Offline Dictionary Attack on Password Authentication Schemes using Smart Cards*

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

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

Abstract. The design of secure and efficient smart-card-based password authentication schemes remains a challenging problem today despite two decades of intensive research in the security community, and the current crux lies in how to achieve truly two-factor security even if the smart cards can be tampered. In this paper, we analyze two recent proposals in this area, namely, Hsieh-Leu's scheme and Wang's PSCAV scheme. We demonstrate that, under their non-tamper-resistance assumption of the smart cards, both schemes are still prone to offline dictionary attack, in which an attacker can obtain the victim's password when getting temporary access to the victim's smart card. This indicates that compromising a single factor (i.e., the smart card) of these two schemes leads to the downfall of both factors (i.e., both the smart card and the password), thereby invalidating their claim of preserving two-factor security. Remarkably, our attack on the latter protocol, which is not captured in Wang's original protocol security model, reveals a new and realistic attacking scenario and gives rise to the strongest adversary model so far (Note that Wang's PSCAV scheme is secure within its own but weak security model). In addition, we make the first attempt to explain why smart cards, instead of common cheap storage devices (e.g., USB sticks), are preferred in most two-factor authentication schemes for security-critical applications.

Keywords: Password authentication; Offline dictionary attack; Smart card; Common memory device; Non-tamper resistant.

1 Introduction

Back in 1992, Bellovin and Merritt [5] showed how two parties, who only share a low-entropy password and communicate over a public network, can authenticate each other and agree on a cryptographically strong session key to secure their subsequent communications. Their work, known as encrypted key exchange, is a great success in protecting poorly-chosen passwords from the notorious offline dictionary attacks and thus confirms the feasibility of using password-only protocols to establish virtually secure channels over public networks, which is one of the main practical applications of cryptography. Due to the practical significance of password-based authentication, Bellovin-Merritt's seminal work has been followed by a number of remarkable proposals (e.g., [1,8,9]) with various levels of security and complexity.

* This is a full version of the paper that appears in the proceedings of the 16th Information Security Conference (ISC 2013), November 13-15, 2013, Dallas, Texas, LNCS, Springer–Verlag, pp.1-16.

While password authentication protocols are well suited to applications with moderate security demands, they are inadequate for use in security-critical applications such as e-banking and e-health. Since password-based protocols generally require the server to store and manage a sensitive password-related file, a compromise of this file will lead to an exposure of the passwords of all the registered users, resulting in the downfall of the entire system. With the prevalence of zero-day attacks [6], these days it is no news to see the headlines about catastrophic leakages of tens of millions of passwords [14, 17]. It is due to this inherent limitation that two-factor authentication schemes[1] are introduced to enhance the systems' security and privacy. Owing to its portability, simplicity and cryptographic capability, smart-card-based password authentication has become one of the most effective, prevalent and promising two-factor mechanisms.

Since Chang and Wu [10] developed the first smart-card-based password authentication scheme in 1991, there have been ample (in the hundreds) of this type of schemes proposed [16, 21, 24, 27, 28, 31, 32, 49, 57, 61]. Unfortunately, as stated in [35, 46], although there has been no lack of literature, it remains an immature area – all existing schemes are far from ideal and each has been shown to be either insecure or short of important features. For an intuitive grasp, we summarize the "break-fix-break-fix" history of this area in Fig. 1. Note that many other important schemes cannot be included into the picture only due to space constraints.

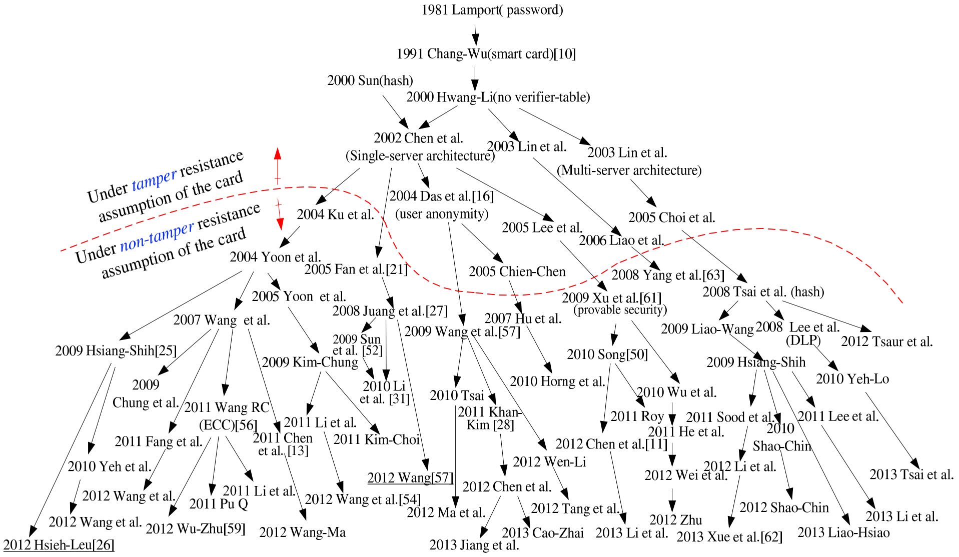

Fig. 1. A brief history of smart-card-based password authentication

Motivations. The past thirty years of research on password-only protocols have proved that it is incredibly difficult to get a single-factor protocol right [44, 64], while the past twenty years of "break-fix-break-fix" cycle of smart-card-based password protocols have manifested designing a two-factor scheme can only be harder [34, 35, 41]. It remains an open problem to develop an efficient and secure two-factor protocol that can meet all the security goals

[1] Note that "protocol" and "scheme" will be used interchangeably through-out this paper.

