Computers & Security 88 (2020) 101619

Contents lists available at ScienceDirect

Computers & Security

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

Understanding security failures of multi-factor authentication schemes for multi-server environments☆

Ding Wang[a,b,c], Xizhe Zhang[a], Zijian Zhang[d], Ping Wang[c,d,e,f,*]

[a] School of EECS, Peking University, Beijing 100871, China
[b] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[c] Key Lab of High-Condence Software Technology (PKU), Ministry of Education, China
[d] School of ECE, Peking University Shenzhen Graduate School, Shenzhen 518055, China
[e] School of Software and Microelectronics, Peking University, Beijing 100260, China
[f] National Engineering Research Center for Software Engineering, Beijing 100871, China

ARTICLE INFO

Article history:
Received 12 January 2019
Revised 16 August 2019
Accepted 22 September 2019
Available online 24 September 2019

Keywords:
Multi-factor authentication
Password
User anonymity
Smart card loss attack
Multi-factor security
Forward secrecy

ABSTRACT

Revealing the security flaws of existing cryptographic protocols is the key to understanding how to achieve better security. Dozens of multi-factor authentication schemes for multi-server environments were successively proposed, yet most of them have been shortly found problematic. The research pattern of this area has fallen into the undesirable "break-fix-break-fix" cycle, in which lots of efforts have been devoted but little real progress has been made. In this paper, we revisit five leading two-factor authentication schemes for multi-server environments (i.e., Xu et al. scheme at ICICS'17, Wu et al. scheme at FC'17, Leu-Hsieh's scheme at IET IS'14, Zhou et al. scheme at WINET'18 and Roy et al. scheme at IEEE TII'19), and demonstrate that all of them suffer from critical security defects (e.g., no truly multi-factor security and temporary information leakage attack) or are short of important properties (e.g., no user anonymity). Our results invalidate any use of these five schemes for practical applications without further improvement, and underscore some new challenges (e.g., attacks arising from the leakage of session-specific parameters and from malicious insiders) in designing sound multi-factor schemes for multi-server environments. We also draw some useful lessons from the cryptanalysis results.

1. Introduction

User authentication plays a crucial part in ensuring that resources and services at the remote server can only be accessed by legitimate parties. In 1991, Chang and Wu (1991) suggested the first two-factor authentication scheme based on passwords and smart cards, and this influential work has given rise to a number of enhanced proposals, with each different in terms of security (Wang and Wang, 2018; Xie et al., 2017), anonymity (Karuppiah et al., 2018; Wazid et al., 2017), usability (Gupta et al., 2019; Wang and Chang, 1996) and efficiency (Moon et al., 2017; Yu et al., 2014).

However, most of these schemes are designed for the single-server architecture, which means that each user needs to register at all service servers repetitively, and memorizes n pairs of identity and password to login n different service servers. Recent research has reported that common users generally have 25–67 (identity, password)-pairs (Stobert and Biddle, 2018). It is a great burden for human-beings to maintain (memorize) such an amount of password pairs. As the number of services is constantly increasing, it is imperative to reduce the demand of human memorization. Accordingly, a number of two-factor authentication protocols (see Amin et al., 2018b; He and Wang, 2015; Tsai, 2008; Wang et al., 2018; Wang et al., 2009; Wu et al., 2017; Xu et al., 2017; Xue et al., 2014) for multi-server architecture have been successively developed. In such schemes, the user needs to memorize only one password to access the services/resources at multiple servers.

As shown in Fig. 1, there are three kinds of participants involved in a two-factor scheme[1] for multi-server architecture: a set

☆ This paper was presented in part at the Proceeding of the 20th International Conference on Information and Communications Security (ICICS 2018) Wang et al. (2018c).

* Corresponding author at: Peking University, Room 1800, Science Building 1#, No.5 Yiheyuan Road, Haidian District, Beijing 100871, China.
E-mail addresses: wangdingg@pku.edu.cn, wangdingg@mail.nankai.edu.cn (D. Wang), zxz5168zxz@pku.edu.cn (X. Zhang), zhangzj@pku.edu.cn (Z. Zhang), pwang@pku.edu.cn (P. Wang).

https://doi.org/10.1016/j.cose.2019.101619
0167-4048/

[1] As with Yang et al. (2008) and Wang et al. (2015a), in this work we mainly consider the most typical kind of two-factor schemes that are composed of password and smart card.

