from the malicious action of an insider to a temporary break-in into a computer system or the prescribed-release of that session key when the session is torn down. Adding this ability to $\mathcal{A}$ allows our model to capture the threat of the known key attack. To evaluate the damage of leakage of server's long-term private key, the capability of learning server's long-time private key is equipped with our adversary. This allows us to deal with forward secrecy.

Furthermore, it is worth noting that, in remote user authentication schemes, for the sake of user-friendliness, a user is often allowed to select her own identity ID at will (maybe confined to a predefined format) during the registration phase; the user usually tends to choose an identity which is easily remembered for her convenience. Consequently, these easy-to-remember identities are of low entropy and thus can also be offline enumerated by an adversary $\mathcal{A}$ within polynomial time in the same way with the passwords. Hence, in practice, it is reasonable and realistic to assume that $\mathcal{A}$ can offline enumerate all the (ID, PW) pairs in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$ within polynomial time. In contrast, most of the proposed dynamic-ID schemes (i.e. user's identity is concealed in session-variant pseudo-identities to provide the property of user anonymity), e.g. [76, 80, 87], explicitly assume $\mathcal{A}$ cannot guess both ID and PW correctly at the same time. In other words, such dynamic-ID schemes may be vulnerable to offline password guessing attack under our assumption.

Last but not the least, it is reasonable and practical to assume that a determined adversary may somehow learn this victim user's identity. Firstly, user's identity is static and often confined to a predefined format, and it is more easily guessed than the password [7]. Secondly, in practice, imprudent users tend to write their identities directly on the card, and after all, the attacker can know more or less about the personal information of the card holder when she gets access to the card. Lastly, the input identity is usually displayed in plain on the screen and is susceptible to shoulder-surfing.

Table 1. Capabilities of the adversary

C-00	The adversary $\mathcal{A}$ can offline enumerate all the elements in the Cartesian product $\mathcal{D}_{id} \times \mathcal{D}_{pw}$ within a reasonable amount of time, where $\mathcal{D}_{pw}$ and $\mathcal{D}_{id}$ denote the password space and the identity space, respectively.
C-01	The (active) adversary $\mathcal{A}$ has the capability of determining the victim's identity.
C-1	The adversary $\mathcal{A}$ has full control of the communication channel between the communicating parties, such as eavesdropping, intercepting, inserting, deleting, and modifying any transmitted messages over the public channel.
C-2	The adversary $\mathcal{A}$ may either (i) learn the password of a victim via malicious card reader, or (ii) extract the secret data in the lost smart card by side-channel attacks, but cannot achieve both. Otherwise, it is a trivial case.
C-3	The adversary $\mathcal{A}$ can learn the previous session key(s).
C-4	The adversary $\mathcal{A}$ has the capability of learning server's long-time private key(s) only when evaluating the eventual failure of the server.

The capabilities of the adversary in our model are summarized in Table 1. As far as we know, our work, following Li-Lee's [44] and Wang's [84] work, is one of the few ones that explicitly specify the capabilities of the adversary. In [84], Wang presented three kinds of security models, namely Type I, II and III, and further proposed four concrete schemes, only two of which are claimed to be secure under the harshest model, i.e. Type III security model. In Wang's Type III model, three assumptions are made:

(1) an adversary $\mathcal{A}$ is allowed to have full control of the communication channel between the user and the server, which is consistent with C-1;

(2) the smart card is assumed to be non-tamper resistant and the user's password may be intercepted by $\mathcal{A}$ using a malicious smart card reader, but not both, which is consistent with C-2;

(3) there is no counter protection in the smart card, i.e. $\mathcal{A}$ can issue a large amount of queries to the smart card using a malicious card reader to learn some useful information.

With regard to assumption 3, we argue that this assumption may not be of much practical significance, because whether assumption 3 is valid or not in practice has little relevance with protocol security under assumption 2. On the one hand, if there is no verification of the input password before the function (the run mode) of the smart card, the only way that $\mathcal{A}$ can learn some useful information (except the static data stored in the card, which can be learnt by $\mathcal{A}$ under assumption 2) is to interact with the remote server, which can be effectively thwarted by the server, e.g., locking the corresponding user account after a few failed login attempts. On the other hand, if this verification exists, $\mathcal{A}$ can always find the password that passes the verification by exhaustively inputting her guessing passwords into the malicious card reader(and with assumption 2, secret data stored in the card can also be extracted out), which is explicitly not allowed in the Type III model. Hence, assumption 3 is not incorporated into our model. As with some other studies [76, 84], we may simply assume that there is counter protection in the smart card, i.e., the smart card will be self-destroyed or locked for a time period if the query number exceeds a certain threshold (e.g., the GSM SIM card V2 or later has this capability).

According to the above analysis, our model is in much similarity with the Type III model (the most powerful one) introduced in [84], and the major difference is that $\mathcal{A}$ in Type III model is not provided with the capabilities of C-3 and C-4. Hence, Type III may fail to deal with some important security features, such as forward secrecy and resistance to known key attack. As compared to Li-Lee's model [44] and Yang et al.'s model [93], our model has explicitly taken the malicious card reader into consideration, and $\mathcal{A}$ is further armed with the capabilities of C-3 and C-4. Moreover, $\mathcal{A}$ in our model is assumed to be able to offline enumerated all the (ID, PW) pairs in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$ within polynomial time, which enables our model to deal with the special security issues such as resistance to offline password (more precisely, (ID, PW) pair) guessing attack and undetectable online password guessing attack, in dynamic-ID schemes. Note that, C-01 has also been but implicitly made in [44,84,93], all of which do not concern the feature of user anonymity, for the emphasis of C-01 (e.g., we deliberately separate it from C-00 and list it as an independent item) is meaningful only when this admired feature is considered. Consequently, our model is stronger and indeed reasonable as it incorporates the previous assumptions as well as other new practical (i.e., the computational power of $\mathcal{A}$ is large but not omnipotent) assumptions, especially when considering the current and future proliferation of mobile device use cases.

2.2 Evaluation criteria

As pointed out in [92], although the construction and security analysis of password-based authentication schemes with smart cards have a long history, there is no common set of desirable security properties that has been widely adopted for the construction of this type of schemes. In 2006, Liao et al. [49] made an attempt to consolidate a large set of ten desirable properties, including six security requirements, for evaluating the goodness of a password-based authentication scheme using smart card. Later on, Yang et al. [92] argued that Liao et al.'s criteria set has some redundancies and proposed a new set of only five criteria for rating the schemes. Yang et al.'s criterion set is too conceptual (and thus ambiguous, not specific) to be adopted in real applications. Almost at the same time, Tsai et al. [72] also presented another list of nine security requirements and ten desirable features that an ideal password authentication scheme should achieve. A common feature of both Liao et al.'s and Tsai et al.'s criteria is that, the security requirements are based on the temper-resistance assumption of the smart cards, which may be inconsistent with the reality when taking into account the state-of-the-art techniques of side-channel cryptanalysis.

More recently, Madhusudhan and Mittal [54] pointed out that earlier criteria sets have redundancies and ambiguities and also proposed a new criteria set of nine security requirements and ten desirable features to evaluate this type of schemes. Since the security requirements of their criteria are based the non-temper resistance assumption of the smart cards, their criteria set is superior to other proposed sets. However, it fails to include some important security requirements for an authentication protocol with key agreement, i.e., resistance to known key attack, key compromise impersonation attack and unknown key share attack [4, 40].

By summarizing these earlier studies, we put forward a comprehensive list of twelve independent criteria in terms of user friendliness and security that a password-based remote user authentication scheme with smart card should satisfy:

- C1. the server needs not to maintain a database for storing the passwords or some derived values of the passwords of its clients [49, 54, 72, 93];
- C2. the password is memorable, and can be chosen freely and changed locally by the user [49, 54, 72, 93];
- C3. the password cannot be derived by the privileged administrator of the server [49, 54, 72, 93];
- C4. the scheme is free from smart card loss attack, i.e., unauthorized users should not be able to easily change the password of the smart card, guess the password of the user by using password guessing attacks, or impersonate the user to login to the system, even if the smart card is obtained and/or secret data in the smart card is revealed [51, 54, 69, 72, 81, 89];
- C5. the scheme can resist various kinds of sophisticated attacks, such as offline password guessing attack, replay attack, parallel session attack, denial of service attack, stolen verifier attack, impersonation attack, key compromise impersonation attack, known key attack [46, 51, 54, 72, 81];
- C6. the scheme provides smart card revocation with good repairability, i.e., the client can revoke the smart card without changing her identity [18, 46, 54, 80, 81];;
- C7. the client and the server can establish a common session key during the authentication process [46, 51, 54, 69, 72];
- C8. the scheme is not prone to the problems of clock synchronization and time-delay [11, 44, 80, 81, 87];
- C9. the scheme provides the property of timely wrong password detection, i.e. the user will be timely notified if he inputs wrong password by mistake in login phase [54, 72, 82];

Table 2. A comparative evaluation of two-factor authentication schemes

Scheme	Year	Ref.	No verifier table (C1)	Password friendly (C2)	No password exposure (C3)	No smart card loss problem (C4)	Resistance to known attacks (C5)	Sound repairability (C6)	Provision of key agreement (C7)	No clock synchronization (C8)	Timely typo detection (C9)	Mutual authentication (C10)	User anonymity (C11)	Forward secrecy (C12)
Islam-Biswas	2013	[30]	·	·	·	✓	·	✓	✓	✓	·	✓	·	✓
Chang et al.	2013	[12]	✓	·	·	·	·	·	·	·	·	✓	·	·
Li	2013	[45]	✓	·	✓	·	·	✓	✓	✓	·	✓	✓	✓
Li-Zhang	2012	[48]	✓	·	·	✓	·	✓	✓	·	·	✓	✓	✓
Kim *et al.*	2012	[35]	✓	·	·	✓	·	✓	✓	✓	·	✓	✓	·
Wu *et al.*	2012	[87]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	✓	✓
Wang-Ma	2012	[75]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	✓	✓
Hsieh-Leu	2012	[27]	✓	·	·	·	✓	✓	·	·	·	✓	·	·
Wen-Li	2012	[85]	✓	·	·	·	·	✓	✓	·	·	✓	·	·
Chen *et al.*	2012	[13]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Wang *et al.*	2012	[76]	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓
He *et al.*	2012	[24]	✓	·	✓	✓	✓	·	✓	·	·	✓	·	·
Wang *et al.*	2011	[80]	✓	✓	·	·	·	✓	✓	✓	·	✓	✓	✓
Chen *et al.*	2011	[15]	✓	✓	✓	·	·	·	✓	·	·	✓	·	·
Khan *et al.*	2011	[33]	✓	✓	✓	·	✓	✓	✓	·	✓	✓	·	·
Kim *et al.*	2011	[34]	✓	·	·	✓	✓	✓	✓	·	·	✓	·	·
Awasthi *et al.*	2011	[1]	✓	·	·	·	·	·	·	·	·	✓	·	·
Li-Lee	2011	[44]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Li *et al.*	2010	[47]	✓	✓	✓	✓	·	✓	✓	✓	·	✓	✓	·
Tsai *et al.*	2010	[73]	✓	✓	·	·	✓	·	✓	·	·	✓	✓	·
Song	2010	[68]	✓	·	·	·	✓	·	✓	·	·	✓	·	·
Yeh *et al.*	2010	[94]	✓	✓	·	·	✓	✓	✓	·	·	✓	·	·
Horng *et al.*	2010	[25]	✓	·	✓	✓	·	·	✓	✓	·	✓	✓	✓
Sun *et al.*	2009	[70]	✓	✓	✓	·	✓	✓	✓	✓	·	✓	·	✓
Hsiang-Shi	2009	[26]	✓	✓	·	·	✓	✓	·	·	·	✓	·	·
Xu *et al.*	2009	[91]	✓	·	·	·	·	·	✓	·	·	✓	·	✓
Kim-Chung	2009	[36]	✓	✓	·	·	·	·	·	·	✓	✓	·	·
Chung *et al.*	2009	[18]	✓	·	✓	✓	✓	✓	✓	✓	·	✓	·	✓
Ramasamy	2009	[63]	✓	·	·	✓	✓	✓	·	·	·	✓	·	·
Yang *et al.*	2008	[93]	✓	✓	✓	·	✓	·	✓	✓	·	✓	·	✓
Juang *et al.*	2008	[31]	✓	·	✓	✓	·	✓	✓	✓	·	✓	·	·
Chen *et al.*	2008	[16]	✓	·	✓	·	·	·	·	✓	·	✓	·	·
Wang *et al.*	2007	[82]	·	✓	✓	·	·	·	✓	·	✓	✓	·	·
Wang *et al.*	2007	[81]	✓	·	✓	·	·	✓	✓	✓	·	✓	·	·
Liao *et al.*	2006	[49]	✓	✓	✓	·	·	·	✓	·	·	✓	·	✓
Lee et al.	2005	[42]	✓	·	·	·	·	·	·	·	·	✓	·	·
Fan et al.	2005	[20]	✓	·	·	✓	✓	✓	·	✓	·	✓	·	·
Lu-Cao	2005	[50]	✓	·	·	✓	✓	·	·	·	·	·	·	·
Yoon *et al.*	2004	[95]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·
Ku *et al.*	2004	[41]	✓	✓	✓	·	·	✓	·	·	·	✓	·	·

C10. the scheme can achieve mutual authentication [49, 54, 72, 93];
C11. the scheme preserves user anonymity[1] to avoid partial information leakage. [51, 54, 69, 72, 79];
C12. the scheme provides the property of forward secrecy [18, 53, 54, 72, 87].