(see Section 2 of [58]) and preserve all the desirable features such as user anonymity and repairability (see [35] for a comprehensive list of features).

We have analyzed more than one hundred and fifty smart-card-based password protocols and observe that, as with the domain of password authentication, offline dictionary attack is still the most prominent issue in two-factor authentication. This attack against many of previous protocols has emphasized the need for rigorous proofs of security in a formal, well-defined model. On the other hand, the practicality of a formal model largely depends on whether it "accurately captures the realistic capabilities of the adversary" [18]. As stated by Alfred Menezes [37, 38], although many formal security definitions "have an appealing logical elegance and nicely reflect certain notions of security, they fail to take into account many types of attacks and do not provide a comprehensive model of adversarial behavior", and "the old-fashioned cryptanalysis continues to play an important role in establishing confidence in the security of a cryptographic system". All this and the continuous failures in designing a practical two-factor scheme outline the need for exploring the adversarial behaviors and for revealing the underlying subtleties by cryptanalysis.

Recent studies have demonstrated that the secret parameters stored in common smart cards can be extracted by side-channel attacks such as power analysis [29, 39, 40] or reverse engineering techniques [36, 43]. Even though the card manufacturers may have considered the risks of side-channel attacks and provided countermeasures to cope with the problem, how much confidence can one have that these countermeasures residing in the card are still effective after three years of the card production and operation? Considering this, since 2004 most schemes have preferred to use a non-tamper-resistant smart card (see the dash line of Fig.1). This brings forth a question: While non-tamper resistance assumption has been made about smart cards (which means the core feature of the smart cards is lost), why not just use cheap memory devices (e.g., USB sticks) instead? As far as we know, little attention has been paid to this interesting (and fundamental) question.

Our contributions. This paper examines the security of two recently proposed schemes, namely Hsieh-Leu's scheme [26] and Wang's PSCAV scheme [58]. These two schemes are the foremost ones and claimed to be secure against various known attacks. However, this work invalidates their claims by demonstrating that, under their assumption of the capabilities of the adversary, both schemes are vulnerable to the offline dictionary attack. This indicates that none of them can achieve the "precious" two-factor security.

Interestingly, our attack on Wang's PSCAV scheme [58] highlights a new attacking scenario: Firstly, an attacker gets temporary access to the victim's card and extracts its security data (hereafter we say this card is exposed); Secondly, she returns the exposed card without awareness of the victim; Finally, she performs malicious attacks when the victim uses this exposed card. *This new attacking scenario has already given rise to the strongest adversary model so far* (see the "Returned stolen token" Section of [65]).

In addition, we take the first step toward giving an explanation to the rather confusing question – why smart cards, instead of common cheap storage devices (e.g., USB sticks and flash-memory cards), are preferred in most two-factor authentication schemes for security-critical applications, even if smart cards can be tampered?

The remainder of this paper is organized as follows: Section 2 sketches the system architecture and elaborates on the adversary model; Then, Hsieh-Leu's scheme is reviewed and analyzed in Section 3; In Section 4, we review Wang's scheme and show its weakness; The conclusion is drawn in Section 5.

2 System architecture and adversary models

2.1 System architecture of two-factor authentication

In this paper, we mainly focus on the most general case of two-factor authentication where the communication parties only involve a single server and a set of users, i.e. the traditional client/server architecture, as illustrated in Fig.2.1. It is not difficult to see that our results in this paper can be applied to more complex architectures where more than one server are involved, such as the multi-server authentication environments [62], the mobile network roaming environments [49] and the hierarchical wireless sensor networks [15].

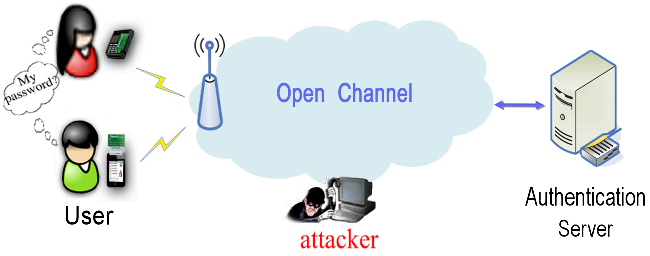

Fig. 2. Smart-card-based password authentication

In this sort of schemes, firstly the user chooses an identity (often as well as a password) and registers at the server; The server returns the user a smart card storing some security parameters. After registration, whenever the user wants to login to the server, she inserts the card into a terminal and enters her password. Then the card constructs a login request and sends it to the server. Upon receiving the request, the server checks its validity and will offer the requested service if the verification holds. Generally, a session key is established for securing the subsequent data communications. More sophisticated schemes also achieves mutual authentication, i.e. the client is also convinced that the server on the other end is authentic. What a truly two-factor protocol can guarantee is that, only the user who is in possession of both the smart card and the corresponding password can pass the verification of the server. This implies that a compromise of either factor will pose no danger to the security of a truly two-factor protocol.

