Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing

Yaosheng Shen[1,3], Ding Wang[2,3], and Ping Wang[1,3,4⋆]

[1] School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China
[2] School of EECS, Peking University, Beijing 100871, China
[3] National Engineering Research Center for Software Engineering, Beijing, China
[4] School of Software and Microelectronics, Peking University, Beijing 100260, China
{ysshen; wangdingg; pwang}@pku.edu.cn

Abstract. Investigating the security pitfalls of cryptographic protocols is crucial to understanding how to improve security. At ICCCS'17, Wu and Xu proposed an efficient smart-card-based password authentication scheme to cope with the vulnerabilities in Jiang *et al.*'s scheme. However, in this paper, we reveal that Wu-Xu's scheme actually is subject to critical security defects, such as offline password guessing attack and replay attack. Besides security, user friendly is also another great concern. In 2017, Roy et al. found that in most previous two-factor schemes a user has to manage different credentials for different services, and further suggested a user-friendly scheme which is claimed to be suitable for multi-server architecture and robust against various attacks. In this work, we show that Roy et al.'s scheme cannot achieve truly two-factor security and is of poor scalability. Our results invalidate any use of the scrutinized schemes for cloud computing environments.

Keywords: Cloud computing; Two-factor authentication; Offline password guessing attack; User untraceability.

1 Introduction

With the emerging paradigm of cloud computing, various services are provided over the cloud. As cloud-based services can be accessed anytime and anywhere just with a connection to the Internet, it is important to protect users and cloud servers from severe security threats, such as fraudulence, eavesdropping and falsification, posed either by external attackers or malicious internal entities. To guarantee that the resources and services can only be accessed by legitimate parties, user authentication plays an important part in securing electronic transactions by acquiring collaborative evidence. In 2011, Hao *et al.* [5] suggested the first two-factor authentication protocol which combines passwords and smart cards for the cloud computing environments. This initial work has brought about a number of enhanced proposals [9, 16, 24, 25, 27] with each different in terms of security, anonymity, usability and efficiency.

⋆ Corresponding author.

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

Without loss of generality, we consider the most common client-server architecture in which two participants (i.e. a server S and a user U) get involved in two-factor authentication. User U holds a memorable password and a smart card stored with some initial security parameters, and server S only needs to keep some secret key material of the system. Since there is no need to keep a table with password-related verification information on the server side, the server is free from the threat of password dataset leaks and ameliorated from the burden of maintaining a large password dataset. This feature that makes this type of schemes rather desirable, considering that there are incessant leakages of password databases from large websites [1]. The most important security goal of this kind of schemes is the so-called "two-factor security" [23]. This security concept essentially means that only the user who has the smart card as well as knows the right password can be verified by the server.

Most of existing two-factor schemes (e.g, [9, 16, 25]) for cloud computing are built on the basis of generic two-factor schemes like [24, 29]. Nevertheless, past research [7, 12, 13, 21] have, again and again, proved that designing a "password+smart-card" two-factor authentication scheme that can attain "two-factor security" is a tough task. In 2009, Xu *et al.* [28] developed such a two-factor authentication scheme relying on the intractability of computational Deffie-Hellman problem, and stated that their scheme is able to support "two-factor security" under the hypothesis that smart cards can be tampered. In addition, their scheme was "proved secure" in the random oracle model. Later on, Sood *et al.* [19] however illustrated that Xu *et al.*'s scheme cannot resist against user impersonation attack if the parameters kept in the smart cards can be extracted, invalidating Xu *et al.*'s claim of ensuring "two-factor security". In 2010, Song [18] independently found this severe flaw in Xu *et al.*'s scheme. Furthermore, Song presented an improvement to counter the problem emerged in Xu *et al.*'s scheme.

In 2012, Chen *et al.* [3] pointed out that various security drawbacks still existed in both Sood et al.'s [19] and Song's [18] schemes. More specifically, Sood *et al.*'s scheme is unable to withstand server impersonation attack and Song's scheme is vulnerable to offline password guessing attack, in case the attacker can obtain those sensitive information kept in the smart card. Chen *et al.* [3] also put forward an improvement and argued that their improvement is robust under the condition that the sensitive data in smart card has been revealed by the attacker. It should be noted that, recent rapid developments in side-channel attacks have proved that the sensitive information kept in general commercial smart cards could be extracted by power analysis or reverse engineering [11, 15]. Based on a weak yet realistic assumption, Chen *et al.*'s scheme [3] appears very practical.