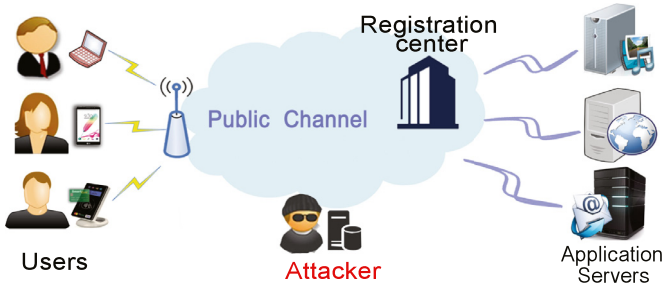

Fig. 1. Multi-factor authentication for the multi-server architecture.

of users, a register center *RC* (also called a control server in some literature (Li et al., 2012; Xue et al., 2014)) and a set of service servers. User *U* can login any service server under the same control server by using the same (identity, password)-pair. User *U* holds a memorable password and a smart card stored with some initial security parameters; The servers (including *RC* and service server *S*) only need to keep some secret key material of the system (but not the user). 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 makes smart-card based password authentication schemes for multi-server architecture rather desirable, especially when considering the incessant leakages of password databases from large websites (see some recent password leakages: 3 billion Yahoo Hackett, 2017, 115 million MyFitnessPal Hackett, 2018, and 68 million Dropbox Heim, 2016, to name just a few).

Besides protocol participants, there is another entity in Fig. 1 (i.e., the attacker) who aims to break the security of the protocol (e.g., impersonating as the user, offline guessing the password, and breaching user privacy). The most important security goal of a two-factor authentication protocol is the so-called "two-factor security" (Wang and Wang, 2018). This security concept essentially means that only the user that has the smart card as well as knowing the correct password can be verified by the server. Nevertheless, past research (Huang et al., 2014; Li et al., 2018; Wang et al., 2016a; 2015a; Xie et al., 2017; Zhang et al., 2017) have, again and again, proved that designing a two-factor authentication scheme with "two-factor security" for single-server architecture is a hard task, and the design of a truly two-factor scheme for multi-server architecture can only be harder. For a concrete grasp of the undesirable situation of this research area, see Fig. 2.

As password guessing attacks (Goodin, 2013; Wang et al., 2016b) and smart-card side-channel attacks (Mangard et al., 2007; Zhou et al., 2013) are advanced rapidly, it is natural to adopt a defense-in-depth approach for security-critical applications (e.g., mobile banking and e-health): introducing another factor (i.e., biometric like fingerprint and gait) into two-factor authentication schemes. This is particularly promising for situations where biometric sensors are already in place in users' mobile devices and two-factor security is deemed inadequate. As with Huang et al. (2011) and He and Wang (2015), in this work we mainly consider the most typical kind of three-factor schemes that are composed of password, smart card and biometric. Generally, three-factor schemes can be built on two-factor ones with biometric, and their key security goal is "three-factor security".

In the vicious "break-fix-break-fix" cycle of past research (see Fig. 2), lots of efforts have been devoted, yet little real progress has been made. More often than not, the protocol designers (see Amin et al., 2018a; Barman et al., 2019; Irshad et al., 2016; Lu et al., 2015a; Lu et al., 2015b; Odelu et al., 2015; Yao et al., 2019 for concrete examples) first revisit one previous scheme and show its security defects and/or usability issues, and then propose an enhanced scheme to remedy the identified flaws. Indeed, many improved schemes are able to overcome the defects in their predecessor (precedent scheme), but the protocol designers often unconsciously overlook many other difficulties and challenges in designing this kind of schemes. Without a holistic view of past research, we can only be stuck in the rut.

To alleviate such an undesirable situation, in this work we investigate five foremost protocols and show that they are all unable to achieve the claimed important design goals, and also draw some lessons herein. We believe that the illustration of repeated pitfalls in previous schemes can help the research community to avoid common mistakes, and it is the key to understanding how to achieve better security. Particularly, four of our five investigated protocols are claimed to be "proved secure" through some formal methods (e.g., Random oracle model Bellare et al., 2000 and Burrows–Abadi–Needham (BAN) logic Burrows et al., 1989), but they turn out to be insecure. This once again underscores the essential role of old-fashioned cryptanalysis and the importance of understanding potential threats when designing a protocol, suggesting the necessity of this work. In a nutshell, our contributions are three-fold:

1. We examine five foremost multi-factor authentication schemes for multi-server architecture, namely Xu et al. (ICICS'17 Xu et al., 2017), Wu et al. (FC'17 Wu et al., 2017), Leu-Hsieh's (IET IS'14 Leu and Hsieh, 2014), Zhou et al. (WINET'18 Zhou et al., 2018) and Roy et al. (IEEE TII'19 Roy et al., 2019)), and reveal that most of them fail to attain truly multi-factor security, forward secrecy and other important properties like user anonymity and repairability.
2. We highlight two frequently repeated downfalls in previous multi-factor schemes for multi-server environments: most schemes providing no truly multi-factor security (see these underlined by a dashed line in Fig. 2) are because of their vulnerability to the smart-card loss attack, and most schemes providing no forward secrecy (see these underlined by a solid line in Fig. 2) are due to the violation of our "forward secrecy principle" first published in Oct. 2012, about seven years ago (Ma et al., 2014).
3. We underline some new challenges (e.g., attacks arising from the leakage of session-specific parameters and from malicious insiders) in designing sound multi-factor schemes for multi-server environments. For instance, an attacker can obtain system-wide parameters by colluding with a malicious service server, and then breach multi-factor security and/or user anonymity, leading to the downfall of schemes in Leu and Hsieh (2014), Fan et al. (2011), Xu et al. (2009), Tsai (2008), Lu et al. (2015a), Mishra et al. (2014) and Lee et al. (2011).

The remainder of the paper is organized as follows: previous literature is briefly reviewed in Section 2; We describe the security loopholes of Xu et al. scheme in Section 3; The pitfalls of Wu et al. scheme are illustrated in Section 4; In Section 5, we point out some weaknesses in Leu–Hsieh's scheme; The flaws in Zhou et al. scheme and Roy et al. scheme are illustrated in Sections 6 and 7, respectively; In Section 8, some lessons learned are drawn. Finally, we conclude in Section 9.

2. Related work

Here we briefly review and summarize the existing literature, providing some necessary backgrounds for our later cryptanalysis.

2.1. Technology roadmaps

In 1991, Chang and Wu (1991) for the first time proposed a "password+smart card" two-factor authentication scheme for