2.2 Adversary models for smart-card-based authentication and for common-memory-based authentication

In this section, we attempt to take the initial step to justify the use of smart cards rather than common memory devices in security-critical applications. Firstly, we explicitly define the practical capabilities that an attacker may have in the smart-card-based authentication environment and in the memory-device-based authentication environment, respectively. Then, we investigate into the advantages and disadvantages of these two kinds of authentication.

Two kinds of mobile devices. A smart card is an integrated circuit card with a processor for executing applications and a memory, coupled to the processor, for storing multiple applications. This kind of device has been widely used for various security-critical applications ranging from online-banking over digital rights management (DRM) to stream media (e.g., Pay-TV). For example, a HiPerSmart-P9SC648 smart card [48] from the 32-bit

HiPerSmart^{TM} family is of the RISC MIPS32 architecture, with a maximum clock speed of 36MHz, a 512 Kbyte Flash, a 142 Kbyte EEPROM and a 16 Kbyte RAM. In the current market, such a smart card is priced at \$3.0 ∼ 3.5. In contrast, a common USB memory stick is a data storage device that includes the flash memory with an integrated Universal Serial Bus (USB) interface, and the typical cost of an 1 GB USB stick is \$1.3 ∼ 1.5 [22]. Since USB memory sticks[2] are not equipped with micro-processors, they cannot execute cryptographic operations as opposed to smart cards, and thus the operations have to be performed on the user terminal (e.g., PCs and PDAs).

As discussed in the previous section, it is prudent and reasonable to take into consideration the side-channel attacks [39, 40, 43, 53] when designing a smart-card-based two-factor authentication scheme. In other words, the secret data stored in the card memory are assumed to be extractable when the smart card is in the hands of an attacker. On the other hand, in the past it was just the tamper-resistant feature that makes smart cards prevail over other cheap (but non-tamper-resistant) memory devices. *Now that smart cards can be tampered, why we do not choose cheap USB memory sticks instead of expensive smart cards?* Or equally, what's the rationale under these propositions [21,31,52,54,56,59,63] that endeavor to construct two-factor authentication schemes using non-tamper resistant smart cards rather than memory sticks? To the best of our knowledge, until now, little attention has been given to this question.

Two kinds of adversary models. To identify the differences in security provisions offered by two-factor authentication schemes using these two different devices, we need to discuss the realistic capabilities that the attacker may have under these two different authentication environments. On the basis of the studies [52,58,61,63], the following assumptions are made on the capabilities of the adversary $\mathcal{M}$ in the smart-card-based environment:

S(i) $\mathcal{M}$ can fully control the communication channel between the user and the server. In other words, she can inject, modify, block, and delete messages exchanged in the channel at will. This assumption is consistent with the Dolev-Yao model;

S(ii) $\mathcal{M}$ is able to get access to the smart card and may compromise the user's smart card through side-channel attacks when getting access to the smart card for a relatively long period of time(e.g., a few hours) [58,63];

S(iii) $\mathcal{M}$ may comprise the user's password (e.g., by shoulder-surfing or malicious card reader [20,33]).

S(iv) $\mathcal{M}$ is *not able to* extract the sensitive information on the smart card while intercepting the victim's password by using a malicious card reader, since the user is on the scene and the time is not sufficient for launching a side-channel attack [4,58];

S(v) $\mathcal{M}$ is not allowed to first compromise the user's password and then compromise the smart card [54,58,61]. Clearly, if $\mathcal{M}$ has compromised both factors, there is no way to prevent $\mathcal{M}$ from impersonating the user, since these two factors together precisely identify the user. It is a trivial case.

The above Assumptions $S(i) \sim S(iv)$ have been made in most recent schemes and their reasonableness is quite evident. For a detailed justification, readers are referred to [55]. It is worth noting that, Assumptions $S(ii)$ and $S(iv)$ together imply that the common non-tamper-resistance assumption made about the smart cards is *conditional*. In particular, it is Assumption $S(iv)$ that makes it possible for the smart-card-based schemes to be adopted

[2] Hereafter, we use "USB sticks" and "common memory devices" interchangeably. In this paper, we do not consider the hybrid devices such as the Trust Extension Devices [2].

in completely hostile environments, yet most studies [12,21,26,28,51,54,61,63] (except few ones [55,58]) do not make this assumption clear and just implicitly rely on it. Failing to catch this subtlety may cause great misconceptions and lead to curious situations as it did in the works [11,45,46], which will be discussed later in this section.