However, soon after Chen *et al.*'s scheme [3] was presented, Ma *et al.* [13] figured out that it is susceptible to exactly the same problem (i.e., prone to offline password guessing attack and no supply of forward secrecy) with the original scheme (i.e., Song's scheme [18]). Based on their past experience of protocol design and analysis, for the first time Ma *et al.* [13] suggested three generic principles for designing a secure and efficient two-factor protocol, namely,

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

the public-key principle, the forward secrecy principle and the security-usability balance principle. Nevertheless, none of the two-factor authentication protocols mentioned above can satisfy all these three design principles and moreover, as we illustrate, all the schemes studied in this work fail to comply with at least one of these principles.

In 2017, Wu and Xu [26] also observed that previous schemes (e.g., [3,18]) are prone to various security loopholes (e.g., user impersonation attack and insider attack), and also developed an enhanced scheme. Wu-Xu argued that their new scheme not only eradicates the security pitfalls being overlooked in previous schemes but also maintains strengths of previous schemes. Notwithstanding their claims, we will show that this scheme still has serval serious defects be overlooked: (1) It cannot withstand offline password guessing attack once the parameters in smart card can be extracted out, which means that the primary goal of "truly two-factor security" cannot be satisfied with; (2) It is subject to replay attack; (3) It ensures no timely typo detection.

More recently, Roy *et al.* [17] found that, in Tsai-Lo's scheme [20], a user has to manage different credentials for different services, and further suggested a user-friendly scheme which is claimed to be suitable for multi-server architecture and robust against various attacks. Therefore, this protocol shows a good application potential in multi-server cloud computing environments. In this work, we show that Roy et al.'s scheme cannot provide truly two-factor security and user untraceability. We observe that the first failure of Roy *et al.*'s scheme is due in large part to the non-compliance with Ma et al.'s [13] security-usability tradeoff principle. Our attacks highlights the necessity of being aware of basic protocol design principles.

The remainder of the paper is organized as follows: We review Wu-Xu's scheme in Section 2, and describe its security loopholes in Section 3; Roy et al.'s scheme are presented in Section 4 and cryptanalyzed in Section 5; Finally, we conclude the paper in Section 6.

2 Cryptanalysis of Wu-Xu's scheme

In this section, we briefly review the chaotic-map based authentication scheme for cloud computing proposed by Wu and Xu [26] in ICCCS 2017. Their scheme consists of four phases: initialization, registration, authentication and password change. For better presentation, we will follow the notations in Wu-Xu's scheme as closely as possible and list the notations in Table 1.

2.1 Registration Phase

At the beginning, the server S creates a random positive number $s \in Z_p^*$ and a symmetric key cryptosystem with $E_k(.)$ and $D_k(.)$, then chooses a secret key x, and two one-way hash functions $h(.)$ and $h_1(.)$.

Step R1. U_i chooses her identity ID_i, PW_i and b_i, then computes $HPW_i = h(PW_i \parallel b_i)$

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

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 an open channel from A to B
$A \Rightarrow B : C$	message C is transferred through a secure channel from A to B

Step R2. $U_i \Rightarrow S : \{ID_i, HPW_i\}$.
Step R3. S selects a random integer r_i, computes $IM_i = E_x((ID_i \oplus r_i) \| r_i)$ and $B_1 = h(ID_i \| HPW_i) \oplus HPW_i \oplus h(x \| IM_i)$, and stores IM_i, B_1, $h(.)$, $E_k(.)/D_k(.)$, s and p into the smart card.
Step R4. $S \Rightarrow U_i$: A smart card containing $IM_i, B_1, E_k(.)/D_k(.), h(.), s, p$.
Step R5. U_i computes $B_2 = h(ID_i \| PW_i) \oplus b_i$ and stores B_2 into the card.

2.2 Login and Mutual Authentication Phase

When wanting to login, U_i performs as follows:

Step 1. U_i inserts the smart card into card reader and inputs ID_i and PW_i.
Step 2. Smart card computes $b_i' = B_2 \oplus h(ID_i \| PW_i)$.
Step 3. Smart card generates two random integers $u \in [1, p+1]$ and r_u and computes $HPW_i = h(PW_i \| b_i')$, $C_1 = T_u(s)$, $C_2 = B_1 \oplus h(ID_i \| HPW_i)$, $C_3 = h(ID_i \| C_1 \| C_2 \| r_u)$, $C_4 = C_2 \oplus C_3$, $C_5 = EC_3(ID_i \| C_1 \| r_u)$.
Step 4. Smart card$\rightarrow S : \{C_4, IM_i, C_5\}$.
Step 5. On receiving the message from U_i, S decrypts IM_i and gets ID_i' and r_i', then computes $C_2' = h(x \| IM_i)$, $C_3' = C_4 \oplus C_2'$.
Step 6. S decrypts C_5 to obtain ID_i'', C_1', and r_u', then checks if $ID_i' \stackrel{?}{=} ID_i''$ and $C_3' \stackrel{?}{=} h(ID_i' \| C_1' \| C_2' \| r_u')$. If both verifications are correct, S proceeds. Otherwise, the login request is interrupted.
Step 7. S prefers three random numbers $v \in [1, p+1]$, r_s and r_i^{new}, computes $IM_i' = E_x((ID_i' \oplus r_i^{new}) \| r_i^{new})$, $C_6 = T_v(s)$, $sk_s = T_v(C_1')$, $C_7 = h(x \| IM_i')$, $C_8 = h_1(ID_i' \| IM_i' \| C_1' \| C_2' \| C_6 \| C_7 \| sk_s \| r_s)$, $C_9 = C_2' \oplus C_8$, $C_{10} = E_{C_8}(IM_i' \| C_6 \| C_7 \| r_s)$, where sk_s is server-side session key.
Step 8. $S \rightarrow U_i : \{C_9, C_{10}\}$.
Step 9. Smart card calculates $C_8' = C_9 \oplus C_2$ and obtains IM_i'', C_6', C_7', r_s' by decrypting C_{10}. Smart card further computes $sk_u = T_u(C_6')$ and

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

checks $C_8' \stackrel{?}{=} h_1(ID_i \parallel IM_i'' \parallel C_1 \parallel C_2 \parallel C_6' \parallel C_7' \parallel sk_u \parallel r_s')$. If they are unequal, U_i aborts the protocol. Otherwise, smart card computes $B_1' = C_2 \oplus C_7' \oplus B_1$ and replaces (B_1, IM_i) with (B_1', IM_i'').

3 Flaws in Wu-Xu's scheme

In this section, the security loopholes of Wu-Xu's scheme [26] will be pointed out. More specifically, it cannot resist offline password guessing attack and suffers from replay attack, which make this scheme unpractical for real use. Before giving the detailed security analysis, we first define the various adversary models for smart-card-based password authentication.

3.1 Adversary models

To analyze the security provisions of password-based authentication schemes with smart cards, generally three assumptions about the attacker's capabilities are made since the landmark work of Yang et al. [29], and we summarize them as follows:

Assumption 1. The malicious attacker $\mathcal{M}$ is able to eavesdrop, delete, insert, modify or block any transcripts communicated in the public channel. That is to say, the communication channel between the common users and the server can be completely manipulated by $\mathcal{M}$. This well complies with the standard adversary model that is widely accepted for distributed computing [4];

Assumption 2. The malicious attacker $\mathcal{M}$ can somehow got the victim user's smart card and use side-channel attack techniques to acquire sensitive security parameters from the card memory, which is realistic according to the recent research advancements in side-channel attacks [11, 15];

Assumption 3. User's password space is very constrained and the malicious attacker $\mathcal{M}$ can offline enumerate it. To be user-friendly, most protocols (e.g., the ones in [10, 24, 29]) enable the users to select their own password at will in the initial process of registration or later process of password change. Because human beings are incapable of memorizing random strings, instead they are likely to choose passwords that relate to their personal lives or short strings for convenience. As a result, these human generated passwords often are very weak and belong to a small dictionary.

Obviously, if both *Assumption 2* and *Assumption 3* hold simultaneously, then an attacker (without any other assumptions/abilities) can successfully impersonate a victim user and any scheme is trivially insecure. Therefore, it is widely regarded that attackers should not be granted to obtain a victim user's smart card as well as his password [8, 24, 29].

In [26], Wu and Xu reported that their scheme is secure under the above three assumptions. For example, they stated that their scheme can resist offline password guessing attacks even when the security parameters in smart card have been extracted. However, contrary to their claims, we will illustrate that this scheme is still prone to offline guessing as well as other pitfalls. Based on

the above listed assumptions, we cryptanalyze the security provisions of Wu-Xu's scheme in the following, and assume that $\mathcal{M}$ can extract the secret data $\{B_1, B_2, h(\cdot), E_k(\cdot), D_k(\cdot)\}$ stored in U_i's smart card, and $\mathcal{M}$ can also eavesdrop the messages $\{C_4, IM_i, C_5\}$ exchanged between the parties.