Our criteria set is a refinement and an extension of some previously proposed requirement sets, it not only eliminates the redundancies and ambiguities of the old requirement sets, but also facilitates cryptanalysis due to its concreteness. It is not difficult to check that Madhusudhan and Mittal's criteria set is entirely included into our set. And it is also worth noting that, unlike the criteria sets proposed by Tsai et al. [72] and Liao et al. [49], the criterion concerning with performance, which says "The scheme must be efficient and practical", is not incorporated into our set. The main reason is that, it does not seem to be measurable without referring to other related schemes, in other words, isolating it from the criteria set can make our set more concrete and decidable. Furthermore, the efficiency of a scheme may depend on the implementation environment, while practicality is largely related to the target applications [93]. Except this criterion, all the other criteria in [72] and [49] are included into our set. However, as one could argue that there is the probability that someone else claims tomorrow that a list they come up with is better, we maintain focus on criteria that have been discussed in the past literature, for a criteria that has drawn little or no attention probably can not be considered essential. Though not cast in stone, our criteria set is more comprehensive and concrete than other ones.

Having said that, we expect it is the systematic evaluation framework, as a whole, that constitutes the main long-term scientific value, but neither our criteria set nor our adversary model alone does. The effectiveness of this framework is demonstrated and tested by rating 40 two-factor authentication schemes without hidden agenda[4], as summarized in a carefully constructed comparative table (see Table 2). Both the rating criteria and their definitions were iteratively refined over the evaluation of these schemes. It is also worth noting that, in selecting a particular scheme for inclusion in the comparison table, we do not necessarily endorse it as better than alternatives that are not included in the table—merely that it is reasonably representative, or illuminates in some way what the category (from a point view of the development tree where a specific scheme lies, see Fig.1) it belongs to can achieve.

3 Our proposed scheme

In this section, we present a robust and efficient smart card based password authentication scheme that provides all of the twelve criteria introduced in Section 2.2. Our scheme (summarized in Fig.4, in Appendix D) is based on one of our previous works [76] and consists of four phases: the registration phase, the login phase, the verification phase and the password change phase. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 3.

[1] Note that, in the scenario of remote user authentication, user anonymity includes two aspects, i.e. the identity protection and the user un-traceability, it is defined against the public (eavesdropping attackers) rather than the server because the server has to first identify the legitimacy of the user and then obtain the user's real identity for accounting and billing purposes. [56]

[4] The present authors have examined more than one hundred such schemes (some quite recent cryptanalysis results include [52, 77–79]), and also contributed to the proposal of the following schemes: Wang et al. [76], Wang-Ma [75] and Ma et al. [51]. We invite readers to verify that we have evaluated them impartially.

Table 3. Notations

Symbol	Description
U_i	i^{th} user
S	remote server
$\mathcal{A}$	malicious attacker
SC	smart card
ID_i	identity of user U_i
PW_i	password of user U_i
x	the secret key of remote server S
$\oplus$	the bitwise XOR operation
$\|$	the string concatenation operation
$A \rightarrow B : C$	message C is transferred through a common channel from A to B
$A \Rightarrow B : C$	message C is transferred through a secure channel from A to B

3.1 Registration phase

The protocol is defined over a finite cyclic group $\mathbb{G} =< g >$ of order a ℓ-bit prime number q. This group could be $\mathbb{G} = Z_q^*$, where $Z_q^* = 1, 2, \cdots q-1$, or it could be a subgroup of $GF(p)$, or it could be an elliptic curve group. We denote the group operation multiplicatively. In this paper, we assume $\mathbb{G}$ is a prime order subgroup of F_p, where p also is a large prime number such that $q|p-1$. Hash functions from $\{0,1\}^* \rightarrow \{0,1\}^{l_i}$ are denoted by $\mathcal{H}_i (i = 0, 1, 2, 3)$, where l_i is the bit length of function output, e.g. $l_i = 160$. We also define a medium integer n, $2^8 \leq n < 2^{16}$, which determines the capacity of the pool of the (ID, PW) pair against offline guessing attack. Let $(x, y = g^x \bmod p)$ denote the server S's private key and its corresponding public key, where x is kept secret by S and y is stored inside each user's smart card. The registration phase involves the following operations:

Step R1. U_i chooses her identity ID_i, password PW_i and a random number b.
Step R2. $U_i \Rightarrow S : \{ID_i, \mathcal{H}_0(b \parallel PW_i)\}$.
Step R3. On receiving the registration message from U_i at time T, S first checks whether U_i is a registered user. If it is U_i's initial registration, S creates an entry for U_i in the account-database and stores $(ID_i, T_{reg} = T)$ in this entry.[2] Otherwise, S updates the value of T_{reg} with T in the existing entry for U_i. Next, S computes $N_i = \mathcal{H}_0(b \parallel PW_i) \oplus \mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ and $A_i = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \parallel PW_i)) \bmod n)$.
Step R4. $S \Rightarrow U_i$: A smart card containing security parameters $\{N_i, A_i, q, g, y, n, \mathcal{H}_0(\cdot), \mathcal{H}_1(\cdot), \mathcal{H}_2(\cdot), \mathcal{H}_3(\cdot)\}$.
Step R5. Upon receiving the smart card SC, U_i enters b into SC.

3.2 Login phase

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

Step L1. U_i inserts her smart card into the card reader and inputs ID_i^*, PW_i^*.
Step L2. SC computes $A_i^* = \mathcal{H}_0((\mathcal{H}_0(ID_i^*) \oplus \mathcal{H}_0(b \| PW_i^*)) \bmod n)$ and verifies the validity of ID_i^* and PW_i^* by checking whether A_i^* equals the stored A_i. If the verification holds, it implies $ID_i^* = ID_i$ and $PW_i^* = PW_i$ with a probability of $\frac{n-1}{n} (\approx \frac{99.90}{100}$, when $n = 2^{10})$. Otherwise, the session is terminated.

[2] For the mere sake of achieving provable security, in Section IV we assume another parameter $P_i = \mathcal{H}_0(b \| PW_i)$ is also stored in this entry.

Step L3. SC chooses a random number u and computes $C_1 = g^u \bmod p$, $Y_1 = y^u \bmod p$, $k = \mathcal{H}_0(x \parallel ID_i \parallel T_{reg}) = N_i \oplus \mathcal{H}_0(b \parallel PW_i)$, $CID_i = ID_i \oplus \mathcal{H}_0(C_1 \parallel Y_1)$ and $M_i = \mathcal{H}_0(Y_1 \parallel k \parallel CID_i)$.
Step L4. $U_i \rightarrow S : \{C_1, CID_i, M_i\}$.

3.3 Verification phase

After receiving the login request, the server S performs the following operations:

Step V1. S computes $Y_1 = (C_1)^x \bmod p$ using its private key x. Then, S derives $ID_i = CID_i \oplus \mathcal{H}_0(C_1 \parallel Y_1)$ and checks whether ID_i is in the correct format. If ID_i is not valid, the session is terminated. Then, S computes $k = \mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ and $M_i^* = \mathcal{H}_0(Y_1 \parallel k \parallel CID_i)$, where T_{reg} is extracted from the entry corresponding to ID_i. If M_i^* is not equal to the received M_i, the session is terminated. Otherwise, S generates a random number v and computes the temporary key $K_S = (C_1)^v \bmod p$, $C_2 = g^v \bmod p$ and $C_3 = \mathcal{H}_1(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_S)$.
Step V2. $S \rightarrow U_i : \{C_2, C_3\}$.
Step V3. On receiving the reply message from the server S, SC computes $K_U = (C_2)^u \bmod p$, $C_3^* = \mathcal{H}_1(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_U)$, and compares C_3^* with the received C_3. This equivalency authenticates the legitimacy of the server S, and U_i goes on to compute $C_4 = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_U)$.
Step V4. $U_i \rightarrow S : \{C_4\}$
Step V5. Upon receiving $\{C_4\}$ from U_i, the server S first computes $C_4^* = \mathcal{H}_2(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, S authenticates the user U_i and the login request is accepted else the connection is terminated.
Step V6. The user U_i and the server S agree on the common session key $sk_U = \mathcal{H}_3(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_U) = \mathcal{H}_3(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_S) = sk_S$ for securing future data communications.

3.4 Password change phase

For the sake of security, user friendliness and communication efficiency, this phase is performed locally without the hassle of interaction with the remote authentication server, and it involves the following steps:

Step P1. U_i inserts her smart card into the card reader and inputs ID_i and the original password PW_i.
Step P2. The smart card computes $A_i^* = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \parallel PW_i)) \bmod n)$ and verifies the validity of A_i^* by checking whether A_i^* equals to the stored A_i. If the verification holds, it implies the input ID_i and PW_i are valid. Otherwise, the smart card rejects.
Step P3. The smart card asks the cardholder to resubmit a new password PW_i^{new} and computes $N_i^{new} = N_i \oplus \mathcal{H}_0(b \parallel PW_i) \oplus \mathcal{H}_0(b \parallel PW_i^{new})$, $A_i^{new} = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b \parallel PW_i^{new}) \bmod n)$. Then, smart card updates the values of N_i and A_i stored in its memory with N_i^{new} and A_i^{new}, respectively.

Notes and Rationales. To achieve criteria C2 and C4 at the same time, a verification of the authenticity of the original password before updating the value of N_i in the memory of smart card is essential. And thus, besides N_i, some additional parameter(s) should be stored in the smart card, which may introduce new vulnerabilities, such as offline guessing attack and user

impersonation attack. To gain a better insight into this issue, now let's assume an additional parameter $A_i = \mathcal{H}_0(ID_i \parallel \mathcal{H}_0(PW_i))$ is stored in the smart card. Whenever U_i wants to change her password, first she must submit her identity ID_i^* and password PW_i^*, then the smart card checks whether $\mathcal{H}_0(ID_i^* \parallel \mathcal{H}_0(PW_i^*))$ equals the stored A_i. One can easily find that an adversary $\mathcal{A}$ can exhaustively search the correct (ID_i, PW_i) pair in an offline manner once the parameter A_i is obtained, which definitely leads to an offline guessing attack, resulting in the violation of C4. What we have just described directly applies to one of our earlier works [76], the parameter A_i in [76] is exactly computed in this insecure manner and thus $\mathcal{A}$ can obtain the exactly correct (ID_i, PW_i) pair once the parameter A_i is revealed under our adversary model illustrated in Table 1 [3].

However, if the parameter A_i is computed as $A_i = \mathcal{H}_0(\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(PW_i)) \bmod n)$, one can be assured that there exists $\frac{|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|}{n} \approx 2^{30}$ candidates of (ID, PW) pair to frustrate $\mathcal{A}$ when $|\mathcal{D}_{id}| = |\mathcal{D}_{pw}| = 10^6$ [6,38,88] and $n = 2^{10}$, where $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the size of the identity space and password space, respectively. Even with the capability of C-01 (i.e., the victim user's identity has already been learnt), $\mathcal{A}$ will still be frustrated for there exists $\frac{|\mathcal{D}_{pw}|}{n} \approx 2^{10}$ candidates of (ID, PW) pair, and there is no other way than launching an online password guessing attack to determine the exactly correct one. In this manner, we prevent $\mathcal{A}$ from obtaining the exactly correct (ID, PW) pair and we call the parameter A_i calculated through this new method "a fuzzy verifier". An obvious "side effect" of this "fuzzy verifier" is that it can be used to fulfill criterion C9. One may argue that what if U_i happens to submit a wrong (ID_i^*, PW_i^*) pair such that $\mathcal{H}_0((\mathcal{H}_0(ID_i^*) \oplus \mathcal{H}_0(PW_i^*)) \bmod n) = A_i$, while $(ID_i^*, PW_i^*) \neq (ID_i, PW_i)$? The reality is that, this possibility is $\frac{1}{n} < \frac{1}{1000}$, which is too small to significantly degrade the effectiveness of C9, and we conjecture that there is an unavoidable trade-off when fulfilling C4 and C9.

To achieve C3, $\mathcal{H}_0(b \parallel PW_i)$ instead of PW_i or $h(PW_i)$ is submitted to server S, where b is a random number unknown to the server S; to achieve C6, an entry (ID_i, T_{reg}) corresponding to U_i is stored in S's database,[4] only T_{reg} needs to be updated when user U_i revokes her smart card; to achieve C8, a nonce based mechanism instead of the timestamp based design is preferred to provide the freshness of the messages; to achieve user anonymity (C11), user's real identity ID_i is concealed in the session-variant pseudo-identity CID_i, by employing an analogous Elgamal encryption; to achieve forward secrecy (C12), Diffie-Hellman key exchange technique is adopted.

4 Security analysis

In the following, we first describe a formal security model for smart card based password authentication schemes, and then show that our scheme is secure in this model under the assumptions that the hash function closely behaves like a random oracle and that the computational Diffie-Hellman problem is difficult. In particular, our protocol achieves forward secrecy property and security against known key attack, key compromise impersonation attack.

[3] In [76], the authors claimed that their scheme can withstand offline guessing attack even if the smart card security is breached. Why we get quite contradictory results? The reason is that the claim made in [76] is based on the assumption that $\mathcal{A}$ cannot guess both ID_i and PW_i correctly at the same time in polynomial time. In other words, $\mathcal{A}$ in [76] is not equipped with the capability C-00 in Table 1. This explicates the paradox.

[4] Note that the storage of such data in server's database will not violate criterion C1.