Regarding USB-stick-based schemes, Assumption $S(iv)$ will not be valid, because it is not difficult for a malware to copy all the contents in the USB memory stick within only seconds, even if the user appears on the scene. Nevertheless, the other four assumptions do roughly hold for common-memory-based environment:

M(i) $\mathcal{M}$ can fully control the communication channel between the user and the server. In other words, she can inject, modify, block, and delete messages exchanged in the channel at will. This assumption is consistent with the Dolev-Yao model;

M(ii) $\mathcal{M}$ is able to compromise the user's memory device through malware within a short time period (e.g., in a few seconds);

M(iii) $\mathcal{M}$ may comprise the user's password (e.g., by shoulder-surfing or social engineering);

M(iv) $\mathcal{M}$ is *able to* extract the sensitive data on the memory device while intercepting the password that the user input by using malwares;

M(v) $\mathcal{M}$ is not allowed to first compromise the user's password and then compromise the user's memory device. Clearly, if $\mathcal{M}$ compromises both factors, there is no way to prevent $\mathcal{M}$ from impersonating the user, since these two factors together precisely identify the user. It is a trivial case.

Justification for using smart cards. Having examining the differences of adversary models between the smart card scenario and the common memory device scenario, we proceed to look into the rationales underlying the wide use of smart cards rather than common memory devices in two-factor authentication schemes.

Recently, with the popularity of mobile devices, a few studies [11,45] advocated the use of common memory devices instead of smart cards to construct two-factor schemes, and claimed that their schemes can "enjoy all the advantages of authentication schemes using smart cards" [45]. However, such a claim is a bit optimistic. According to the above adversary model for common-memory-device environment, such a claim holds only when the scheme is adopted in a trusted user terminal (otherwise, the malicious terminal could just intercept the password and copy the content of the memory device, and with no doubt the attacker is able to impersonate the victim in future). Smart-card-based schemes, in contrast, do not subject to this restriction. For example, under the five assumptions $S(i) \sim S(v)$, Wang et al. [55] manage to construct a smart-card-based scheme with provable security, and this scheme can well operate in a hostile user terminal.

In contrast to the optimistic view of [11,45], the work in [46] pessimistically stated that, if the smart card are assumed to be non-tamper-resistant, then "it is no better than a passive token". Consequently, smart cards are abandoned in their choice and static clonable tokens are in place, and a software-only two-factor scheme is proposed. Obviously, according to our above analysis, such a software-only scheme can never achieve the same level of security as compared to smart-card-based schemes. Nevertheless, this scheme may be suitable for applications where costs gain more concerns than security. The authors in [63] also explicitly advocate that they "do not make assumption on the existence of any special security features supported by the smart-cards" and "simply consider a smart-card to be a memory card with an embedded micro-processor for performing required operations specified in a scheme." It is not difficult to see that, in the light of their statements, their proposed smart-card-based scheme [63] can never achieve the claimed two-factor security.

In security-critical applications, user terminals are often the targets of attackers and may be infected with viruses, trojans and malwares, only two-factor authentication schemes using smart cards (which is though only conditionally tamper-proof) are suitable for such environments, common-memory-device-based two-factor schemes cannot "enjoy all the advantages of authentication schemes using smart cards". This explicates why most studies adhere to use smart cards rather than common memory devices when designing two-factor schemes (which is made for security-critical applications), even if it is supposed the data stored in the card memory can be extracted.

3 Cryptanalysis of Hsieh-Leu's scheme

In 2012, Hsieh and Leu [26] demonstrated several attacks against Hsiang-Shih's [25] smart-card-based password authentication scheme. To remedy the identified security flaws, they proposed an enhanced version over Hsiang-Shih's scheme [25] by "exploiting hash functions", and claimed that their improved scheme can withstand offline dictionary attack even if the sensitive parameters are extracted by the adversary. However, as we will show in the following, under their non-tamper-resistance assumption of the smart cards, Hsieh-Leu's scheme is still vulnerable to offline dictionary attack, which is similar to the one that Hsiang-Shih's scheme suffers.

3.1 A brief review of Hsieh-Leu's scheme

For ease of presentation, we employ some intuitive notations as listed in Table 1 and will follow the descriptions in Hsieh-Leu's scheme [26] as closely as possible. This scheme is composed of four phases: registration, login, verification and password change.

Table 1. Notations and abbreviations

Symbol	Description
U_i	i^{th} user
S	remote server
$\mathcal{M}$	malicious attacker
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
$h(\cdot)$	collision free one-way hash function
$A \rightarrow B : C$	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

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 is 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 $V = h(h(PW_i) \oplus h(x))$, and 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_i 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.

3.2 Offline dictionary attack

Offline dictionary attack is the most damaging threat that a practical password-based protocol must be able to guard against [1,3,5]. Hsieh and Leu showed that Hsiang-Shih's scheme [25] is vulnerable to offline dictionary attack once the secret parameters stored in the victim's 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 user U_i's smart card is somehow (stolen or picked up) in the possession of an adversary $\mathcal{M}$, and the parameters R and b can be revealed using side-channel attacks [36, 39]. With the previously intercepted authentication transcripts $\{ID_i, C_2, T_i\}$ from the public channel, $\mathcal{M}$ can obtain U_i's password PW_i as follows:

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

Step 2. Computes $C_1^* = R \oplus h(b \oplus PW_i)$, 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.