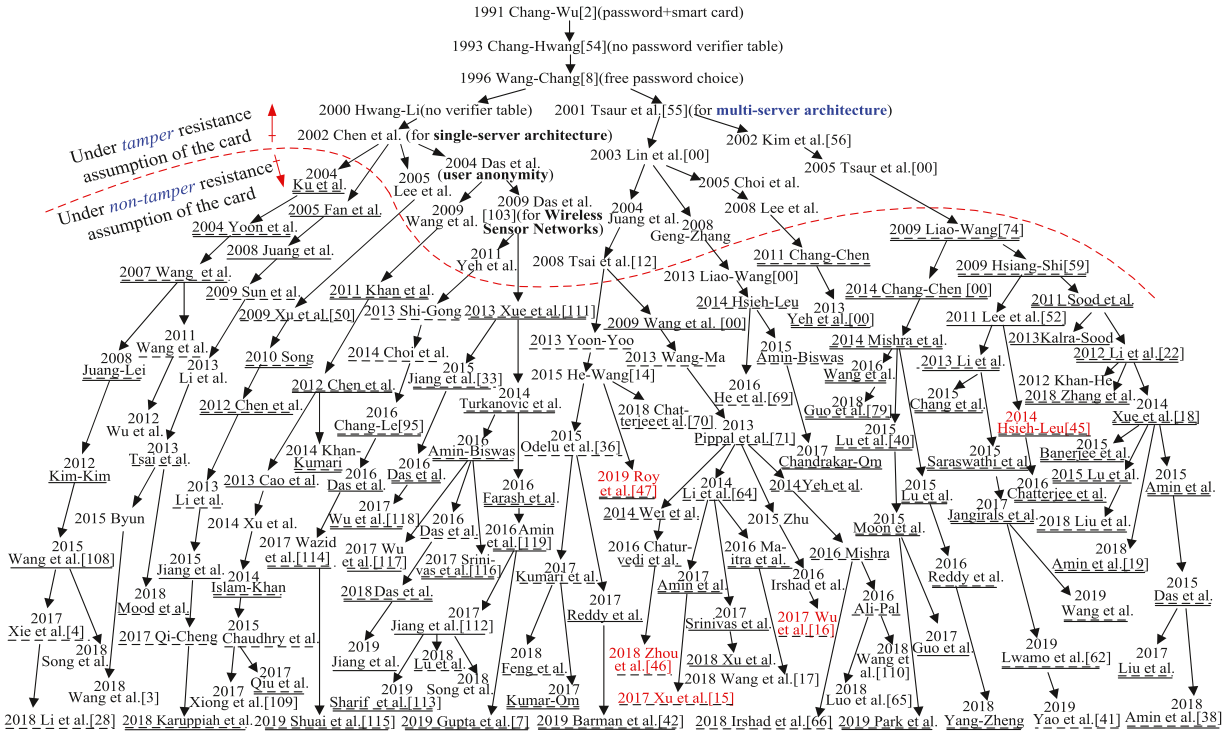

Fig. 2. A brief history of multi-factor authentication for multi-server architecture based on Fig. 1 of Wang et al. (2019a). Schemes underlined by a dashed line cannot achieve *truly multi-factor security*, schemes underlined by a solid line fail to provide *forward secrecy*, while the remaining schemes are prone to other security/usability issues.

the traditional client-server architecture. The security of their scheme is based on the discrete logarithm problem (DLP). One year later, Chang and Laih (1992) showed that Chang–Wu's scheme (Chang and Wu, 1991) suffers from the problem of leaking users' passwords, because a malicious insider user can derive the secret decryption keys of the computer system by using the public information of a smart card. In 1993, Chang and Hwang (1993) proposed an enhanced DLP-based scheme and its server is free from maintaining a verifier table, but it still does not allow a user to choose her own password. In 1996, Wang and Chang (1996) suggested the first two-factor scheme that allows a user to choose her own password. This scheme is based on the intractability of both DLP and the integer factorization problem (IFP).

However, all the aforementioned schemes do not work well in multi-server environments, because each user has to register at various servers repetitively and memorizes n passwords (and hold n smart cards) for n servers. To fill the gap, in 2001 Tsaur (2001) developed the first two-factor authentication scheme for the multi-server architecture. Tsaur's scheme is based on IFP. Later, Kim et al. (2002) pointed out that Tsaur's scheme cannot defend the off-line guessing attack if the attacker intercepts the transmitted authentication message, but Kim et al. did not propose any countermeasure. In 2005, Tsaur et al. (2005) revealed that there is another kind of off-line guessing attack that can be launched by a malicious server, and they further gave a DLP-based improvement over Tsaur's scheme (Tsaur, 2001). At the same time, Lin et al. (2003) proposed another DLP-based improvement that enables a user to be removed/revoked from the system easily.

Following the above seminal works, a large number of two-factor authentication schemes for multi-server architecture have been successively developed (Ali and Pal, 2018; Amin et al., 2018b; He and Wang, 2015; Hsiang and Shih, 2009; Kalra and Sood, 2013; Lwamo et al., 2019; Mishra et al., 2014; Roy et al., 2019; Tsai, 2008; Wang et al., 2018; 2009; Wu et al., 2017; Xu et al., 2017; Xue et al., 2014; Yeh et al., 2013)). These schemes can be classified into five groups in terms of the underlying intractability problem: DLP-based ones (Li et al., 2015; Lin et al., 2003; Lwamo et al., 2019) (and their analogues, ECDLP-based ones (Irshad et al., 2018; Luo et al., 2018; Mishra, 2016; Odelu et al., 2015)), Pairing-based ones (He et al., 2016; Liao and Hsiao, 2013; Roy et al., 2019), Chaotic-map based ones (Chatterjee et al., 2018; Wu et al., 2017), IFP-based ones (Pippal et al., 2013; Tsaur, 2001; Xu et al., 2017), and Hash-based ones (Barman et al., 2019; Hsiang and Shih, 2009; Li et al., 2012; Liao and Wang, 2009; Lu et al., 2015a; Yeh et al., 2013). Each group is subject to its own design difficulties, and we will show this through examining five representative schemes, with each scheme based on a different intractability problem (see a summarization in Table 2).

In 2017, Amin et al. (2017) developed an anonymous two-factor authentication scheme for multi-server architecture based on the intractability of IFP, and stated that their scheme is able to support "two-factor security" under the hypothesis that smart cards can be tampered. Later on, Xu et al. (2017) found that Amin et al. scheme cannot resist against user impersonation attack if the parameters kept in the smart cards can be extracted. This invalidates Amin et al.'s claim of ensuring "two-factor security". Accordingly, Xu et al. (2017) further proposed a new two-factor scheme based on the same cryptographic primitive (i.e., IFP) at ICICS'17. In addition, their scheme was "proved secure" in the random oracle model. Surprisingly, we find that Xu et al. scheme Xu et al. (2017) is subject to a new damaging security flaw: anyone can impersonate any legitimate user.