3.2 Offline password guessing attack

It is known that password-based authentication schemes are apt to be subject to two kinds of attacks regarding guessing [7, 23], i.e., offline password guessing and online password guessing, due to the limited size of the password space. Among them, the online guessing can be relatively easily detected due to the abnormal number of login requests issued by the attacker against the victim account within a short duration, and thus it can be countered by rate-limiting [2]. In contrast, offline password guessing attack cannot be easily detected. In this attack, the attacker $\mathcal{M}$ first attempts to look for some pieces of information that can be exploited as the comparison target of his password guesses, and then locally determines the exactly correct password by repeatedly testing all the candidates. Since this attack is executed without online communication with the server, there is no means for the server to detect and thwart. Consequently, offline password guessing is considered to be a more serious threat.

Wu and Xu [26], claimed that an attacker is unable to, in an offline manner, determine a user's password even if the sensitive data B_1 has been revealed from user's smart card. However, the following attacking procedure illustrates that this claim is not tenable. Suppose that U_i's smart card is lost/stolen and the attacker $\mathcal{M}$ obtains it. Then $\mathcal{M}$ extracts the content $\{B_1, B_2, h(\cdot)\}$ by using the methods introduced in [15]. The following procedure describes our proposed offline password guessing attack:

Step 1. $\mathcal{M}$ choose a pair (ID_i^*, PW_i^*) from the identity space $\mathcal{D}_{ID}$ and password space $\mathcal{D}_{PW}$, respectively.
Step 2. $\mathcal{M}$ computes $b_i^* = B_2 \oplus h(ID_i^* \| PW_i^*)$, where B_2 is revealed from U_i's smart card.
Step 3. $\mathcal{M}$ computes $C_2^* = B_1 \oplus h(ID_i^* \| h(PW_i^* \| b_i^*)) \oplus h(PW_i^* \| b_i^*)$, where B_1 is extracted from U_i's smart card.
Step 4. $\mathcal{M}$ calculates $C_3^* = C_2^* \oplus C_4$, where C_4 is eavesdropped from the open channel.
Step 5. $\mathcal{M}$ decrypts C_5 by using C_3^* to obtain $ID_i' \oplus C_1' \oplus r_u'$, where C_5 is eavesdropped from the open channel.
Step 6. $\mathcal{M}$ calculates $C_3^* = h(ID_i^* \| C_1' \| C_2^* \| r_u')$;
Step 7. $\mathcal{M}$ examines the authenticity of (ID_i^*, PW_i^*) pair by verifying if $C_4 \stackrel{?}{=} C_2^* \oplus C_3^*$.
Step 8. $\mathcal{M}$ return to Step 1 of this procedure until the right pair of (ID_i, PW_i) is obtained or all pairs in $\mathcal{D}_{ID} \times \mathcal{D}_{PW}$ are exhausted.

It is obvious that the above procedure is with a time complexity of $\{\mathcal{O}(|\mathcal{D}_{ID}| * |\mathcal{D}_{PW}|)\} * (T_D + 4T_H + 7T_X)$, where T_E, T_H and T_X denote the execution time

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

of modular exponentiation, hash and XOR operation, respectively. Based on the results reported in [6, 23], the offline password guessing attack is able to be carried out in seconds on a common computer, for in practice the size of identity space $\mathcal{D}_{ID}$ and the size of dictionary space $\mathcal{D}_{PW}$ are rather limited and $\mathcal{M}$ could try all the possible passwords through an offline method [8, 13]. All in all, the attacker $\mathcal{M}$ can guess (ID_i, PW_i) within polynomial time bound, it follows that our suggested attack is indeed effective.

3.3 Replay attack

Resistance to replay attack is a very basic security goal of any cryptographic protocol [23, 29]. However, Wu-Xu's scheme fails to achieve this goal. More specifically, Wu-Xu's scheme employs random numbers but not timestamps to achieve the freshness of messages. Yet, this scheme has only two protocol flows, making it inherently vulnerable to replay attack. As is well known that, any two-flow random number based scheme is unable to achieve explicit authentication while resisting replay attack, because $\mathcal{M}$ can simply replay the first message of a successful protocol run to impersonate the legitimate user, and the server can never know whether the replayed message is fresh or not unless the server maintains a table of all received messages. However, maintaining a table of all received messages is practically undesirable. In all, replay attack is quite realistic against Wu-Xu's scheme.