Our attack shows that once the smart-card factor is compromised, the corresponding password factor can be offline guessed and hence the entire system collapses. This indicates that Hsieh-Leu's scheme is intrinsically not a two-factor scheme and is as insecure as the original scheme (i.e., Hsiang-Shih's scheme [25]). This also corroborates the "public-key principle" [34] that, under the non-tamper resistance assumption of the smart cards, only symmetric-key techniques (such as Hash, block cipher) are inherently unable to resist against offline dictionary attack.

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 the running time for Hash function and T_X the running time for bitwise XOR operation. It is easy to see that, the time for $\mathcal{M}$ to recover U_i's password is a linear function of the number of passwords in the password space. And hence our attack is quite effective. For an intuitive grasp of the effectiveness of this attack (and the following attack on PSCAV), we further obtain the running time (see Table 2) for the related operations on common Laptop PCs by using the publicly-available, multi-precision integer and rational arithmetic C/C++ library MIRACL [47]. In practice, the password space is very limited, e.g., $|\mathcal{D}_{pw}| \leq 10^6$ [7,19,60], and it follows that the above attack can be completed in seconds on a common PC.

Table 2. Computation evaluation of related operations on common Laptop PCs

Experimental Platform (common PCs)	Exponentiation T_E ($\|n\|$=1024)	Symmetric encryption T_S (AES-128)	Hash operation T_H (SHA-1)	Other lightweight operations(e.g.,XOR)
Intel T5870 2.00 GHz	10.526 ms	2.012 μs	2.437 μs	0.011 μs
Intel E5500 2.80 GHz	7.716 ms	0.530 μs	0.756 μs	0.009 μs
Intel i5-3210 2.50 GHz	4.390 ms	0.415 μs	1.132 μs	0.008 μs

The above attack can be generalized as follows: with the security parameters stored in the smart card and the transcripts intercepted during the previous login session(s), the attacker can repeatedly guess the victim's password via an offline automated program. This attack strategy is not new. Actually, it is the common "Waterloo" of many broken schemes [12,13,25,28,50]. This attack scenario (adversary behavior) has been captured in several two-factor security models [58,59,61]. Yet, the following attacker is still at large.

4 Cryptanalysis of PSCAV from SEC 2012

In SEC'12, Wang [58] observed that the previous papers in this area present attacks on protocols in earlier works and put forward new proposals without proper security justification (let alone a security model to fully identify the practical threats), which constitutes the main cause of the long-standing failure. Accordingly, Wang presented three kinds of security models, namely Type I, II and III. In the Type III model, i.e. the harshest model, mainly three assumptions are made:

(1) an adversary $\mathcal{M}$ is allowed to have full control of the communication channel between the user and the server;

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

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

Note that, the above Assumption 1 is consistent with **S(i)**, and Assumption 2 is consistent with **S(ii)**∼**S(v)**(see Sec.2.2). As for Assumption 3, its opposite is implicitly made in most of previous schemes as well as the model introduced in Section 2.2. Apparently, a scheme which is secure in Type III shall also be secure in a model only with Assumptions 1 and 2. To the best of our knowledge, the Type III model is the strongest model that has ever been proposed for smart-card-based password authentication so far.

Wang [58] further proposed four schemes, only two of which, i.e. PSCAb and PSCAV, are claimed to be secure under the Type III model. However, PSCAb requires Weil or Tate pairing operations to defend against offline dictionary 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 repairability, user anonymity and local password update. As for PSCAV, in this paper, we will demonstrate that it is susceptible to offline dictionary attack under Assumptions 1 and 2 (or equally, **S(i)**∼**S(v)**) plus a new (but realistic) assumption – the attacker can return a victim's exposed card without detection.

4.1 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 [58] as closely as possible. Assume that the server has a master secret β (β could be user specific also). For each user (or called client) C with identity $\mathcal{C}$ and password α, let the user specific generator be $g_{\mathcal{C}} = \mathcal{H}(\mathcal{C}, \alpha, \beta)$, the value $g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}(= \mathcal{E}_{\mathcal{H}_2(\alpha)}(g_{\mathcal{C}})$, and thus $g_{\mathcal{C}} = \mathscr{D}_{\mathcal{H}_2(\alpha)}(g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}))$ is stored in the smart card, where $\mathcal{H}$ and $\mathcal{H}_2$ are two independent hash functions, and $\mathcal{E}/\mathscr{D}$ stand for symmetric encryption/decryption (see Section 3.2 of [58]). 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, computes $g_{\mathcal{C}} = \mathscr{D}_{\mathcal{H}_2(\alpha)}(g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)})$ 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;
4) The card sends $C_{\mathcal{C}} = \mathcal{H}(sk, \mathcal{C}, \mathcal{S}, R_A, R_B, 1)$ to the server;
5) After verifying that $C_{\mathcal{C}}$ is correct, the 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.3. 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 [58] 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.[3] This ambiguity does not affect our security analysis however.

4.2 Offline dictionary attacks

Suppose an adversary $\mathcal{M}$ has got temporary access to the client C's smart card and obtained the stored secret $g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$. Then $\mathcal{M}$ sends back the card without awareness of the victim C.