4.1 Formal security model

We define some notions and recall the BPR2000 security model [3] where the adversary's capabilities are modelled through queries. However, we do not use the original model directly, but adopt the reified version proposed by Bresson et al. [9] with a few changes so that we can define the special security requirements for password authentication schemes using smart cards. We refer the reader to the original papers for more details.

Players. We denote a server S and a user, or client, U that can participate in the authentication protocol $\mathcal{P}$. Each of them 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 $S^j, i, j \in \mathbb{Z}$, and denote any kind of instance by I.

According to our scheme, the server has a long-term private/public key pair $(x, y = g^x)$. User U_i holds a password PW_i, which is uniformly drawn from a small dictionary $\mathcal{D}$ of size $|\mathcal{D}|$. Additionally, when the user U_i enrolls in the server S, S stores N_i and A_i into a smart card and issues it to the user U_i, where N_i and A_i are (injective) transformations of PW_i and x. The assumption of the uniform distribution for the password is just to make notations simpler, but everything would work with any other distribution, replacing the probability $\frac{\lambda}{|\mathcal{D}|}$ by the sum of the probabilities of the λ most probable passwords.

Queries. The interaction between an adversary $\mathcal{A}$ and the protocol participants occurs only via oracle queries, which model the adversary capabilities in a real attack. The query types available to $\mathcal{A}$ are defined as follows.

- Execute(U^i, S^j): This oracle query is used to model passive (eavesdropping) attacks of the adversary. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.
- Send(I, m): This query models an active attack, in which the adversary $\mathcal{A}$ may send a message to instance I and get back the response I generates in processing the message m according to the protocol $\mathcal{P}$. A query Send$(U^i$, Start) initializes the key exchange protocol. Start is a message, and thus the adversary receives the flow the client should send out to the server.
- Test(I): This oracle query is not used to simulate the adversary's attack, but to define session key's semantic security. If no session key for instance I is defined, then undefined symbol $\perp$ is returned. Otherwise, a private coin c is flipped. If $c = 1$ then the session key sk is returned to $\mathcal{A}$, otherwise a random key of the same size is returned. This query can be called only once during its execution.
- Reveal(I): This query models the misuse of session keys (i.e. C-3 in Table 1). It returns to the adversary the session key sk of participant instance I, if the targeted instance actually "holds" a session key, and I and its partner were not asked by a Test query. Otherwise the $\perp$ is returned.
- Corrupt(I, a): This query models corruption capability of the adversary (i.e. C-2 and C-4 in Table 1). $\mathcal{A}$ can indeed steal/break either one of the two authentication factors of clients, but not both:
  - If $a = 1$, it outputs the password PW_i of U.
  - If $a = 2$, it outputs parameters, i.e. $\{N_i, A_i, b\}$, stored in the smart card.
  - If $a = 3$, it outputs the private key x of S.

It is easy to see that, the above oracle queries indeed can model all the adversary capabilities listed in Table 1.

Partnering. We define partnering by using the notion of session identifier *sid*. Let U^i and S^j be a pair of instances. 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 *sid*; ③The partner identifier (pid) of U^i is S and vice-versa. In general, we let *sid* be the ordered concatenation of all messages sent and received by the instance U^i (or S^j).

Freshness. The freshness notion captures the intuitive fact that a session key can not 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 the client involved (either I or its partner) and no Corrupt-query is made to the server involved (either I or its partner), since the beginning of the game.

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

Authentication. A fundamental goal of the authentication schemes is to prevent the adversary from impersonating the client or the server. We denote by $\mathrm{Adv}_{\mathcal{P}}^{\mathrm{auth}}(\mathcal{A})$ the probability that $\mathcal{A}$ successfully impersonates a participant as an instance of either U or S in an execution of $\mathcal{P}$, which means that S (resp. U) agrees on a key, while the latter is shared with no instance of U (resp. S).

Semantic security. Another 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 Execute-query, Reveal-query, Corrupt-query and Send-query. It can also ask a single Test query to a fresh instance. In the end of the game, $\mathcal{A}$ outputs a guess bit c' for the bit c involved in the Test-query. We say that $\mathcal{A}$ wins the game if $c' = c$, and this event is denoted by Succ. Accordingly, the advantage of $\mathcal{A}$ in breaking the semantic security of the protocol $\mathcal{P}$ is defined to be

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

where the probability space is over all the random coins of the adversary and all the oracles. The protocol $\mathcal{P}$ is said to be semantically secure if any polynomial time (PPT) adversary $\mathcal{A}$'s advantage is negligible in the security parameter l.

4.2 Formal security proof

Before stating the security results, we recall the computational assumption on which the formal security proof relies.

Computational Diffie–Hellman (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. A (t, ϵ)-CDH attacker in $\mathbb{G}$ is a probabilistic machine Δ running in time t such that

$$\mathrm{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(\Delta) = \Pr_{x,y}[\Delta(g^x, g^y) = g^{xy}],$$
$$\mathrm{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(t) = \max_{\Delta}\{\mathrm{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(\Delta)\},$$

where the probability is taken over the random values x and y. The CDH-Assumption states that $\mathrm{Adv}_{g,\mathbb{G}}^{\mathrm{CDH}}(t) \leq \epsilon$ for any t/ϵ not too large.

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, with less than q_{send} Send-queries and q_{exe} Execution-queries, and making less than q_h random oracle queries. Then we have*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ake}}_{\mathcal{P},\mathcal{G}}(\mathcal{A}) &= 2\Pr[\mathrm{Succ}_7] - 1 + 2(\Pr[\mathrm{Succ}_0] - \Pr[\mathrm{Succ}_7]) \\ &\leq \tfrac{2q_{send}}{|\mathcal{D}|} + 12q_h\mathrm{Adv}^{\mathrm{CDH}}_{\mathcal{P}}(t') + \tfrac{q_h^2+6q_{send}}{2^l} + \tfrac{(q_{send}+q_{exe})^2}{p},\end{aligned}$$

where $t' \leq t + (q_s + q_p + 1)\tau_e$, τ_e *is the computational time for an exponentiation in* $\mathbb{G}$, *and* $l = min\{l_i\}, i = 0, 1, 2, 3$.

Proof. Let be $\mathcal{A}$ an adversary against the semantic security of our scheme. The idea is to employ $\mathcal{A}$ to construct adversaries for each of the underlying primitives in such a way that if $\mathcal{A}$ succeeds in breaking the semantic security, then at least one of these adversaries manages to breach the security of an underlying primitive. We prove Theorem 1 through a series of hybrid games, starting with the real attack and ending up with 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. And the details of the proof can be found in Appendix D.

Theorem 2. *Let* $\mathbb{G}$ *be a represent 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 mutual authentication within a time bound* t, *with less than* q_{send} *Send-queries and* q_{exe} *Execution-queries, and making less than* q_h *random oracle queries. Then we have*

$$\mathrm{Adv}^{\mathrm{auth}}_{\mathcal{P},\mathcal{G}}(\mathcal{A}) \leq \tfrac{q_{send}}{|\mathcal{D}|} + 5q_h\mathrm{Adv}^{\mathrm{CDH}}_{\mathcal{P}}(t') + \tfrac{q_h^2+6q_{send}}{2^{l+1}} + \tfrac{(q_{send}+q_{exe})^2}{2p},$$

where $t' \leq t + (q_s + q_p + 1)\tau_e$, τ_e *is the computational time for an exponentiation in* $\mathbb{G}$ *and* $l = min\{l_i\}, i = 0, 1, 2, 3$.

Proof. The proof is similar to the previous one. But one can find more details in Appendix D.2.

4.3 Withstand possible attacks

As stated by Menezes [57], a formal security proof isn't an absolute guarantee of security and the old-fashioned cryptanalysis still plays an important (even indispensible) role in establishing and maintaining confidence in the security of a cryptographic protocol. Accordingly, in the following, we discuss some possible attacks presented in related schemes and perform the heuristic security analysis for these attacks on the proposed scheme to verify whether the security requirements mentioned in Section 2 have been satisfied under our adversary model. The security of our proposed authentication scheme is based on the secure hash function and the discrete logarithm problem. To facilitate the analysis, we define two adversaries:

- $\mathcal{A}_1$: with the capabilities of C-00, C-01, C-1, C-2(i), C-3 and C-4.
- $\mathcal{A}_2$: with the capabilities of C-00, C-01, C-1, C-2(ii), C-3 and C-4.

One can easily see that all the adversaries under our adversary model are either $\mathcal{A}_1$ or $\mathcal{A}_2$. Since some high entropy parameters, such as N_i and b, are involved in the authentication process, without them an adversary poses no threat to our scheme. Hence, in the following we only focus on the adversary with capability C-2(ii), that is the adversary $\mathcal{A}_2$.

In particular, with the ability of C-2(ii), the secret information stored in the smart card can be revealed by the adversary $\mathcal{A}_2$, i.e., the security parameters N_i, A_i, b and y can be obtained by a malicious privileged attacker. It is obvious to see that: 1) our scheme can resist against replay attack as fresh nonces are included in the message M_i, C_3 and C_4; 2) our scheme can resist reflection attack and parallel session attack as the message structures of M_i, C_3 and C_4 are inherently different; 3)our scheme is free from known key attack as neither the session key $SK = \mathcal{H}_3(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_U)$ is of any part of the transmitted messages nor $\mathcal{A}_2$ can learn any other useful information form SK under the one-way nature of hash functions; 4)

our scheme is free from stolen-verifier attack as the user-specific information (ID_i, T_{reg}) stored on the server are not password-involved verifiers; 5)our scheme can achieve forward secrecy as the Diffie-Hellman key exchange technique is employed.

(1) User anonymity: Suppose that the attacker has intercepted U_i's authentication messages $\{CID_i, M_i, C_1, C_2, C_3, C_4\}$. Then, the attacker may try to retrieve any static parameter from these messages, but these messages are all session-variant and indeed random strings due to the randomness of u and/or v. Accordingly, without knowing the random number u, the adversary (an eavesdropping adversary) will face to solve the discrete logarithm problem to retrieve the correct value of ID_i from CID_i, while ID_i is the only static element corresponding to U_i in the transmitted messages. Hence, the proposed scheme can preserve identity-protection and user un-linkability.

(2) Password disclosure to server: With $\mathcal{H}_0(b \parallel PW_i)$ instead of plaintext password PW_i submitted to server S, it is computationally infeasible to derive PW_i from $h(b \parallel PW_i)$ without knowing the random number b due to the one-way property of the secure hash function.

(3) User impersonation attack: As M_i and C_4 are protected by secure one-way hash function, any modification to the legitimate user U_i's authentication messages will be detected by the server S if the attacker cannot fabricate these two valid M_i and C_4. Even with the smart card security breached, without U_i's password the attacker has no way to obtain the correct value of $\mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$, and thus she cannot fabricate the valid M_i and C_4. Therefore, the proposed protocol is secure against user impersonation attack.

(4) Server masquerading attack: On the one hand, in the proposed protocol, a malicious server MS cannot compute the correct $Y_1 = (C_1)^x \bmod n$ because she does not know the value of S's private key x. On the other hand, without knowing the valud of U_i's registration timestamp T_{reg} and S's private key x, MS has to break the secure one-way hash function to retrieve $k = \mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$. As a result, without the correct value of Y_1 and k, it is impossible for MS to fabricate the proper $C_3 = \mathcal{H}_1(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_U)$ to pass the verification of U_i in Step $V3$ of the verification phase. Therefore, the proposed protocol is secure against server masquerading attack.

(5) Smart card loss attack(offline password guessing attack): Generally, the attacker $\mathcal{A}_2$ has two potential ways to launch an offline password guessing attack: one by interacting with server S (e.g, the attack introduced in [66]) and the other by not interaction with S. We show that these two ways are infeasible even $\mathcal{A}_2$ has learnt U_i's identity ID_i and gathered the secret information b, N_i, A_i, n and y stored in the card. On the one hand, since both M_i and C_3 are computed with the contribution of the key parameter k, the receiver will detect the abnormality and reject the session if $\mathcal{A}_2$ (the sender) cannot obtain the correct value of k, and thus the former way will not be viable. On the other hand, if $\mathcal{A}_2$ adopts the latter approach, as there is no transcripts over the public channel involving user password, the only way for $\mathcal{A}_2$ to learn some information about PW_i is to find out the correct pair (ID_i^*, PW_i^*) such that $\mathcal{H}_0((\mathcal{H}_0(ID_i^*) \oplus \mathcal{H}_0(PW_i^*)) \bmod n) = A_i$. In Section 3, we have shown that this strategy of $\mathcal{A}_2$ is infeasible as A_i is a "fuzzy verifier". Without the knowledge of PW_i, the password change procedure will not function; without the knowledge of PW_i, the correct value of $\mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ cannot be obtained and thus user impersonation attack will not succeed. In conclusion, the proposed scheme can resist smart card loss attack.

(6) Mutual authentication: In our dynamic ID-based scheme, the server authenticates the user by checking the validity of M_i and C_4 in the access request. To pass the authentication of server S, the smart card first needs U_i's identity ID_i and password PW_i to get through the verification in Step L2 of the login phase. In this Section, we have shown that our

scheme can resist offline password guessing attack. Therefore, only the legal user U_i who owns correct ID_i and PW_i can compute the correct value of $\mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ to pass the authentication of server S. On the other hand, the user U_i authenticates server S by explicitly checking whether the other party communicated with can compute the valid C_3 or not. Since the malicious server does not know the correct value of x corresponding to server S, only the legitimate server can compute the correct $C_3 = \mathcal{H}_1(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_U)$. From the above analysis, we conclude that our scheme can achieve mutual authentication.