4 Review of Roy *et al.*'s scheme

In this section, we first concisely review Roy *et al.*'s scheme [17] proposed in 2017. This scheme improves over Tsai-Lo's scheme [20] and aims to attain forward secrecy that is lacked in Tsai-Lo's scheme. Roy et al.'s protocol involves three participants, i.e., the mobile user (MU_i), the cloud server (CS_j) and registration center (RC). Five phases are involved in their scheme: registration, login, authentication and key establishment, password change, and revocation of mobile device. The notations and initial system parameters employed in Roy *et al.*'s scheme are same as employed in the scheme of Wu-Xu (see Table 1).

4.1 Mobile User Registration

Step 1. MU_i chooses her identity ID_i, password PW_i, biometrics B_i and two 128-bit random numbers b and k.
Step 2. MU_i produces $(\theta_i, \phi_i) = Gen(B_i)$ and computes the masked password $RPWB_i = H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b))$.
Step 3. $MU_i \Rightarrow RC : \{ID_i, (RPWB_i \oplus k)\}$.
Step 4. RC selects an 1024-bit master secret key X_j for server CS_j. RC also selects an 1024-bit random number r_{ij} for each MU_i and CS_j pair.
Step 5. RC computes $A_{ij} = H(H(ID_i \oplus r_{ij}) \parallel X_j)$, $V_{ij} = A_{ij} \oplus RPWB_i$ and $RID_{S_j} = H(ID_{S_j} \parallel X_j)$ as the pseudo-identity of CS_j.

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

Step 6. RC selects a unique and random temporary identity TID_i for MU_i and saves n server key-plus-id combinations $\{TID_i, (ID_{S_j}, V_{ij}, RID_{S_j}) | 1 \leq j \leq n\}$ in mobile device of MU_i.
Step 7. $RC \Rightarrow MU_i$: A mobile device contains $\{TID_i, (IDS_j, V_{ij}, RID_{S_j}) | 1 \leq j \leq n\}$.
Step 8. MU_i computes $D_i^1 = H(PW_i \parallel \theta_i) \oplus b$, $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b)$, $V'_{ij} = V_{ij} \oplus k = A_{ij} \oplus H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b))$, $RID_{ij} = TID_i \oplus H(ID_i \parallel V'_{ij})$ and $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i \parallel b)$ for $1 \leq j \leq n$.

Finally, MU_i stores ϕ_i, D_i^1, D_i^2, V'_{ij}s, RID_{ij}s and RID'_{S_j}s into her own mobile device, and deletes V_{ij}s, TID_i and RID_{S_j}s from the mobile device.

4.2 Cloud server registration

Step 1. CS_j chooses her identity ID_{S_j}.
Step 2. $CS_j \Rightarrow RC : \{ID_{S_j}\}$.
Step 3. RC provides the master secret key X_j to each CS_j.
Step 4. For all MU_is, the RC saves the credentials $\{TID_i, (ID_i, r_{ij})\}$ (for $1 \leq i \leq m$) in database of CS_j, and also stores $\{ID_{S_j}, X_j\}$ in the database of CS_j.

Finally, RC saves pair (ID_i, SN_i) in its own database, where SN_i is the serial number of MU_i's mobile device.

4.3 Login phase

When wanting to login to CS_j, MU_i performs the following operations:

Step L1. MU_i inputs her identity ID_i, password PW_i and biometrics B'_i into her own mobile device. MU_i computes $\theta_i = Rep(B'_i, \phi_i)$ with ϕ_i through the fuzzy extractor reproduction procedure and generates $b' = D_i^1 \oplus H(PW_i \parallel \theta_i)$ with the stored parameter D_i^1, MU_i.
Step L2. MU_i computes $H(ID_i \parallel PW_i \parallel \theta_i \parallel b')$ and checks whether $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b')$. An equality indicates that MU_i is legal.
Step L3. MU_i calculates $RPWB_i = H(ID_i \parallel H(PW_i \parallel \theta_i \parallel b'))$. MU_i also generates $A_{ij} = V'_{ij} \oplus RPBW_i$ using the device parameter V'_{ij}.
Step L4. MU_i selects an 128 bit random number RN_i, generates the current timestamp TS_i, and then computes $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(IDS_j)$, $H_1 = H(ID_i \parallel C_1 \parallel RN_i \parallel TS_i)$, $TID_i = RID_{ij} \oplus H(ID_i \parallel V'_{ij})$, $RID_{S_j} = RID'_{S_j} \oplus H(\theta_i \parallel b')$, $TID_i^* = TID_i \oplus H(RID_{S_j} \parallel TS_i)$.
Step L5. $MU_i \rightarrow CS_j : \{TID_i^*, C_1, H_1, TS_i\}$.