[3] This ambiguity and our suggested remedy have been confirmed by the author of [58], and he earns our deep respect for his frankly and quickly acknowledgement.

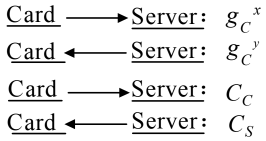

Fig. 3. Message flows of PSCAV

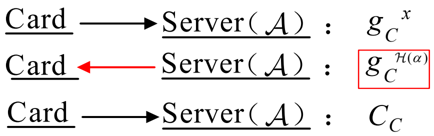
Card
Server($\mathcal{A}$) : $g_C^{\,x}$
Card
Server($\mathcal{A}$) : $g_C^{\,\mathcal{H}(\alpha)}$
Card
Server($\mathcal{A}$) : C_C

Fig. 4. Our attack

Once client C uses the exposed smart card, the attacker can impersonate as the server to interact with C and to learn C's password. The attack, as summarized in Fig.4, can be carried out by $\mathcal{M}$ as follows:

Step 1. On intercepting $R_A = g_{\mathcal{C}}^x$ from client C, $\mathcal{M}$ blocks it and sends $R_B = g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$ to the client on behalf of the server, where $g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)}$ is extracted from C's card;

Step 2. On receiving the response $C_{\mathcal{C}}$, $\mathcal{M}$ 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 $\mathcal{D}_{pw}$.

Step 4. Computes $g_{\mathcal{C}}^* = \mathscr{D}_{\mathcal{H}_2(\alpha^*)}(g_{\mathcal{C}}^{\mathcal{H}_2(\alpha)})$;

Step 5. Computes $sk^* = g_{\mathcal{C}}^{x\mathcal{H}_2(\alpha^*)} \cdot (g_{\mathcal{C}}^*)^{u\alpha^*\mathcal{H}_2(\alpha^*)}$
$= (R_A)^{\mathcal{H}_2(\alpha^*)} \cdot (g_{\mathcal{C}}^*)^{u\cdot\alpha^*\cdot\mathcal{H}_2(\alpha^*)}$;

Step 6. $\mathcal{M}$ computes $C_{\mathcal{C}}^* = \mathcal{H}(sk^*, \mathcal{C}, \mathcal{S}, R_A, R_B, 1)$;

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.

The time complexity of the above attack is $\mathcal{O}(|\mathcal{D}_{pw}| * (3T_E + T_S + 3T_H))$. As the size of the password dictionary, i.e. $|\mathcal{D}_{pw}|$, is very limited in practice [7,19,60], e.g. $|\mathcal{D}_{pw}| \le 10^6$, the above attack can be completed in polynomial time. Further considering the experimental timings listed in Table 2, $\mathcal{M}$ may recover the password in minutes on a PC by a single run of PSCAV.

Interestingly, our attack on Wang's PSCAV scheme [58] highlights *a new attacking scenario*: Firstly, an attacker gets temporary access to the victim's card and extracts its security parameters; Secondly, she sends back the exposed card without awareness of the victim; Finally, she performs malicious attacks when the victim uses this exposed card. Note that this attacking scenario is quite realistic. For example, an employee accidentally leaves her bank card on her desk after work, the attacker picks this card and performs the side-channel attacks herself (or with recourse to professional labs) in the evening and puts it back before the victim comes to work the next morning. The victim will find no abnormality and use this card as usual. Unfortunately, once this card is put to use, the corresponding password may be leaked, while the above procedure well serves to illustrate how the password can be leaked to an attacker. As reported in [42], "agencies are interested in quickly accessing someone's room, install some bug in the her mobile device and then return it without detection". This also confirms the practicality of our attack.

Wang's PSCAV scheme is secure in their security model yet vulnerable to our new attacking strategy. Since the identified attacking scenario is realistically oriented towards a serious threat, it deserves special attention when defining the underlying security model for smart-card-based password authentication. This once again suggests that, a good security model is not one that denies the capabilities of the attacker but rather one designed to capture the attacker's practical abilities as comprehensively as possible, and the powers not allowed to the attacker are those that would allow her to trivially break any of this type of schemes [23, 30]. Fortunately, *it has already given rise to the strongest adversary model so far*: Just two weeks ago, a new security model named Type III-r was developed in [65].

5 Conclusion

Understanding security failures of cryptographic protocols is the key to both patching existing protocols and designing future schemes. In this paper, we have shown that Hsieh-Leu's scheme and Wang's PSCAV scheme suffer from the offline dictionary attack under two different attacking strategies, which reveals the challenges in constructing a practical two-factor authentication scheme. Remarkably, our attack on Wang's PSCAV scheme highlights a new realistic attack scenario and thus uncovers a new behavior of the attacker – returning the exposed smart card without awareness of the victim. As for future work, we are considering designing a practical scheme that can survive in the Type III-r model.

Acknowledgment. We would like to thank the anonymous reviewers for their constructive comments on improving this study and the shepherd of the paper for his efforts. And we are particularly grateful to Prof. David Naccache for generous advice and for referring us to [42]. This research was partially supported by the National Natural Science Foundation of China (NSFC) under No. 61170001 and No. 61170243.