(7) Denial of service attack: Assume that an adversary has got a legitimate user U_i's smart card. However, in our scheme, the user-specific security parameters stored in the server S do not need to be updated during the authentication phase, the risk of in-consistence is eliminated. Furthermore, the smart card checks the validity of user input identity ID_i^* and password PW_i^* before the password update process. Since the smart card computes $A_i^* = \mathcal{H}_0((\mathcal{H}_0 ID_i^* \oplus \mathcal{H}_0(b \parallel PW_i^*) \bmod n)$ and compares it with the stored value of A_i in its memory to verify the legality of the user before the smart card accepts the password update request. Accordingly, once the number of login failure exceeds the predefined system value, the smart card will be locked immediately. Therefore, denial of service attack is thwarted.

(8) Key compromise impersonation attack: Assume that an adversary $\mathcal{A}$ has got server S's private key x. However, $\mathcal{A}$ cannot compute the correct value of $\mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$ as she has no knowledge of a legitimate user U_i's registration time T_{reg}. Without the correct value of $\mathcal{H}_0(x \parallel ID_i \parallel T_{reg})$, $\mathcal{A}$ cannot impersonate U_i.

(9) Undetectable online password guessing attack: Assume an adversary $\mathcal{A}$ has got a legitimate user U_i's smart card and managed to obtain the security parameters N_i, A_i and y. Based on this strong assumption, this type of attack is still infeasible. Malicious actions will be detected by S if $\mathcal{A}$ guesses the password online, as S can compute $\mathcal{A}$'s ID in Step V1 of the verification phase and lock the corresponding account after a predefined number of failures. Hence, this type of attack can be effectively thwarted.

5 Performance analysis

To evaluate our scheme, we compare the performance and the satisfaction of the criteria among relevant authentication schemes and our proposed scheme in this section. The reason why the schemes presented in [17, 18, 25, 76, 90, 91], instead of other works mentioned earlier in this paper, are selected to compare with is that, these four schemes are the few ones that can achieve forward secrecy. The criteria of a secure and practical remote user authentication scheme are introduced in Section II, and the comparison results of performance are depicted in Table 4.

Table 4. Performance comparison among relevant authentication schemes

	Computation cost	Communication cost	Storage overhead
Our scheme	$6T_E + 14T_H$	2560 bits	3488 bits
Xie [90](2012)	$8T_E + 7T_H$	3328 bits	1280 bits
Wang et al. [76](2012)	$6T_E + 12T_H$	2560 bits	3456 bits
Horng et al. [25](2010)	$7T_E + 4T_S + 8T_H$	2432 bits	3328 bits
Chen et al. [17] (2010)	$6T_E + 5T_H$	2560 bits	3200 bits
Chung et al. [18](2009)	$4T_E + 2T_I + 12T_H$	2560 bits	3200 bits
Xu et al. [91] (2009)	$6T_E + 8T_H$	2816 bits	3200 bits

Since the login phase and verification phase are executed much more frequently than the other two phases, only the computation cost, communication overhead and storage cost during

the login phase and verification phase are taken into consideration. Without loss of generality, the security parameter n is assumed to be 32-bit long, the identity ID_i, password PW_i, random numbers, timestamp values and output of secure one-way hash function are all recommended to be 128-bit long, while p, y and g are all 1024-bit long. Let T_H, T_E, T_I, T_S and T_X denote the time complexity for hash function, exponential operation, inverse operation, symmetric cryptographic operation and XOR operation respectively. Since the time complexity of XOR operation is negligible as compared to the other four operations, we do not take T_X into account. Typically, time complexity associated with these operations can be roughly expressed as $T_E \approx T_I > T_S \geq T_H \gg T_X$ [21].

In our scheme, the parameters $\{b, N_i, A_i, y, g, p, n\}$are stored in the smart card, thus the storage cost is $3488(= 32 + 3 * 128 + 3 * 1024)$bits. The communication overhead includes the capacity of transmitting message involved in the authentication scheme, which is $2560(= 4 * 128 + 2 * 1024)$ bits. During the login and verification phase, the total computation cost of the user and server is $6T_E + 14T_H$. As illustrated in Table 4, the proposed scheme is more efficient than Horng et al.'s scheme and Xie's scheme, enjoys nearly the same performance with Wang et al.'s scheme, Chen et al.'s scheme and Chung et al.'s scheme.

As the smart cards are characterized as low-power, computing-capability-limited devices, it is undoubtedly a critical factor that deserves special attention in designing an authentication protocol suitable for real-life applications, and we give more detailed numerical results for the computation cost of the aforementioned cryptographic operations. Recently, some implementations [29, 61, 65, 86] of cryptographic primitives on microprocessors for low-power mobile devices (e.g., smart cards and sensor nodes) have been presented. According to [86], AES, a widely-in-use symmetric encryption algorithm, gives a throughput of 14.3 Kbps on Palm IIIc with a 20MHz 32-bit Motorola DragonBall-EZ microprocessor, while the the typical Hash algorithm SHA-1 exhibits a throughput of 23.2 Kbps on the same platform. As shown in [65], one 1024-bit T_E only takes 140 ms on a popular 36MHz 32-bit RISC MIPS32-based smart card. The results in [29, 61] also confirm the feasibility and acceptability of our scheme to conduct $3T_E + 8T_H$ on the resource-limited user side.

Table 5. Criteria comparison among relevant authentication schemes

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Our scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Xie [90]	✓	✓	✓	×	×	×	✓	✓	×	✓	✓	✓
Wang [76]	✓	✓	✓	×*	✓	✓	✓	✓	✓	✓	✓	✓
Horng et al. [25]	✓	×	✓	✓	×**	×	✓	✓	×	✓	✓	✓
Chen et al. [17]	✓	×	×	×	×***	×	✓	×	×	✓	×	✓
Chung et al. [18]	✓	×	✓	✓	✓	✓	✓	✓	×	✓	×	✓
Xu et al. [91]	✓	×	×	×	×	×	✓	×	×	✓	×	✓

* As stated in Section 3, the scheme in [76] is vulnerable to an offline password guessing attack once the smart card security is breached under the adversary model illustrated in Table 1.

** In an unpublished work we found Horng et al.'s scheme is prone to undetectable online password guessing attack and key compromise impersonation attack.

*** A reflection attack is identified in their scheme as any adversary can impersonate server S to send $\{M = T, U = V\}$ to U_i on receiving the login request message $\{ID, V, T, N\}$ at any given session.

Table 4 gives a comparison of the admired features of our proposed scheme with the other relevant authentication schemes. Our proposed scheme provides the feature of local password update(C2), while the schemes presented by Horng et al., chen et al., Chung et al. and Xu et al. fail to achieve this feature; Our proposed scheme is free from smart card loss problem (C4), while the schemes presented by Xie, Wang et al., Chen et al. and Xu et al. fail to provide this property; Our proposed scheme preserves user anonymity (C11), while the schemes presented by Chen et al., Chung et al. and Xu et al. do not provide this property; our proposed scheme, Wang et al.'s scheme and Chung et al.'s scheme can resist various kinds of known attacks (C5), while the other three latest schemes suffer from several security vulnerabilities; our proposed scheme and Wang et al.'s scheme provides the property of timely wrong password detection, while the other five schemes fail to achieve it. It is clear that our scheme meets more criteria as compared to other relevant authentication schemes using non-tamper resistant smart cards.

6 Conclusion

In this paper, we have demonstrated damaging attacks on two smart card based password authentication schemes to uncover the subtleties and challenges in designing this type of schemes. As our main contribution, a robust scheme is proposed to remedy these identified flaws, the security and performance analysis demonstrate that our presented scheme achieves all of the twelve independent requirements with high efficiency and thus our scheme is more secure and efficient for practical use. In addition, a practical adversary model and a comprehensive criterion set is proposed to facilitate the design and analysis of this type of schemes in a systematic approach.

Acknowledgments. We sincerely thank Prof. Wang Yongge at UNCC for his quickly and frankly acknowledgement of the ambiguity in the specification of their scheme PSCAV, and for his invaluable comments and generous advice on our attack and our adversary model. He earns our deep respect. We thank Dr. He Debiao at Wuhan University for his insightful discussions on the crystallization of this study. And we are particularly grateful to Prof. M. Scott at Dublin City University for his insightful observations and constructive comments on improving this study. This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61170241 and No. 61073042.

References

1. Awasthi, A.K., Srivastava, K., Mittal, R.: An improved timestamp-based remote user authentication scheme. Computers & Electrical Engineering 37(6), 869–874 (2011)
2. Bao, F., Deng, R.: Privacy protection for transactions of digital goods. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001, LNCS, vol. 2229, pp. 202–213. Springer Berlin / Heidelberg (2001)
3. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000, LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
4. Blake-Wilson, S., Johnson, D., Menezes, A.: Key agreement protocols and their security analysis. In: Darnell, M. (ed.) Crytography and Coding 1997, LNCS, vol. 1355, pp. 30–45. Springer Berlin / Heidelberg (1997)
5. Bogdanov, A., Kizhvatov, I.: Beyond the limits of dpa: Combined side-channel collision attacks. IEEE Transactions on Computers 61(8), 1153–1164 (2012)
6. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: IEEE S&P 2012. pp. 538–552. IEEE Computer Society (2012)

7. Bonneau, J., Just, M., Matthews, G.: Whats in a name? In: Sion, R. (ed.) FC 2010, LNCS, vol. 6052, pp. 98–113. Springer Berlin/ Heidelberg (2010)
8. Boyd, C., Mathuria, A.: Protocols for authentication and key establishment. Springer-Verlag (2003)
9. Bresson, E., Chevassut, O., Pointcheval, D.: Security proofs for an efficient password-based key exchange. In: ACM CCS 2003. pp. 241–250. ACM (2003)
10. Chang, C.C., Wu, T.C.: Remote password authentication with smart cards. IEE Proceedings-Computers and Digital Techniques 138(3), 165–168 (1991)
11. Chang, C.C., Lee, C.Y.: A secure single sign-on mechanism for distributed computer networks. IEEE Transactions on Industrial Electronics 59(1), 629–637 (2012)
12. Chang, Y.F., Tai, W.L., Chang, H.C.: Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. International Journal of Communication Systems (2013), doi: http://dx.doi.org/10.1002/dac.2368
13. Chen, B., Kuo, W., Wuu, L.: Robust smart-card-based remote user password authentication scheme. International Journal of Communication Systems (2012), doi: http://dx.doi.org/10.1002/dac.2368
14. Chen, C.L., Lu, M.S., Guo, Z.M.: A non-repudiated and traceable authorization system based on electronic health insurance cards. Journal of Medical Systems 36(4), 2359–2370 (2011)
15. Chen, T., Hsiang, H., Shih, W.: Security enhancement on an improvement on two remote user authentication schemes using smart cards. Future Generation Computer Systems 27(4), 377–380 (2011)
16. Chen, T.H., Lee, W.B.: A new method for using hash functions to solve remote user authentication. Computers & Electrical Engineering 34(1), 53–62 (2008)
17. Chen, Y.L., Chou, J.S., Huang, C.H.: Improvements on two password-based authentication protocols. Cryptology ePrint Archive, Report 2009/561 (2009), http://eprint.iacr.org/2009/561.pdf
18. Chung, H., Ku, W., Tsaur, M.: Weaknesses and improvement of wang et al.'s remote user password authentication scheme for resource-limited environments. Computer Standards & Interfaces 31(4), 863–868 (2009)
19. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: An empirical analysis. In: INFOCOM 2010. pp. 1–9. IEEE Communications Society (2010)
20. Fan, C., Chan, Y., Zhang, Z.: Robust remote authentication scheme with smart cards. Computers & Security 24(8), 619–628 (2005)
21. Ferguson, N., Schneier, B., Kohno, T.: Cryptography Engineering: Design Principles and Practical Applications. John Wiley & Sons (2010)
22. GROUP, I.W., et al.: Ieee 1363: Standard specifications for public key cryptography. IEEE standard, IEEE, New York, NY 10017 (2005)
23. Hao, F.: On robust key agreement based on public key authentication. In: Sion, R. (ed.) FC 2010, LNCS, vol. 6052, pp. 383–390. Springer-Verlag (2010)
24. He, D., Chen, J., Hu, J.: Improvement on a smart card based password authentication scheme. Journal of Internet Technology 13(3), 38–42 (2012)
25. Horng, W., Lee, C., Peng, J.: A secure remote authentication scheme preserving user anonymity with non-tamper resistant smart cards. WSEAS Transactions on Information Science and Applications 7(5), 619–628 (2010)
26. Hsiang, H.C., Shih, W.K.: Weaknesses and improvements of the yoon-ryu-yoo remote user authentication scheme using smart cards. Computer Communications 32(4), 649–652 (2009)
27. Hsieh, W., Leu, J.: Exploiting hash functions to intensify the remote user authentication scheme. Computers & Security 31(6), 791–798 (2012)
28. Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.H.: A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Transactions on Parallel and Distributed Systems 22(8), 1390–1397 (2011)
29. Hutter, M., Wenger, E.: Fast multi-precision multiplication for public-key cryptography on embedded microprocessors. In: Preneel, B., Takagi, T. (eds.) CHES 2011, LNCS, vol. 6917, pp. 459–474. Springer Berlin / Heidelberg (2011)