At FC'17, Wu et al. (2017) demonstrated that various security drawbacks exist in both (Irshad et al., 2016; Zhu, 2015) schemes. More specifically, Irshad et al. scheme is vulnerable to stolen-verifier attack and insider attack, and provides no user anonymity; Zhu's scheme suffers from insider attack, provides no user anonymity, and has the de-synchronization problem in case the malicious attacker $\mathcal{M}$ simply modifies the third message flow. Wu et al. (2017) also put forward a Chaotic-map based scheme

and argued that their new scheme is robust under the condition that the sensitive data in smart card has been extracted by $\mathcal{M}$. It should be noted that, recent rapid developments in side-channel attacks have proved that the sensitive information stored in general commercial smart cards could be extracted by power analysis (Messerges et al., 2002) or reverse engineering (Amiel et al., 2007). Based on a weak yet realistic assumption, Wu et al. scheme (Wu et al., 2017) appears very practical.

However, as we will show, this scheme is prone to a much more serious problem than the original schemes (i.e., Irshad et al., 2016; Zhu, 2015 schemes): it fails to achieve truly two-factor security, which is the most important goal of a two-factor scheme. Besides, Wu et al. scheme will leak the user's long-term secret key once a session-specific parameter is leaked. This is rather undesirable, because session-specific data is often less well protected than long-term keys, and the leakage of the former should not affect the latter. This attack highlights the challenges arising from the leakage of session-specific data.

Besides robust security guarantees, protocol efficiency is also an important concern due to the resource-constrained nature of user devices. Leu and Hsieh (2014) presented an anonymous two-factor scheme, which is claimed to ensure user privacy and robust security, while only requiring a few lightweight hash operations. Unlike their claims, we show that their scheme still cannot provide truly two-factor security and user anonymity. In addition, forward secrecy cannot be attained. We note that Maitra et al. (2016) have also analyzed Leu–Hsieh's scheme and presented some attacks, but their attacks are different from ours. Besides, Maitra et al. (2016) further gave an improved scheme, which suffers some critical issues as pointed out in Wang et al. (2018).

As password guessing attacks (Goodin, 2013; Wang et al., 2016b) are advanced rapidly, smart-card side-channel attacks are becoming more practical (Mangard et al., 2007; Zhou et al., 2013), and biometric sensors are already deployed in various devices, it is natural to adopt a defense-in-depth approach for security-critical applications (e.g., mobile banking and e-health): introducing the biometric authentication factor (e.g., fingerprint, iris and gait) into two-factor authentication schemes to construct three-factor schemes. Accordingly, there has been a recent upsurge to develop three-factor schemes (see Chang and Nguyen, 2016; Guo et al., 2018; He and Wang, 2015; Odelu et al., 2015; Roy et al., 2017; Song et al., 2018).

While the design of a two-factor authentication scheme with "two-factor security" is hard as shown in Wang and Wang (2018), the question of how to construct a truly three-factor scheme for multi-server architecture can only be harder. In 2018, Zhou et al. (2018) pointed out that most existing three-factor schemes (e.g., Chang and Nguyen, 2016; Guo et al., 2018; He and Wang, 2015; Odelu et al., 2015; Roy et al., 2017; Song et al., 2018) for multi-server architecture are subject to a common flaw: they either need to interact with the registration center during the login phase (e.g., Guo et al., 2018; He and Wang, 2015; Odelu et al., 2015; Song et al., 2018), or need to store parameters of each service server (e.g., Chang and Nguyen, 2016; Roy et al., 2017). Accordingly, they attempted to suggest a new DLP (and bilinear-map) based scheme free from this flaw. However, we show that Zhou et al. scheme Zhou et al. (2018) is far from a practical one and prone to serious security flaws (i.e., no truly two-factor security and temporal information leakage) and usability issues.

In 2019, Roy et al. (2019) also proposed a pairing-based scheme which needs to store parameters of each service server. Roy et al. scheme is proved secure by using two kinds of formal methods (i.e., Random oracle model and ProVerif) to enhance the confidence of its security. As "formal proofs" are no panacea for assuring actual security (Wang et al., 2015a), their scheme is unsurprisingly vulnerable to two kinds of smart card loss attack that lead to the breach of truly three-factor security. In addition, it cannot provide the feature of forward secrecy, while this security feature is becoming more and more desirable as zero day attacks are prevailing (e.g., Cerrudo, 2014; Heartbleed – openssl zero-day bug leaves millions of websites vulnerable, 2014).