References

1. Abdalla, M., Chevassut, O., Pointcheval, D.: One-time verifier-based encrypted key exchange. In: Vaudenay, S. (ed.) PKC 2005, LNCS, vol. 3386, pp. 47–64. Springer Berlin Heidelberg (2005)
2. Asokan, N., Ekberg, J.E., Kostiainen, K.: The untapped potential of trusted execution environments on mobile devices. In: Sadeghi, A.R. (ed.) Financial Cryptography and Data Security, LNCS, vol. 7859, pp. 293–294. Springer-Verlag (2013), full version available at `http://www.cs.helsinki.fi/group/secures/mobiletee-may28.pdf`
3. Bao, F.: Security analysis of a password authenticated key exchange protocol. In: Boyd, C., Mao, W. (eds.) ISC 2003, LNCS, vol. 2851, pp. 208–217. Springer/Berlin Heidelberg (2003)
4. Barenghi, A., Breveglieri, L., Koren, I., Naccache, D.: Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. Proceedings of the IEEE 100(11), 3056–3076 (2012)
5. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proc. IEEE S&P 1992. pp. 72–84. IEEE (1992)
6. Bilge, L., Dumitras, T.: Before we knew it: an empirical study of zero-day attacks in the real world. In: Proc. ACM CCS 2012. pp. 833–844. ACM (2012)
7. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proc. IEEE Security & Privacy 2012. pp. 538–552. IEEE Computer Society (2012)
8. Boyd, C., Montague, P., Nguyen, K.: Elliptic curve based password authenticated key exchange protocols. In: Varadharajan, V., Mu, Y. (eds.) Proc. of 28th Australasian Conference on Information Security and Privacy (ACISP 2001), LNCS, vol. 2119, pp. 487–501. Springer-Verlag (2001)
9. Bresson, E., Chevassut, O., Pointcheval, D.: New security results on encrypted key exchange. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004, LNCS, vol. 2947, pp. 145–158. Springer-Verlag (2004)
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. Chen, B.L., Kuo, W.C., Wuu, L.C.: A secure password-based remote user authentication scheme without smart cards. Information Technology And Control 41(1), 53–59 (2012)
12. Chen, B., Kuo, W., Wuu, L.: Robust smart-card-based remote user password authentication scheme. Int. J. Commun. Syst. (2012), doi:`http://dx.doi.org/10.1002/dac.2368`
13. Chen, T.H., Hsiang, H.C., Shih, W.K.: Security enhancement on an improvement on two remote user authentication schemes using smart cards. Future Gener. Comput. Syst. 27(4), 377–380 (2011)
14. Constantin, L.: Sony stresses that PSN passwords were hashed (May 2011), `http://news.softpedia.com/news/Sony-Stresses-PSN-Passwords-Were-Hashed-198218.shtml`
15. Das, M.L.: Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications 8(3), 1086–1090 (2009)
16. Das, M., Saxena, A., Gulati, V.: A dynamic id-based remote user authentication scheme. IEEE Transactions on Consumer Electronics 50(2), 629–631 (2004)
17. Dazzlepod Inc.: CSDN cleartext passwords (Mar 2013), online news, Available at `http://dazzlepod.com/csdn/`
18. Degabriele, J.P., Paterson, K., Watson, G.: Provable security in the real world. IEEE Security & Privacy 9(3), 33–41 (2011)
19. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: Proc. INFOCOM 2010. pp. 1–9. IEEE (2010)
20. Drimer, S., Murdoch, S.J., Anderson, R.: Thinking inside the box: system-level failures of tamper proofing. In: Proc. IEEE Security & Privacy 2008. pp. 281–295. IEEE (2008)
21. Fan, C., Chan, Y., Zhang, Z.: Robust remote authentication scheme with smart cards. Computers & Security 24(8), 619–628 (2005)
22. Focus Technology Co., Ltd.: Prices for 1GB Usb Flash Drive (April 2013), `http://www.made-in-china.com/products-search/hot-china-products/1gb_Usb_Flash_Drive.html`
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 Berlin / Heidelberg (2010)
24. He, D., Ma, M., Zhang, Y., Chen, C., Bu, J.: A strong user authentication scheme with smart cards for wireless communications. Comput. Commun. 34(3), 367–374 (2011)
25. Hsiang, H., Shih, W.: Weaknesses and improvements of the yoon-ryu-yoo remote user authentication scheme using smart cards. Comput. Communi. 32(4), 649–652 (2009)
26. Hsieh, W., Leu, J.: Exploiting hash functions to intensify the remote user authentication scheme. Computers & Security 31(6), 791–798 (2012)
27. Juang, W.S., Chen, S.T., Liaw, H.T.: Robust and efficient password-authenticated key agreement using smart cards. IEEE Trans. Ind. Electron. 55(6), 2551–2556 (2008)
28. Khan, M., Kim, S.: Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme'. Comput. Commun. 34(3), 305–309 (2011)
29. Kim, T.H., Kim, C., Park, I.: Side channel analysis attacks using am demodulation on commercial smart cards with seed. J. Syst. Soft. 85(12), 2899 – 2908 (2012)
30. Krawczyk, H.: HMQV: A high-performance secure diffie-hellman protocol. In: Shoup, V. (ed.) CRYPTO 2005, LNCS, vol. 3621, pp. 546–566. Springer-Verlag (2005)
31. Li, X., Qiu, W., Zheng, D., Chen, K., Li, J.: Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. IEEE Trans. Ind. Electron. 57(2), 793–800 (2010)
32. Li, X., Xiong, Y., Ma, J., Wang, W.: An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications 35(2), 763–769 (2012)
33. Long, J.: No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing. Syngress (2011)
34. Ma, C.G., Wang, D., Zhao, S.D.: Security flaws in two improved remote user authentication schemes using smart cards. Int. J. Commun. Syst. (2013), doi: `http://dx.doi.org/10.1002/dac.2468`