30. Islam, S.H., Biswas, G.: Design of improved password authentication and update scheme based on elliptic curve cryptography. Mathematical and Computer Modelling 57(11-12), 2703–2717 (2013)
31. Juang, W.S., Chen, S.T., Liaw, H.T.: Robust and efficient password-authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics 55(6), 2551–2556 (2008)
32. Kasper, T., Oswald, D., Paar, C.: Side-channel analysis of cryptographic rfids with analog demodulation. In: Juels, A., Paar, C. (eds.) RFIDSec 2012, LNCS, vol. 7055, pp. 61–77. Springer Berlin / Heidelberg (2012)
33. Khan, M., Kim, S., Alghathbar, K.: Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme'. Computer Communications 34(3), 305–309 (2011)
34. Kim, J., Choi, H., Copeland, J.: Further improved remote user authentication scheme. IEICE Trans. Fund. Electron. Comm. Comput. Sci. 94(6), 1426–1433 (2011)
35. Kim, K.K., Kim, M.H.: An enhanced anonymous authentication and key exchange scheme using smartcard. In: Juels, A., Paar, C. (eds.) Proceedings of the 16th Annual International Conference on Information Security and Cryptology (ICISC 2012), LNCS, vol. 7839, pp. 487–494,. Springer Berlin / Heidelberg (2012)
36. Kim, S.K., Chung, M.G.: More secure remote user authentication scheme. Computer Communications 32(6), 1018–1021 (2009)
37. Kim, T.H., Kim, C., Park, I.: Side channel analysis attacks using AM demodulation on commercial smart cards with SEED. Journal of Systems and Software 85(12), 2899 – 2908 (2012)
38. Klein, D.V.: Foiling the cracker: A survey of, and improvements to, password security. In: Proceedings of the 2nd USENIX Security Workshop. pp. 5–14 (1990)
39. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology–CRYPTO 1999, LNCS, vol. 1666, pp. 388–397. Springer Berlin / Heidelberg (1999)
40. Krawczyk, H.: HMQV: A high-performance secure diffie-hellman protocol. In: Shoup, V. (ed.) CRYPTO 2005, LNCS, vol. 3621, pp. 546–566. Springer (2005)
41. Ku, W.C., Chen, S.M.: Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron. 50(1), 204–207 (2004)
42. Lee, S., Kim, H., Yoo, K.: Improvement of chien et al.'s remote user authentication scheme using smart cards. Computer Standards & Interfaces 27(2), 181–183 (2005)
43. Leroy, X.: Smart card security from a programming language and static analysis perspective (2013), available at http://pauillac.inria.fr/~xleroy/talks/language-security-etaps03.pdf
44. Li, C.T., Lee, C.C.: A robust remote user authentication scheme using smart card. Information Technology and Control 40(3), 236–245 (2011)
45. Li, C.T.: A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. IET Information Security 7(1), 3–10 (2013)
46. Li, C.T., Lee, C.C., Liu, C.J., Lee, C.W.: A robust remote user authentication scheme against smart card security breach. In: Li, Y. (ed.) Proceedings of 25th Annual IFIP Conference on Data and Applications Security and Privacy (DBSec 2011), LNCS, vol. 6818, pp. 231–238. Springer Berlin / Heidelberg (2011)
47. Li, X., Qiu, W., Zheng, D., Chen, K.F., Li, J.: Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics 57(2), 793–800 (2010)
48. Li, X., Zhang, Y.: A simple and robust anonymous two-factor authenticated key exchange protocol. Security and Communication Networks (2012), doi: http://dx.doi.org/10.1002/sec.605
49. Liao, I., Lee, C., Hwang, M.: A password authentication scheme over insecure networks. Journal of Computer and System Sciences 72(4), 727–740 (2006)
50. Lu, R., Cao, Z.: Efficient remote user authentication scheme using smart card. Computer Networks 49(4), 535–540 (2005)
51. Ma, C.G., Wang, D., Zhang, Q.M.: Cryptanalysis and improvement of sood et al.s dynamic id-based authentication scheme. In: Ramanujam, R., Ramaswamy, S. (eds.) ICDCIT 2012, LNCS, vol. 7154, pp. 141–152. Springer-Verlag (2012)
52. Ma, C.G., Wang, D., Zhao, S.D.: Security flaws in two improved remote user authentication schemes using smart cards. International Journal of Communication Systems, 2012, (2012), doi: http://dx.doi.org/10.1002/dac.2468

53. Ma, C.G., Wang, D., Zhao, P., Wang, Y.H.: A new dynamic id-based remote user authentication scheme with forward secrecy. In: Wang, H., Zou, L., Huang, G., He, J., Pang, C., Zhang, H., Zhao, D., Yi, Z. (eds.) APWeb 2012, LNCS, vol. 7234, pp. 199–211. Springer Berlin / Heidelberg (2012)
54. Madhusudhan, R., Mittal, R.: Dynamic id-based remote user password authentication schemes using smart cards: A review. Journal of Network and Computer Applications 35(4), 1235–1248 (2012)
55. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks: Revealing the secrets of smart cards. Springer-Verlag (2007)
56. Mangipudi, K., Katti, R.: A secure identification and key agreement protocol with user anonymity (sika). Computers & Security 25(6), 420–425 (2006)
57. Menezes, A.: Another look at provable security. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, Lecture Notes in Computer Science, vol. 7237, pp. 8–8. Springer Berlin / Heidelberg (2012), availe at http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf
58. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers 51(5), 541–552 (2002)
59. Nam, J., Kim, S., Won, D.: Security analysis of a nonce-based user authentication scheme using smart cards. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 90(1), 299–302 (2007)
60. Nohl, K., Evans, D., Starbug, S., Plötz, H.: Reverse-engineering a cryptographic RFID tag. In: USENIX Security 2008. pp. 185–193. USENIX Association (2008)
61. Potlapally, N., Ravi, S., Raghunathan, A., Jha, N.: A study of the energy consumption characteristics of cryptographic algorithms and security protocols. IEEE Transactions on Mobile Computing 5(2), 128–143 (2006)
62. Pu, Q., Wang, J., Zhao, R.: Strong authentication scheme for telecare medicine information systems. Journal of Medical Systems pp. 1–11 (2011), doi:http://dx.doi.org/10.1007/s10916-011-9735-9
63. Ramasamy, R., Muniyandi, A.P.: New remote mutual authentication scheme using smart cards. Transactions on Data Privacy 2, 141–152 (2009)
64. Roy, S., Das, A.K., Li, Y.: Cryptanalysis and security enhancement of an advanced authentication scheme using smart cards, and a key agreement scheme for two-party communication. In: Proceedings of 30th International Performance Computing and Communications Conference (IPCCC 2011). pp. 1–7. IEEE (2011)
65. Scott, M., Costigan, N., Abdulwahab, W.: Implementing cryptographic pairings on smartcards. In: Goubin, L., Matsui, M. (eds.) CHES 2006, LNCS, vol. 4249, pp. 134–147. Springer Berlin / Heidelberg (2006)
66. Scott, M.: Cryptanalysis of a recent two factor authentication scheme. Cryptology ePrint Archive, Report 2012/527 (2012), http://eprint.iacr.org/2012/527.pdf
67. Shim, K.: Security flaws in three password-based remote user authentication schemes with smart cards. Cryptologia 36(1), 62–69 (2012)
68. Song, R.: Advanced smart card based password authentication protocol. Computer Standards & Interfaces 32(5), 321–325 (2010)
69. Sood, S.K.: Secure dynamic identity-based authentication scheme using smart cards. Information Security Journal: A Global Perspective 20(2), 67–77 (2011)
70. Sun, D., Huai, J., Sun, J., Li, J., Zhang, J., Feng, Z.: Improvements of juang et al.'s password-authenticated key agreement scheme using smart cards. IEEE Transactions on Industrial Electronics 56(6), 2284–2291 (2009)
71. Tang, C., Wu, D.: Mobile privacy in wireless networks-revisited. IEEE Transactions on Wireless Communications 7(3), 1035 –1042 (march 2008)
72. Tsai, C., Lee, C., Hwang, M.: Password authentication schemes: current status and key issues. International Journal of Network Security 3(2), 101–115 (2006)
73. Tsai, J., Wu, T., Tsai, K.: New dynamic id authentication scheme using smart cards. International Journal of Communication Systems 23(12), 1449–1462 (2010)

74. Veyrat-Charvillon, N., Standaert, F.X.: Generic side-channel distinguishers: Improvements and limitations. In: Rogaway, P. (ed.) CRYPTO 2011, LNCS, vol. 6841, pp. 354–372. Springer Berlin/ Heidelberg (2011)
75. Wang, D., Ma, C.G.: Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards. The Journal of China Universities of Posts and Telecommunications 19(5), 104–114 (2012), doi: http://dx.doi.org/10.1016/S1005-8885(11)60307-5
76. Wang, D., Ma, C.G., Wu, P.: Secure password-based remote user authentication scheme with non-tamper resistant smart cards. In: Cuppens, N., Cuppens, F. (eds.) Proceedings of the 26th IFIP International Conference on Data and Applications Security and Privacy (DBSec 2012), LNCS, vol. 7371, pp. 114–121. Springer-Verlag (2012)
77. Wang, D., Ma, C.G., Zhao, S., Zhou, C.: Breaking a robust remote user authentication scheme using smart cards. In: Park, J.J. (ed.) NPC 2012, LNCS, vol. 7513, pp. 110–118. Springer Berlin / Heidelberg (2012)
78. Wang, D., Ma, C.G., Zhao, S., Zhou, C.: Cryptanalysis of two dynamic id-based remote user authentication schemes for multi-server architecture. In: Xu, L., Bertino, E., Mu, Y. (eds.) NSS 2012, LNCS, vol. 7645, pp. 462–475. Springer Berlin / Heidelberg (2012)
79. Wang, D., Ma, C.: Cryptanalysis of a remote user authentication scheme with provable secuirty for mobile client-server environment based on ECC. Information Fusion, 14(4), 498–503 (2013)
80. Wang, R.C., Juang, W.S., Lei, C.L.: Robust authentication and key agreement scheme preserving the privacy of secret key. Computer Communications 34(3), 274–280 (2011)
81. Wang, R.C., Juang, W.S., Lei, C.L.: A simple and efficient key exchange scheme against the smart card loss problem. In: Denko, M., Shih, C.s., Li, K.C., Tsao (eds.) IEEE/IFIP EUC 2007, LNCS, vol. 4809, pp. 728–744. Springer Berlin / Heidelberg (2007)
82. Wang, X., Zhang, W., Zhang, J., Khan, M.: Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. Computer Standards & Interfaces 29(5), 507–512 (2007)
83. Wang, Y., Liu, J., Xiao, F., Dan, J.: A more efficient and secure dynamic id-based remote user authentication scheme. Computer communications 32(4), 583–585 (2009)
84. Wang, Y.G.: Password protected smart card and memory stick authentication against off-line dictionary attacks. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012, IFIP AICT, vol. 376, pp. 489–500. Springer Boston (2012)
85. Wen, F., Li, X.: An improved dynamic id-based remote user authentication with key agreement scheme. Computers & Electrical Engineering 38(2), 381–387 (2012)
86. Wong, D., Fuentes, H., Chan, A.: The performance measurement of cryptographic primitives on palm devices. In: ACSAC 2001. pp. 92–101. IEEE (2001)
87. Wu, S.H., Zhu, Y.F., Pu, Q.: Robust smart-cards-based user authentication scheme with user anonymity. Security and Communication Networks 5(2), 236–248 (2012)
88. Wu, T.: A real-world analysis of kerberos password security. In: Proceedings of the 1999 ISOC Network and Distributed System Security Symposium. pp. 1–14. Internet Soc. (1999)
89. Xiang, T., Wong, K., Liao, X.: Cryptanalysis of a password authentication scheme over insecure networks. Journal of Computer and system Sciences 74(5), 657–661 (2008)
90. Xie, Q.: Dynamic id-based password authentication protocol with strong security against smart card lost attacks. In: Snac, P., Ott, M., Seneviratne, A., Akan, O. (eds.) Wireless Communications and Applications, LNICST, vol. 72, pp. 412–418. Springer Berlin / Heidelberg (2012)
91. Xu, J., Zhu, W., Feng, D.: An improved smart card based password authentication scheme with provable security. Computer Standards & Interfaces 31(4), 723–728 (2009)
92. Yang, G.M., Wong, D.S., Wang, H.X., Deng, X.T.: Formal analysis and systematic construction of two-factor authentication scheme. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006, LNCS, vol. 4307, pp. 82–91. Springer Berlin / Heidelberg (2006)
93. Yang, G.M., Wong, D.S., Wang, H.X., Deng, X.T.: Two-factor mutual authentication based on smart cards and passwords. Journal of Computer and System Sciences 74(7), 1160–1172 (2008)
94. Yeh, K.H., Su, C., Lo, N.W.: Two robust remote user authentication protocols using smart cards. Journal of Systems and Software 83(12), 2556–2565 (2010)

95. Yoon, E., Ryu, E., Yoo, K.: Further improvement of an efficient password based remote user authentication scheme using smart cards. IEEE Trans. Consum. Electron. 50(2), 612–614 (2004)
96. Zhao, Z., Dong, Z., Wang, Y.G.: Security analysis of a password-based authentication protocol proposed to IEEE 1363. Theoretical Computer Science 352(1), 280–287 (2006)

A Cryptanalysis of Yang et al.'s Scheme

In the paper presented in ICICS'06 and its extended version [93], Yang *et al.* proposed a generic construction framework to convert the conventional provably secure PAKE protocols to smart-card-based versions and further proposed a new scheme to demonstrate its effectiveness. Their new scheme is claimed to be secure and can satisfy all their proposed criteria. In the following, we will show that their scheme is actually vulnerable to a smart card loss attack in which an attacker can easily render the victimized smart card completely unusable once getting temporary access to it, thereby contradicting the claim made in [93] that the new scheme is secure even if the smart card is lost.