4.4 Authentication phase

In this phase, CS_j and MU_i mutually verify each other and agree on a session key. Since this phase is unrelated to our discussions, we omit it.

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

5 Flaws in Roy *et al.*'s scheme

Recall that the three assumptions listed in Section 3 are also clearly made in Roy *et al.*'s scheme. However, we observe that this scheme still remains feasible for an attacker to offline guess a user's password and break forward secrecy. This means that the primary goal of "truly two-factor security" cannot be satisfied with. In addition, the scheme cannot provide forward secrecy and sound scalability.

5.1 Offline password guessing attack

Note that Roy et al.'s scheme [17] is originally a three-factor one, here we are only interested in its two-factor version by assuming that the third factor (i.e., the biometric) has been known to $\mathcal{A}$. This is realistic as user biometrics are constant during their lives, and how to protect user biometric template is still an open issue [14]. We find that this scheme cannot achieve truly two-factor security: it is subject to two types of offline password guessing attack. This in turn indicates that it cannot achieve truly three-factor security.

Type-I attack. Suppose U_i's biometric B_i and the secret parameters $\{D_i^1, D_i^2, \varphi_i, h(\cdot)\}$ stored in the smart card are somehow obtained by $\mathcal{A}$. At this point, $\mathcal{A}$ can find out U_i's identity and password as follows:

Step 1. Guesses U_i's identity ID_i^* and password PW_i^* from dictionary space $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$.
Step 2. Computes $\theta_i = Rep(B_i, \varphi_i)$, $b^* = H(PW_i \parallel \theta_i) \oplus D_i^1$, where D_i^1 is extracted from the smart card.
Step 3. Computes $D_i^{2*} = H(ID_i^* \parallel PW_i^* \parallel \theta_i \parallel b^*)$, where θ_i is extracted from the smart card.
Step 4. Checks the validity of (ID_i^*, PW_i^*) by comparing the calculated D_i^{2*} with the extracted D_i^2.
Step 5. Repeats Step 1∼4 until find the correct pair of (ID_i^*, PW_i^*).

The time complexity of this attack is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * 2T_H + T_B)$ [23, 24]. Generally, it is only needed to calculate the bio-hashing function once, thus T_B can be ignored in practice. According to the running time in [23], $\mathcal{A}$ may complete the above attacking procedure within 17.6 days on a Laptop. This issue arises due to the inherent "usability-security tension": to achieve local password change (i.e., C-2 in [24]) and timely typo detection (i.e., C-9 in [24]), there is an explicit password verifier $D_i^2 = H(ID_i \parallel PW_i \parallel \theta_i \parallel b)$ stored in U_i's smart card, yet this verifier leads to a Type-I offline password attack.

To eliminate this security issue without loss of usability, a promising countermeasure is to adopt the "fuzzy-verifier" technique [13] and store D_i^2=$h((H(ID_i \parallel PW_i \parallel \theta_i \parallel b))$ `mod` $n)$ in U_i's smart card, where n determines the capacity of (ID, PW) pair, $2^4 \leq n \leq 2^8$. In this way, even if $\mathcal{A}$ obtains D_i^2, she can not determine the correct (ID, PW) from the above attack, because there will be $\frac{|\mathcal{D}_{id} * \mathcal{D}_{pw}|}{n} \approx 2^{32}$ candidate (ID, PW) pairs that make $D_i^{2*} = D_i^2$ in Step 4. To further identify the exactly correct (ID, PW) pair, $\mathcal{A}$ needs to interact with the

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

server, and we can adopt the "honeywords" technique [22, 24] to confine $\mathcal{A}$'s advantage to a very limited value.