35. 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)
36. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks: Revealing the secrets of smart cards. Springer-Verlag (2007)
37. Menezes, A.: Another look at HMQV. Journal of Mathematical Cryptology 1(1), 47–64 (2007)
38. Menezes, A.: Another look at provable security. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 8–8. Springer Berlin / Heidelberg (2012), availe at http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf
39. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Trans. Comput. 51(5), 541–552 (2002)
40. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the vulnerability of FPGA bitstream encryption against power analysis attacks: extracting keys from xilinx Virtex-II FPGAs. In: Proc. ACM CCS 2011. pp. 111–124. ACM, New York, NY, USA (2011)
41. Murdoch, S.J., Drimer, S., Anderson, R., Bond, M.: Chip and pin is broken. In: Proc. IEEE Security & Privacy 2010. pp. 433–446. IEEE Computer Society (2010)
42. Naccache, D.: National security, forensics and mobile communications. In: Won, D., Kim, S. (eds.) ICISC 2005, LNCS, vol. 3935, p. 1. Springer-Verlag (2005)
43. Nohl, K., Evans, D., Starbug, S., Plötz, H.: Reverse-engineering a cryptographic rfid tag. In: Proc. USENIX Security 2008. pp. 185–193. USENIX Association (2008)
44. Pointcheval, D.: Password-based authenticated key exchange. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012, LNCS, vol. 7293, pp. 390–397. Springer Berlin / Heidelberg (2012)
45. Rhee, H.S., Kwon, J.O., Lee, D.H.: A remote user authentication scheme without using smart cards. Computer Standards & Interfaces 31(1), 6–13 (2009)
46. Scott, M.: Replacing username/password with software-only two-factor authentication. Tech. rep., Cryptology ePrint Archive, Report 2012/148 (2012), http://eprint.iacr.org/2012/148.pdf
47. Shamus Software Ltd.: Miracl library (May 2013), http://www.shamus.ie/index.php?page=home
48. Smart Card Alliance: Philips advances smart card security for mobile applications (April 2013), http://www.ceic-cn.org/files/NXP2006zcard.pdf
49. Son, K., Han, D., Won, D.: A privacy-protecting authentication scheme for roaming services with smart cards. IEICE Trans. Commun. 95(5), 1819–1821 (2012)
50. Song, R.: Advanced smart card based password authentication protocol. Computer Standards & Interfaces 32(5), 321–325 (2010)
51. Sun, D.Z., Huai, J., Sun, J., Li, J.: Cryptanalysis of a mutual authentication scheme based on nonce and smart cards. Computer Communications 32(6), 1015–1017 (2009)
52. Sun, D.Z., Huai, J.P., Sun, J.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)
53. 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 (2011)
54. Wang, D., Ma, C.G., Wu, P.: Secure password-based remote user authentication scheme with non-tamper resistant smart cards. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds.) DBSec 2012, LNCS, vol. 7371, pp. 114–121. Springer Berlin / Heidelberg (2012)
55. Wang, D., Ma, C., Wang, P., Chen, Z.: Robust smart card based password authentication scheme against smart card security breach. Cryptology ePrint Archive, Report 2012/439 (2012), http://eprint.iacr.org/2012/439.pdf
56. 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)

57. 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)
58. 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)
59. 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)
60. Wu, T.: A real-world analysis of kerberos password security. In: Proc. NDSS 1999. pp. 13–22. Internet Society (1999)
61. Xu, J., Zhu, W., Feng, D.: An improved smart card based password authentication scheme with provable security. Comput. Stand. & Inter. 31(4), 723–728 (2009)
62. Xue, K., Hong, P., Ma, C.: A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J. Comput. System Sci. (2013), doi: http://dx.doi.org/10.1016/j.jcss.2013.07.004
63. Yang, G., Wong, D., Wang, H., Deng, X.: Two-factor mutual authentication based on smart cards and passwords. J. Comput. Syst. Sci. 74(7), 1160–1172 (2008)
64. 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)
65. Zhao, Z., Wang, Y.G.: Secure Communication and Authentication Against Off-line Dictionary Attacks in Smart Grid Systems (2013), http://coitweb.uncc.edu/~yonwang/papers/smartgridfull.pdf