Here, we just follow the original notations in [93] as closely as possible. Yang et al.'s scheme consists of four phases: the registration phase, the login-and-authentication phase and the password-changing activity.

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

Registration phase. Server S issues a client A as follows.:

1. A arbitrarily chooses a unique identity ID_A and sends it to S.
2. S calculates $B = PRF_x(H(ID_A)) \oplus H(PW_0)$ where PW_0 is the initial password (e.g. a default such as a string of all '0').
3. S issues A a smart card which contains PK_S, ID_A, B, p, g, q. In practice, we can have all these parameters except B be burned in the read-only memory of the smart card when the smart card is manufactured.
4. On receiving the smart card, A changes the password immediately by performing the password-changing activity (described below).

Login-and-authentication phase. When A wants to login to S, she attaches the smart card to a card reader, and then keys ID_A and PW_A. The smart card checks if the input identity is equal to the value stored in it. If not, the login request is rejected. Otherwise, the smart card retrieves the value $LPW = B \oplus H(PW_A)$. A (actually performed by the client's smart card) and S then use LPW as the password to perform the PWAKE protocol:

1. A first chooses a random number $x \in \mathcal{Z}_p^*$, computes g^x and then sends (A, sid, g^x) to S.
2. Upon receiving the login request, S also chooses a random number $y \in \mathcal{Z}_p^*$, computes $\tau = SIG_{SK_S}(S, A, sid, g^x, g^y)$ and sends (S, sid, g^y, τ) to A.
3. S Upon receiving (S, sid, g^y, τ), A first checks the validity of τ using PK_S. If it is valid, A computes $c = ENC_{PK_S}(LPW_A, A, S, sid, g^x, g^y)$ and sends (A, sid, c) to S.
4. Upon receiving (A, sid, c), S first decrypts c using its private key SK_S and gets $(LPW_A, A, S, sid, g^x, g^y)$. S then checks whether: 1) the decrypted g^x, g^y are the same as previously sent by A and S respectively; 2) the decrypted LPW_A is equal to $PRF_x(H(ID_A))$. If both conditions are satisfied, then S confirms that A is a valid user.
5. In the end, A and S agree on the common session key calculated as $\sigma = g^{xy}$.

Password-changing activity. If A wants to change the password, the following steps is carried out:

1. A select a new password PW'_A.
2. Compute $B' = B \oplus H(PW_A) \oplus H(PW'_A)$, where PW_A is the old password.
3. Replace B with B' in the smart card.

A.1 Smart card loss attack

In password authentication, smart card loss attack can introduce unexpected data during authentication and thus cause permanent error. The most vulnerable point is the password changing activity since it usually refreshes the data on storage. With a victim user's card in hand, if the adversary can change the password (or its related verification data), the updated password (or its related data) will then be different from what the user expects. Even if the smart card is returned to the victim user, the subsequent authentication will never succeed.

In Yang *et al.*'s scheme, the password change phase is conducted on the user terminal without the hassle of interaction with the authentication server, which enhances the security of password change for no sensitive data related to the password needs to be transmitted over the insecure network. Meanwhile, it relieves the overhead of the server. However, the scheme does not provide the smart card with any explicit way to check whether the user-given old (current) password is valid or not, which means anyone in possession of the smart card can change the password without knowing the correct old password. This design strategy is very problematic, for it can give rise to the following damaging situations:

- Accidentally, the user U_i changes the parameter B into an arbitrary value by entering an incorrect value for the old password by mistake. Unfortunately, U_i will not observe this abnormality until she fails to login the server S for a number of times.
- Without knowledge of the correct password, a malicious third party, who gains temporary access to U_i's smart card, can change B into an arbitrary value intentionally.

It is self-evident that the above two events is quite realistic. The occurrence of either one will render the smart card unusable, and U_i's all subsequent login requests will be denied unless she re-registers with S. Therefore, it is fair to say that Yang *et al.*'s scheme is vulnerable to a kind of smart card loss attack, which though may not as devastating as the offline password guessing attack. As also observed by Nam et al. [59], such a vulnerability cannot be easily eliminated. Possible solutions include the adoption of "a fuzzy verifier" introduced in Section 3 and the abandonment of the property of "local password change" (i.e., U_i steps back to change her password by interacting with the server S.).

Some remarks. We have analyzed more than one hundred recent smart-card-based password authentication schemes and find one third of them vulnerable to a similar attack as described above, which highlights the difficulties in designing such schemes. Particularly, all of the schemes based on conventional PAKE protocols are either insecure in password-change-activity or vulnerable to password guessing attack, i.e. none can fulfill the criterion C4, which highlights the differences from conventional PAKE protocols. Besides the scheme just described above, another typical example is Wang's two schemes, namely PSCAV and PSCA. Both PSCAV and PSCA is prone to offline password guessing attack, while PSCAV are based on an identity-based key agreement protocol from IEEE 1363.3 [22] and PSCA is based on the well-known HMQV [40].

B Cryptanalysis of Hsieh-Leu's scheme

In 2012, Hsieh and Leu [27] demonstrated several attacks against Hsiang-Shih's [26] smart-card-based password authentication scheme. To remedy the identified security flaws, they proposed an enhanced version over Hsiang-Shih's scheme [26] by "exploiting hash functions", and claimed that their improved scheme can withstand offline password guessing attack even if the sensitive parameters are extracted by the adversary. However, quite recently, Ma *et al.* [52] managed to prove that only symmetric cryptographic primitives (such as hash functions, symmetric encryption and MAC) are intrinsically insufficient for such schemes to resist against offline password guessing attack. As we will show in the following, under their non-tamper resistance assumption of the smart cards, this enhanced scheme is still vulnerable to an offline guessing attack, which is similar to the one that Hsiang-Shih's scheme suffers.

B.1 A brief review of Hsieh-Leu's scheme

In the following, we employ the notations listed in Table 1 and follow the descriptions in Hsieh-Leu's scheme [27] as closely as possible. This scheme is composed of four phases: registration, login, verification and password change.

Registration phase In this phase, the initial registration is different from the re-registration. Since the re-registration process has little relevance with our discussions, it omitted here. The process of the initial registration is depicted as follows.

1) U_i chooses a random number b and computes $h(b \oplus PW_i)$.

2) $U_i \Rightarrow S : ID_i, h(PW_i), h(b \oplus PW_i)$.

3) On receiving the login request, in the account database, server S creates an entry for U_i and stores $n = 0$ in this entry.

4) S computes $EID = (h(ID_i)||n)$, $P = h(EID \oplus x)$, $R = P \oplus h(b \oplus PW_i)$ and $V = h(h(PW_u) \oplus h(x))$. Then S stores V in the entry corresponding to U_i.

5) $S \Rightarrow U_i$: a smart card containing R and $h(\cdot)$.

6) On receiving the smart card, U_i inputs b into his smart card and does not need to remember b since then.

Login phase When user U_i wants to login to S, she inserts her smart card into the card reader and keys her ID_i with PW_i. The smart card performs the following steps:

1) The smart card computes $C_1 = R \oplus h(b \oplus PW_i)$ and $C_2 = h(C_1 \oplus T_i)$, where T_i denotes U_i's current timestamp.

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

Verification phase On receiving the login request from U_i, the remote server S and U_i's smart card perform the following steps:

1) If either ID_i or T_i is invalid or $T_s - T_i \leq 0$, S rejects U_i's login request. Otherwise, S computes $C_2' = h(h(EID \oplus x) \oplus T_i)$, and compares C_2' with the received C_2. If they are equal, S accepts U_i's login request and proceeds to compute $C_3 = h(h(EID \oplus x) \oplus h(T_s))$, where T_s denotes S's current timestamp. Otherwise, U_i's login request is rejected.

2) $S \rightarrow U : \{T_s, C_3\}$.

3) If either T_s is invalid or $T_s = T_i$, U_i terminates the session. Otherwise, U computes $C_3' = h(C_1 \oplus h(T_s))$, and compares the computed C_3' with the received C_3. If they are equal, U_i authenticates S successfully.

Password change phase When U_i wants to update her password, this phase is employed. Since this phase has little relevance with our discussions, it is omitted here.

B.2 Offline password guessing attack

Offline password guessing attack is the most damaging threat that a practical password-based protocol must be able to thwart [3]. Hsieh and Leu showed that Hsiang-Shih's scheme [26] is vulnerable to offline password guessing attack once the secret parameters stored in the stolen smart card are revealed by the adversary "by monitoring the power consumption or by analyzing the leaked information".

Now let's see how exactly the same attack could be successfully launched with Hsieh-Leu's own scheme in place. Suppose a legitimate user U_i's smart card is somehow (stolen or picked up) in the possession of an adversary $\mathcal{A}$, and the stored secret R and b can be revealed using side-channel attacks [55, 58]. With the previously intercepted authentication message $\{ID_i, C_2, T_i\}$ from the public channel, $\mathcal{A}$ can obtain U_i's password PW_i as follows:

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

Step 2. Computes $C_1^* = R \oplus h(b \oplus PW_i)$, where R, b is extracted from U_i's smart card.

Step 3. Computes $C_2^* = h(C_1^* \oplus T_i)$, where T_i is previously intercepted from the public channel.

Step 4. Verifies the correctness of PW_i^* by checking if the computed C_2^* is equal to the intercepted C_2.

Step 5. Repeats Step $1 \sim 4$ of this procedure until the correct value of PW_i is found.

Let $|\mathcal{D}_{pw}|$ denote the number of passwords in $\mathcal{D}_{pw}$. The time complexity of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{pw}| * (2T_H + 3T_X))$, where T_H is is the running time for Hash function and T_X is the running time for bitwise XOR operation. It is easy to see that, the time for $\mathcal{A}$ to recover U_i's password is a linear function of the number of passwords in the password space. In practice, the password space is very limited, e.g., $|\mathcal{D}| = 10^6$ [6, 19], and hence the above attack can be completed in polynomial time.

C Cryptanalysis of PSCAV

In SEC'12, Wang [84] observed that the previous papers in this area present attacks on protocols in previous papers and propose new protocols without proper security justification (or even a security model to fully identify the practical threats), which contributes to the main cause of the above failure. Accordingly, Wang presented three kinds of security models, namely Type I, II and III, and further proposed four concrete schemes, only two of which, i.e. PSCAb and PSCAV , are claimed to be secure under the harshest model, i.e. Type III security model. However, PSCAb requires Weil or Tate pairing operations to defend against offline guessing attack and may not be suitable for systems where pairing operations are considered to be too expensive or infeasible to implement. Moreover, PSCAb suffers from the well-known key escrow problem and lacks some desirable features such as local password update, repairability and user anonymity. As for PSCAV, in this paper, we will demonstrate that it still cannot achieve the claimed security goals and is vulnerable to an offline password guessing attack and other attacks under the Type III security model in the following.

C.1 A brief review of PSCAV

In this section, we firstly give a brief review of PSCAV and then present the attack. Here we just follow the original notations in [84] as closely as possible. Assume that the server has a

master secret β (β could be user specific also). For each user with password α, let the user specific generator be $g_{\mathcal{C}} = \mathcal{H}(\mathcal{C}, \alpha, \beta)$, the value $g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$ is stored in the smart card, where $\mathcal{H}_2$ is another independent hash function. The value $g_{\mathcal{C}} = \mathcal{H}(\mathcal{C}, \alpha, \beta)$ will be stored in the server's database for this user. The remaining of the protocol runs as follows:

1. The card selects random x and sends $R_A = g_{\mathcal{C}}^x$ to the server;
2. Server selects random y and sends $R_A = g_{\mathcal{C}}^y$ to the card;
3. The card computes $u = \mathcal{H}(\mathcal{C}, \mathcal{S}, R_A, R_B)$ and $sk = g_{\mathcal{C}}^{y(x+u)}$, where S is identity string of the server and $g_{\mathcal{C}} = \mathcal{D}_{\mathcal{H}_2(\alpha)}(g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)})$;
4. The card sends $C_{\mathcal{C}} = H(sk, \mathcal{C}, \mathcal{S}, R_A, R_B, 1)$ to the server;
5. After verifying that $C_{\mathcal{C}}$ is correct, server computes $u = \mathcal{H}(\mathcal{C}, \mathcal{S}, R_A, R_B)$, $sk = g_{\mathcal{C}}^{y(x+u\alpha)} = (g_{\mathcal{C}}^x)^y \cdot g_{\mathcal{C}}^{yu\alpha} = (R_A)^y \cdot g_{\mathcal{C}}^{yu\alpha}$, and sends $C_{\mathcal{S}} = \mathcal{H}(sk, \mathcal{S}, \mathcal{C}, R_B, R_A, 2)$ to the card.

The message flows of PSCAV are shown in the Fig.2. Since the session key sk is computed with the contribution of the password α by server S in the above Step 5, the password α (or the parameter g_C^α) is needed to be known by S. However, the original specification in [84] does not explicitly explain how can the server obtain the user's password α to compute sk in the above Step 5. We assume (suggest) g_C^α is also stored in the server's database, i.e. an entry $(\mathcal{C}, g_{\mathcal{C}}, g_C^\alpha)$ corresponding to user $\mathcal{C}$ is stored in the server's database.[5] This ambiguity does not affect our security analysis however.