Type-II attack. In the above attack, $\mathcal{A}$ does not need the protocol messages. In this attack, we presume that $\mathcal{A}$ can somehow obtain user's smart card and extract its secret parameters $\{D_i^1, D_i^2, V'_{ij}, \varphi_i, h(\cdot)\}$, and also can eavesdrop the login messages $\{TID_i^*, C_1, H_1, TS_i\}$ from the open channel. Now, $\mathcal{A}$ can *offline* guess U_i's password and identity simultaneously as follows:

Step 1. Pick a pair of ID_i^*, PW_i^* from dictionary space $\mathcal{D}_{id}$ and $\mathcal{D}_{pw}$.
Step 2. Computes $\theta_i = Rep(B_i, \varphi_i)$, $b^* = H(PW_i \parallel \theta_i) \oplus D_i^1$, where D_i^1 is extracted from the smart card.
Step 3. Computes $A_{ij}^* = V'_{ij} \oplus H(ID_i^* \parallel H(PW_i^* \parallel \theta_i \parallel b^*))$.
Step 4. Computes $RN_i^* = C_1 \oplus A_{ij}^* \oplus TS_i \oplus H(ID_{S_j})$;
Step 5. Computes $H_1^* = H(ID_i^* \parallel C_1 \parallel RN_i^* \parallel TS_i)$;
Step 6. Checks the correctness of (ID_i^*, PW_i^*) by comparing if the calculated H_1^* equals the intercepted H_1.
Step 7. Repeats Step 1∼6 of the above procedure until find the correct value of (ID_i^*, PW_i^*).

To conduct the above attack, the time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (5T_H + T_B))$, and $\mathcal{A}$ may complete the procedure within 44 days on a Laptop. In comparison, the Type-I attack is more practical.

5.2 No forward secrecy

A scheme that supports forward secrecy ensures that, even after the long-term private key(s) of one or more participants were leaked, previously agreed session keys remain secure [21]. This is important, especially when considering the serious situations of today's clouds like the compromise of cloud servers (e.g., [1]).

If an attacker $\mathcal{M}$ has obtained the server CS_j's long-term key X_j from the breached server S and intercepted the messages $\{TID_i^*, C_1, H_1, TS_i\}$ that are exchanged between U_i and S's authentication process from the public channel. $\mathcal{M}$ is able to figure out the session key using the following method:

Step 1. $\mathcal{M}$ computes $RID_{S_j} = H(ID_{S_j} \parallel X_j)$, extracts $TID_i = TID_i^* \oplus H(RID_{S_j} \parallel TS_i)$
Step 2. $\mathcal{M}$ computes $A_{ji} = H(H(ID_i \oplus r_{ij}) \parallel X_j)$, $RN_j = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus A_{ji}$, $M_1 = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji}$;
Step 3. $\mathcal{M}$ gets the session key $sk = H(ID_i \parallel ID_{S_j} \parallel A_{ji} \parallel M_1 \parallel RN_j \parallel TS_i \parallel TS_j)$.

With the session key sk computed, the entire session will be no secret to $\mathcal{M}$.

5.3 Poor scalability

In Roy *et al.*'s scheme, the user side stores all the cloud servers (i.e., CS_j, $1 \leq j \leq n$) related information: $RID_{ij} = TID_i \oplus H(ID_i \parallel V'_{ij})$ and $RID'_{S_j} =$

Please cite this paper as: Yaosheng Shen, Ding Wang, and Ping Wang. Revisiting Anonymous Two-Factor Authentication Schemes for Cloud Computing. Proceedings of the 4th International Conference on Cloud Computing and Security (ICCCS 2018), Springer. Accepted, to appear, 2018.

$RID_{S_j} \oplus H(\theta_i \parallel b)$ for $1 \leq j \leq n$. This means that, when a new server arrives, all the users have to re-register with the registration center RC. This show poor scalability. Similarly, the server side stores all the users (i.e., U_i, $1 \leq i \leq m$) related information: the credentials $\{TID_i, (ID_i, r_{ij})\}$ (for $1 \leq i \leq m$) in database of CS_j. This means that, when a new user arrives, all the servers have to re-register with the registration center RC.

6 Conclusion

A large mount of efforts have been directed to the design of smart-card-based password authentication scheme, yet it is still an open challenge to devise an efficient, secure and privacy-preserving scheme based on the assumption that smart cards can be tampered. Very recently, Wu-Xu and Roy *et al.* made two another attempts. However, through careful analysis we show that all of them still have several serious drawbacks being overlooked. Taken our attacks in mind, we are considering to design an efficient scheme with provable security.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants No. 61472016, and by the National Key Research and Development Plan under Grants Nos. 2016YFB0800603 and 2017YFB1200700.