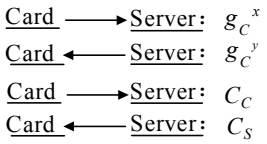

Fig. 2. Message flows of PSCAV

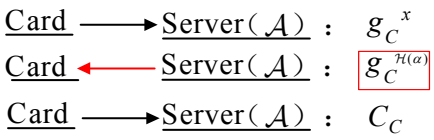

Fig. 3. Our attack

C.2 Offline password guessing attack

In [84], it is claimed that PSCAV is secure in Type III security model, i.e., the scheme can withstand offline password guessing attack even if C's smart card is compromised (i.e., the secret data stored in the card is revealed). In the following, we demonstrate that this is not true. Suppose an adversary $\mathcal{A}$ has compromised C's smart card and obtained the stored secret $g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$. The attack, summarized in Fig.3, can be carried out by $\mathcal{A}$ to obtain C's password α as follows:

Step 1. On intercepting $R_A = g_{\mathcal{C}}^x$ from client $\mathcal{C}$, $\mathcal{A}$ blocks it and sends $R_B = g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$ to the client on behalf of the server;
Step 2. On receiving the response $C_{\mathcal{C}}$, $\mathcal{A}$ computes $u = \mathcal{H}(\mathcal{C}, \mathcal{S}, R_A, R_B)$.
Step 3. Guesses the value of password α to be α^* from dictionary space $\mathcal{D}$.
Step 4. Computes $g_{\mathcal{C}}^* = \mathcal{D}_{\mathcal{H}_2(\alpha^*)}(g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)})$;
Step 5. Computes $sk^* = g_{\mathcal{C}}^{x\mathcal{H}_2(\alpha^*)} \cdot (g_{\mathcal{C}}^*)^{u\alpha^*\mathcal{H}_2(\alpha^*)}$
$= (R_A)^{\mathcal{H}_2(\alpha^*)} \cdot (g_{\mathcal{C}}^*)^{u\cdot\alpha^*\cdot\mathcal{H}_2(\alpha^*)}$;

[5] This ambiguity and our suggested remedy have been confirmed by the author of [84], and he earns our deep respect for his frankly and quickly acknowledgement.

Step 6. $\mathcal{A}$ computes $C_{\mathcal{C}}^{*} = H(sk^{*}, \mathcal{C}, \mathcal{S}, R_A, R_B, 1)$;
Step 7. Verifies the correctness of α^{*} by checking if the computed $C_{\mathcal{C}}^{*}$ is equal to the received $C_{\mathcal{C}}$;
Step 8. Repeats the above Steps 3-8 until the correct value of α is found.

As the size of the password dictionary, i.e. $|\mathcal{D}|$, is very limited in practice [6, 19, 38], e.g. $|\mathcal{D}| = 10^6$, $\mathcal{A}$ may recover the password in seconds on a PC by a single run of the PSCAV protocol.

C.3 Other security drawbacks

In many cases, it is of utmost importance to provide user anonymity so that the adversary cannot trace user activity. For example, in e-commerce applications, once the identity of the user is leaked, the sensitive information such as shopping patterns, individual preferences, etc., can be learnt and abused for marketing purposes [2]. In addition, the leakage of the user identity may also cause an unauthorized entity to track the user's login history and current location [71]. Therefore, user anonymity is a glamorous and important feature that a practical authentication scheme should achieve [51,54,69,72]. However, in PSCAV, the user's identity ID is transmitted in plain, which may leak the identity of the logging user once the login messages were eavesdropped. In other words, without employing any effort an adversary can recognize the particular transaction being performed by the user C. Moreover, the user's identity ID is static in all the login phases, which may facilitate the attacker to trace out the different login request messages belonging to the same user and to derive some information related to the user C. In a word, neither identity protection nor user un-traceability can be preserved in PSCAV.

Since user C's critical parameter is computed as $g_{\mathcal{C}} = \mathcal{H}(\mathcal{C}, \alpha, \beta)$, it is not easily repairable once $g_{\mathcal{C}}$ is leaked out, which can happen in various ways, such as a break of password just as stated in the previous section.

D Formal security analysis of our scheme

D.1 Proof of Theorem 1

Proof. In the proof below, we do not consider forward-secrecy for simplicity. We incrementally define a sequence of games starting at the real attack game G_0 and ending up with G_8. For each game G_n (n=0,1, ..., 8), we define the following events:

- $\mathbf{Succ_n}$ occurs if $\mathcal{A}$ correctly guesses the bit c involved in the Test-query.
- $\mathbf{AskPara_n}$ occurs if $\mathcal{A}$ correctly computes the parameter k by asking a hash query $\mathcal{H}_0$ on $b\|PW_i$ or $x\|ID_i\|T_{reg}$.
- $\mathbf{AskAuth_n}$ occurs if $\mathcal{A}$ correctly computes the parameter k and asks a hash query $\mathcal{H}_1$ (or $\mathcal{H}_2$) on $ID_i\|ID_S\|Y_1\|C_2\|k\|K$, where K is K_U or K_S.
- $\mathbf{AskH_n}$ occurs if the adversary asks a hash query $\mathcal{H}_i$ $(i = 1, 2, 3)$ on $ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K$, where K is K_U or K_S.

Game $\mathbf{G_0}$: This game corresponds to the real attack, in the random oracle model. By definition, we have

$$\mathrm{Adv}_{\mathcal{P}}^{\mathrm{ake}}(\mathcal{A}) = 2\mathrm{Pr}[\mathrm{Succ}_0] - 1. \quad (1)$$

Game $\mathbf{G_1}$: In this game, we simulate the hash oracles $\mathcal{H}_i$(i=0,1,2 and 3, but also four additional hash functions $\mathcal{H}_i'$ that will appear in Game $\mathbf{G_7}$) as usual by maintaining a hash list $\Lambda_{\mathcal{H}}$(and another list $\Lambda_{\mathcal{A}}$ containing the hash-queries asked by the adversary itself). We also simulate all

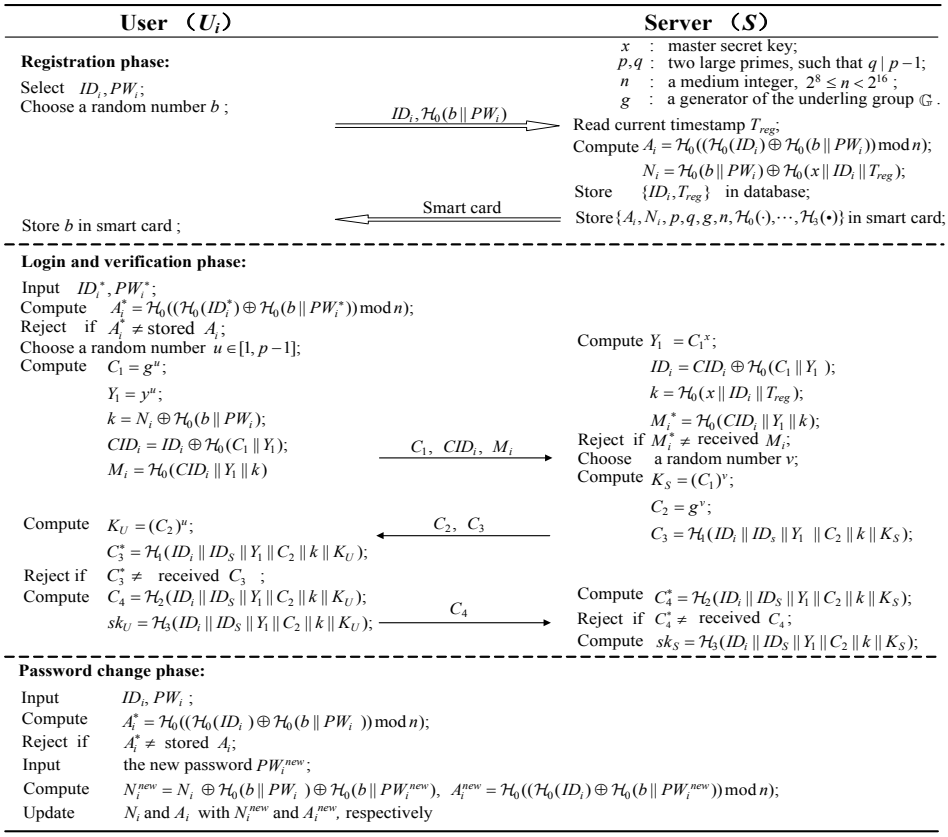

Fig. 4. The proposed scheme

the instances, as the real players would do, for the Send-queries and for the Execute, Reveal, Corrupt and Test-queries (see Figure 5).

From this simulation, one can easily see that this game is perfectly indistinguishable from the real attack. Hence,

$$\left|\Pr[\text{Succ}_1] - \Pr[\text{Succ}_0]\right| = 0 \tag{2}$$

Game $\mathbf{G_2}$: For an easier analysis, in this game, we simulate all oracles as in game $\mathbf{G_1}$ except that we cancel games in which some (unlikely) collisions appear:

- collisions on the partial transcripts $((C_1, M_i, CID_i), (C_2, C_3), C_4)$. Note that transcripts involve at least one honest party, and thus one of C_1 or C_2 is truly uniformly distributed;
- collisions on the output of hash queries.

Both probabilities are bounded by the birthday paradox:

$$\left|\Pr[\text{Succ}_2] - \Pr[\text{Succ}_1]\right| \le \frac{(q_{send} + q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} \tag{3}$$

where $l = min\{l_i\}, i = 0, 1, 2, 3$.

Game $\mathbf{G_3}$: We define game $\mathbf{G_3}$ by aborting the game wherein the adversary may have lucky in guessing the correct authenticator C_3 or C_4 (that is, without asking the corresponding hash

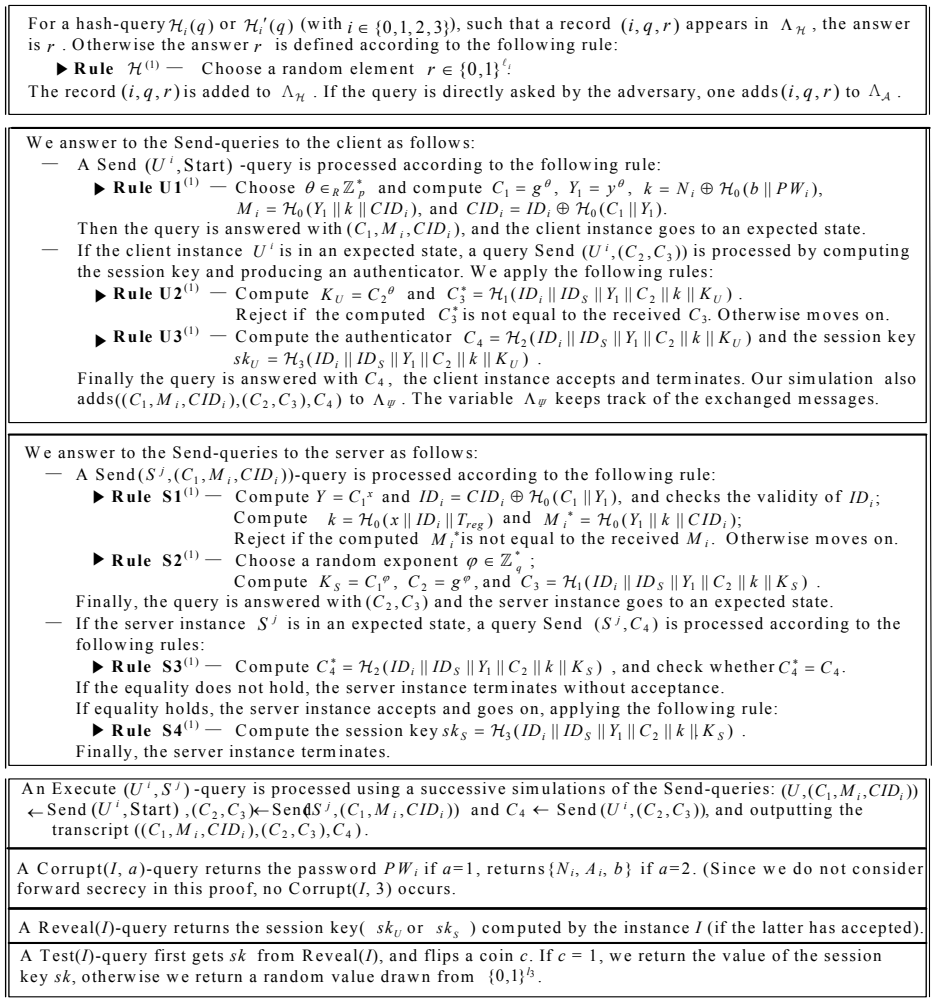

Fig. 5. Simulation of the queries in our scheme

query $\mathcal{H}_1$ or $\mathcal{H}_2$). Since C_1 and C_2 did not appear in a previous session (since the Game $\mathbf{G_2}$), this happens only if the authenticator C_3 (or C_4) had been correctly guessed by $\mathcal{A}$ without asking $\mathcal{H}_1$ (or $\mathcal{H}_2$):

$$\left|\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2]\right| \leq \frac{q_{send}}{2^l} \tag{4}$$

Game $\mathbf{G_4}$: We define game $\mathbf{G_4}$ by aborting the game wherein the adversary may have lucky in guessing the correct parameter k (i.e., without asking the corresponding query). We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U3$^{(4)}$** – Look for a record $(0, *\|ID_i\|*, k)$ in $\Lambda_{\mathcal{A}}$. If such a record does not exist, we abort the game. Otherwise, compute $C_4 = \mathcal{H}_2(ID_i\|ID_S\|Y_1\| C_2\|k\|K_U)$ and the session key $sk_U = \mathcal{H}_3(ID_i\|ID_S\|Y_1\|C_2\|k\|K_U)$.
- **Rule S3$^{(4)}$** – computes $C_4^* = \mathcal{H}_2(ID_i \| ID_S \| Y_1 \| C_2 \| k \| K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record

$(0, *, P_i)$ in $\Lambda_{\mathcal{A}}$ and a record $((C_1, M_i, CID_i), (C_2, C_3), C_4)$ in Λ_{Ψ}. If such records do not exist, we abort the game.

Since C_1 and C_2 did not appear in a previous session (since the Game $\mathbf{G_2}$), this happens only if the parameter k had been correctly guessed by the adversary without asking $\mathcal{H}_0$:

$$\left|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_3]\right| \leq \frac{q_{send}}{2^{l_0}} \tag{5}$$

Game $\mathbf{G_5}$: We define this game by aborting the game wherein the adversary may have computed the correct parameter k and impersonate as a client or server. We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U2$^{(5)}$** – Look for a record $(0, *\|ID_i\|*, k)$ in $\Lambda_{\mathcal{A}}$. If such a record exists, we abort the game. Otherwise, compute $K_U = (C_2)^u \bmod p$, $C_3^* = \mathcal{H}_1(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_U)$, and compare C_3^* with the received C_3. If the equality doesnot hold, terminate without acceptance. Otherwise, move on.
- **Rule S3$^{(5)}$** – computes $C_4^* = \mathcal{H}_2(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record $(0, *, P_i)$ in $\Lambda_{\mathcal{A}}$ and a record $((C_1, M_i, CID_i), (C_2, C_3), C_4)$ in Λ_{Ψ}. If such records exist, we abort the game.

The two games $\mathbf{G_5}$ and $\mathbf{G_4}$ are perfectly indistinguishable unless the event $\mathbf{AskPara_5}$ occurs:

$$\left|\Pr[\text{Succ}_5] - \Pr[\text{Succ}_4]\right| \leq \Pr[\text{AskPara}_5] \tag{6}$$

To obtain the upper bound of $\Pr[\text{AskPara}_5]$, the parameter k is assumed to be correctly computed by $\mathcal{A}$ in all the ensuing games.

Game $\mathbf{G_6}$: We define this game by aborting the game wherein the adversary may have computed the correct authenticator C_3 or C_4 (that is, by asking the corresponding hash query $\mathcal{H}_1$ or $\mathcal{H}_2$) and impersonate as a client or server. We reach this aim by modifying the way the participants process the queries. We use the rule as follows:

- **Rule U3$^{(6)}$** – Check if $(1, ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel *, C_3) \in \Lambda_{\mathcal{A}}$. If it holds, we abort the game. Otherwise, the user goes on to compute C_4 and sk_U.
- **Rule S3$^{(6)}$** – computes $C_4^* = \mathcal{H}_2(ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel K_S)$ and then checks if C_4^* equals the received value of C_4. If this verification holds, the server looks for a record $(0, *, P_i)$ or $(2, ID_i \parallel ID_S \parallel Y_1 \parallel C_2 \parallel k \parallel *, C_4)$ in $\Lambda_{\mathcal{A}}$, and a record $((C_1, CID_i), (C_2, C_3), C_4)$ in Λ_{Ψ}. If such records exist, we abort the game.

The two games $\mathbf{G_6}$ and $\mathbf{G_5}$ are perfectly indistinguishable unless event $\mathbf{AskAuth_6}$ occurs:

$$\left|\Pr[\text{Succ}_6] - \Pr[\text{Succ}_5]\right| \leq \Pr[\text{AskAuth}_6] \tag{7}$$

$$\left|\Pr[\text{AskPara}_6] - \Pr[\text{AskPara}_5]\right| \leq \Pr[\text{AskAuth}_6] \tag{8}$$

Game $\mathbf{G_7}$: In this game, we replace the random oracles $\mathcal{H}_i$ with the private oracles $\mathcal{H}_i'(i = 1, 2, 3)$:

$$\begin{aligned} &C_3 = \mathcal{H}_1'(ID_i \parallel ID_S \parallel C_1 \parallel C_2);\\ &C_4 = \mathcal{H}_2'(ID_i \parallel ID_S \parallel C_1 \parallel C_2);\\ &sk_U = sk_S = \mathcal{H}_3'(ID_i \parallel ID_S \parallel C_1 \parallel C_2) \end{aligned}$$

As a result, the values of C_3, C_4, sk_U, sk_S are completely independent from k, K_U and K_S. $\mathbf{G_7}$ and $\mathbf{G_6}$ are indistinguishable unless the event $\mathbf{AskH_7}$ occurs:

$$\left|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_6]\right| \leq \Pr[\text{AskH}_7] \tag{9}$$

$$\left|\Pr[\text{AskPara}_7] - \Pr[\text{AskPara}_6]\right| \leq \Pr[\text{AskH}_7] \tag{10}$$

$$\left|\Pr[\text{AskAuth}_7] - \Pr[\text{AskAuth}_6]\right| \leq \Pr[\text{AskH}_7] \tag{11}$$

Lemma 1. *The probabilities of the events* $\mathbf{Succ_7}$ *and* $\mathbf{AskPara_7}$ *in this game can be upbounded by the following values:*

$$\left|\Pr[\text{Succ}_7]\right| = \frac{1}{2} \tag{12}$$

$$\left|\Pr[\text{AskPara}_7]\right| \leq \frac{q_{send}}{|\mathcal{D}|} + \frac{q_{send}}{2^{l_0}} \tag{13}$$

Proof. In the game $\mathbf{G_7}$, the session keys are computed with private hash oracle unknown to $\mathcal{A}$, and thus $\Pr[\text{Succ}_7] = \frac{1}{2}$.

Let us denote by $R(U)$ the set of (C_2, C_3) received by a client instance, and by $R(S)$ the set of C_4 used by a server instance. Since we have avoided the cases where $\mathcal{A}$ have been lucky in guessing k, $\mathcal{A}$ can correctly computed k with the help of either a Corrupt($I = U^i$, 1)-query or a Corrupt($I = U^i$, 2)-query, the probability of which is denoted by $\Pr[\text{AskPara}_7\text{WithCorr}_1]$ and $\Pr[\text{AskPara}_7\text{WithCorr}_2]$, respectively. From an information theoretical point of view, since we have avoided collisions in the Game $\mathbf{G_2}$,

$$\begin{aligned}\left|\Pr[\text{AskPara}_7\text{WithCorr}_1]\right| &= \Pr_{pw}[\exists C_3 \in R(U), (1, ID_i \| ID_S \| Y_1 \| C_2 \| k \| *, C_3) \in \Lambda_{\mathcal{A}}] \\ &\quad + \Pr_{pw}[\exists C_4 \in R(S), (0, *, P_i) \in \Lambda_{\mathcal{A}}] \\ &\leq \tfrac{\#R(S)+\#R(U)}{|\mathcal{D}|} = \tfrac{q_{send}}{|\mathcal{D}|}\end{aligned} \tag{14}$$

$$\left|\Pr[\text{AskPara}_7\text{WithCorr}_2]\right| \leq \tfrac{q_{send}}{2^{l_0}} \tag{15}$$

Game $\mathbf{G_8}$: In this game, we simulate the executions using the random self-reducibility of the Diffie-Hellman problem, given one CDH instance (A, B). Note that, we do not need to know the values of θ and φ, since the values of K_U and K_S are no longer needed to compute the authenticators or the session keys:

- **Rule $\mathbf{U1^{(8)}}$** – chooses a random number $\alpha \in Z_p^*$ and computes $C_1 = A^\alpha \bmod p$, $Y_1 = y^\alpha \bmod p$, $k = \mathcal{H}_0(x \parallel ID_i \parallel T_{reg}) = N_i \oplus \mathcal{H}_0(b \parallel PW_i)$, $CID_i = ID_i \oplus \mathcal{H}_0(C_1 \parallel Y_1)$ and $M_i = \mathcal{H}_0(Y_1 \parallel k \parallel CID_i)$. Also add the record (C_1, α) in Λ_A.
- **Rule $\mathbf{S2^{(8)}}$** – chooses a random number $\beta \in Z_p^*$, and computes $C_2 = B^\beta$ and $C_3 = \mathcal{H}_1'(ID_i \parallel ID_S \parallel C_1 \parallel C_2)$. Also adds the record (C_2, β) in Λ_B.

$$\Pr[\text{AskH}_7] = \Pr[\text{AskH}_8] \tag{16}$$

Remember that $\mathbf{AskH_8}$ means that the adversary $\mathcal{A}$ had queried the random oracles $\mathcal{H}_i (i = 1, 2, 3)$ on $(ID_i \parallel ID_S \| Y_1 \| C_2 \| * \| CDH(C_1, C_2))$. By picking randomly in the $\Lambda_{\mathcal{A}}$-list we can get the Diffie-Hellman secret value with probability $\frac{1}{q_h}$. This is a triple $(C_1, C_2, \text{CDH}(C_1, C_2))$. We can then simply look in the lists Λ_A and Λ_B to find the values α and β such that $C_1 = A^\alpha$ and $C_2 = B^\beta$:

$$\text{CDH}(C_1, C_2) = \text{CDH}(A^\alpha, B^\beta) = \text{CDH}(A, B)^{\alpha\beta},$$

and thus:

$$\big|\Pr[\text{AskH}_8]\big| \leq q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t') \tag{17}$$

where $t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$.

Conclusion of the proof: By combining all the above equations, one gets the announced result. Firstly, from Eqs.(1)-(5) we get:

$$\big|\Pr[\text{Succ}_4] - \Pr[\text{Succ}_0]\big| \leq \tfrac{(q_{send}+q_{exe})^2}{2p} + \tfrac{q_h^2}{2^{l+1}} + \tfrac{q_{send}}{2^l}.$$

Secondly, from Eqs.(6)-(9) we get:

$$\big|\Pr[\text{Succ}_7] - \Pr[\text{Succ}_4]\big| \leq \Pr[\text{AskPara}_5] + \Pr[\text{AskAuth}_6] + \Pr[\text{AskH}_7].$$

Thirdly, from definition we know:

$$\Pr[\text{AskAuth}_6] \leq \Pr[\text{AskH}_6].$$

Finally, based on Eqs.(10)-(17) we get:

$$\begin{aligned}\text{Adv}_{\mathcal{P}}^{\text{ake}}(\mathcal{A}) &= 2\Pr[\text{Succ}_7] - 1 + 2(\Pr[\text{Succ}_0] - \Pr[\text{Succ}_7]) \\ &\leq \tfrac{2q_{send}}{|\mathcal{D}|} + 12q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t + (q_{send} + q_{exe} + 1) \cdot \tau_e) \\ &+ \tfrac{q_h^2+6q_{send}}{2^l} + \tfrac{(q_{send}+q_{exe})^2}{p},\end{aligned}$$

where $t' \leq t + (q_{send} + q_{exe} + 1) \cdot \tau_e$ and $l = min\{l_i\}, i = 0, 1, 2, 3$.

D.2 Proof of Theorem 2

Proof. The proof is similar to that of Theorem 1.

Firstly, we define an additional event:

- $\mathbf{Auth_n}$ occurs if $\mathcal{A}$ correctly guesses the authenticator C_3 or C_4 that will be accepted by the corresponding party and that has been built by the adversary herself in game $\mathbf{G_n}, n = 0, 1, \ldots, 7$.

Thus, we define

$$\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) = \Pr[\text{Auth}_0]$$

Secondly, we use the same sequence of games presented in the previous section, and extend Eqs. (1)-(7) to obtain:

$$\begin{aligned}&\big|\Pr[\text{Auth}_1] - \Pr[\text{Auth}_0]\big| = 0 \\ &\big|\Pr[\text{Auth}_2] - \Pr[\text{Auth}_1]\big| \leq \frac{(q_{send} + q_{exe})^2}{2p} + \frac{q_h^2}{2^{l+1}} \\ &\big|\Pr[\text{Auth}_3] - \Pr[\text{Auth}_2]\big| \leq \frac{q_{send}}{2^l} \\ &\big|\Pr[\text{Auth}_4] - \Pr[\text{Auth}_3]\big| \leq \frac{q_{send}}{2^l} \\ &\big|\Pr[\text{Auth}_5] - \Pr[\text{Auth}_4]\big| \leq \Pr[\text{AskPara}_5] \\ &\big|\Pr[\text{Auth}_6] - \Pr[\text{Auth}_5]\big| \leq \Pr[\text{AskAuth}_6] \leq 2\Pr[\text{AskH}_7] \\ &\Pr[\text{Auth}_7] = \Pr[\text{Auth}_6] = 0\end{aligned}$$

Thus, we have

$$\begin{aligned}\text{Adv}_{\mathcal{P}}^{\text{auth}}(\mathcal{A}) \leq\ & \tfrac{q_{send}}{|\mathcal{D}|} + 5q_h \text{Adv}_{\mathcal{P}}^{\text{CDH}}(t + (q_{send} + q_{exe} + 1) \cdot \tau_e) \\ &+ \tfrac{q_h^2+6q_{send}}{2^{l+1}} + \tfrac{(q_{send}+q_{exe})^2}{2p}.\end{aligned}$$