References

1. All Data Breach Sources (Feb 2018), `https://breachalarm.com/all-sources`
2. Alsaleh, M., Mannan, M., Van Oorschot, P.: Revisiting defenses against large-scale online password guessing attacks. IEEE Transactions on Dependable and Secure Computing 9(1), 128–141 (2012)
3. Chen, B., Kuo, W.: Robust smart-card-based remote user password authentication scheme. Int. J. Commun. Syst. 27(2), 377–389 (2014)
4. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory 29(2), 198–208 (1983)
5. Hao, Z., Zhong, S., Yu, N.: A time-bound ticket-based mutual authentication scheme for cloud computing. Int. J. Comput. Communi. & Control 6(2), 227–235 (2011)
6. He, D., Chen, J., Hu, J.: An id-based client authentication with key agreement protocol for mobile client-server environment on ecc with provable security. Inform. Fusion 13(3), 223–230 (2012)
7. Huang, X., Chen, X., Li, J., Xiang, Yang ang Xu, L.: Further observations on smart-card-based password-authenticated key agreement in distributed systems. IEEE Trans. Para. Distrib. Syst. 25(7), 1767–1775 (2014)
8. Islam, S.: Design and analysis of an improved smartcard-based remote user password authentication scheme. Int. J. Commun. Syst. 29(11), 1708–1719 (2016)
9. Jiang, Q., Khan, M.K., Lu, X., Ma, J., He, D.: A privacy preserving three-factor authentication protocol for e-health clouds. The J. Supercomput. 72(10), 3826–3849 (2016)

10. Jiang, Q., Ma, J., Li, G., Li, X.: Improvement of robust smart-card-based password authentication scheme. Int. J. Commun. Syst. 28(2), 383–393 (2014)
11. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: CRYPTO 1999, LNCS, vol. 1666, pp. 388–397. Springer (1999)
12. Li, X., Niu, J., Liao, J., Liang, W.: Cryptanalysis of a dynamic identity-based remote user authentication scheme with verifiable password update. Int. J. Communi. Syst. 28(2), 374–382 (2015)
13. Ma, C.G., Wang, D., Zhao, S.D.: Security flaws in two improved remote user authentication schemes using smart cards. Int. J. Commun. Syst. 27(10), 2215–2227 (2014)
14. Memon, N.: How biometric authentication poses new challenges to our security and privacy [in the spotlight]. IEEE Signal Process. Mag. 34(4), 196–194 (2017)
15. 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)
16. Odelu, V., Das, A.K., Kumari, S., Huang, X., Wazid, M.: Provably secure authenticated key agreement scheme for distributed mobile cloud computing services. Future. Gener. Comput. Syst. 68, 74–88 (2017)
17. Roy, S., Chatterjee, S., Das, A.K., Chattopadhyay, S., Kumar, N., Vasilakos, A.V.: On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. IEEE Access 5, 25808–25825 (2017)
18. Song, R.: Advanced smart card based password authentication protocol. Comput. Stand. & Inter. 32(5), 321–325 (2010)
19. Sood, S.K., Sarje, A.K., Singh, K.: An improvement of xu et al.'s authentication scheme using smart cards. In: Proceedings of ACM COMPUTE 2010. pp. 1–5. ACM (2010)
20. Tsai, J.L., Lo, N.W.: A privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE Syst. J. 9(3), 805–815 (2015)
21. Wang, C., Xu, G.: Cryptanalysis of three password-based remote user authentication schemes with non-tamper resistant smart card. Secur. Commun. Netw. (2017), doi: 10.1002/sec.817
22. Wang, D., Cheng, H., Wang, P., Yan, J., Huang, X.: A security analysis of honeywords. In: Proceedings of NDSS 2018. pp. 1–16. ISOC (2018)
23. Wang, D., He, D., Wang, P., Chu, C.H.: Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. IEEE Trans. Depend. Secur. Comput. 12(4), 428–442 (2015)
24. Wang, D., Wang, P.: Two birds with one stone: Two-factor authentication with security beyond conventional bound. IEEE Trans. Depend. Secur. Comput (2016), doi: 10.1109/TDSC.2016.2605087
25. Wei, F., Zhang, R., Ma, C.: A provably secure anonymous two-factor authenticated key exchange protocol for cloud computing. Fund. Inform. 157(1), 201–220 (2018)
26. Wu, F., Xu, L.: A chaotic map-based authentication and key agreement scheme with user anonymity for cloud computing. In: Proceedings of ICCCS 2017. pp. 189–200. Springer (2017)
27. Xie, Q., Wong, D.S., Wang, G., et al.: Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Trans. Inform. Foren. Secur. 12(6), 1382–1392 (2017)
28. 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)
29. Yang, G.M., Wong, D.S., Wang, H.X., Deng, X.T.: Two-factor mutual authentication based on smart cards and passwords. J. Comput. Syst. Sci. 74(7), 1160–1172 (2008)