2.2. Adversary models

Since a series of influential work (Huang et al., 2011; Wang et al., 2015a; Yang et al., 2008), generally four assumptions are made about $\mathcal{M}$'s capabilities against multi-factor authentication. We summarize them as follows:

Assumption 1. $\mathcal{M}$ completely manipulates the public channel (e.g., eavesdrop, delete, insert, modify or block any transcripts). This well complies with the standard adversary model that is widely accepted for distributed computing (Dolev and Yao, 1983);

Assumption 2. $\mathcal{M}$ can somehow obtain the victim's smart card and exploit side-channel attacks (Amiel et al., 2007; Balasch et al., 2012; Mangard et al., 2007; Messerges et al., 2002) to extract sensitive data from the card memory.

Assumption 3. Users' passwords are selected from a very constrained space and $\mathcal{M}$ can brute force it. To increase usability, most schemes (e.g., the ones in He et al., 2018; He and Wang, 2015; Wang and Wang, 2018; Xie et al., 2017; Yang et al., 2008) allow the users to choose passwords at their discretion during registration phase or password change phase. Generally, human beings are only capable of memorizing 5–7 different passwords, and tend to select popular passwords, use personal information to build passwords and reuse passwords. Recent research has shown that user-chosen passwords follow the Zipf's law (Wang et al., 2017; Wang and Wang, 2016) and come from a small space.

Assumption 4. User's biometric information can be obtained by the malicious attacker $\mathcal{M}$. On the one hand, user's biometric factor is digitally stored in computer and can be leaked like that of passwords (Goodin, Dan, 2018), and it can also be captured, stolen and spoofed (Kokalitcheva, Kia, 2016; Memon, 2017); On the other hand, user's biometric factor is generally static over the user's lifetime, and cannot be revoked/updated like passwords.

Note that, for a two-factor scheme, if both Assumptions 2 and 3 hold at the same time, then the attacker (with no need of other abilities) is able to impersonate any victim user and can trivially breach any scheme. This is the trivial attack. Thus, it is common practice to *not* assume that the attacker acquires a victim user's both (all) authentication factors when analyzing security (Chang and Le, 2016; Huang et al., 2014; Islam, 2016; Jiang et al., 2014; Wang and Wang, 2018; Wang et al., 2009; Wang and Peng, 2015; Xie et al., 2017). This treatment complies with "the extreme-adversary principle" proposed by Hao (2010): "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 of this type of schemes". Similarly, for a three-factor scheme, not all Assumptions 2–4 are allowed to hold at the same time to avoid trivial attacks.

Also note that, an attacker might be an insider of the system, and it is practical for her to obtain both her own card and password. As shown in Section 5.2.3 of Wang et al. (2016a), such an attacker is really powerful and poses great challenges to the security of the system. Overlooking the threats from this kind of attacker is likely to open large security loopholes. Many previous password authentication schemes employing smart cards (e.g., Kumari et al., 2014; Wang, 2012; Xu et al., 2009) fail to achieve "two-factor security" or user un-traceability, when confronted with such a mali-

Table 1
Notations and abbreviations.

Symbol	Description	Symbol	Description
U_i	ith user	S_j	jth server
ID_i	identity of U_i	PW_i	password of U_i
Bio_i	Biometric of U_i	SK_{ij}	secret key established by U_i and S_j
RC	the register center	S_x	the foreign server
S_y	the home server	k_{xy}	secret key shared by S_x and S_y
SID_j	the identity of S_j	$Skey_j$	secret key of S_j
$\mathcal{M}$	the malicious attacker	k_y	the secret key of S_y
$\mathcal{D}_{id}$	the distribution of identity space	$\mathcal{D}_{pw}$	the distribution of password space
x/d	the secret key of RC	e/PK	the public key of RC
$\Rightarrow$	a secure channel	$\oplus$	bitwise XOR operation
$\rightarrow$	a common channel	$h(\cdot)$	one-way hash function

cious insider. In this work, special attention is devoted to this kind of attacker and we show its perniciousness.

According to the abilities that are exploited by an attacker to launch a attack, nine types of attackers can be further classified as follows:

(I) **Basic attacker**. This attacker is only based on Assumption 1.
(II) **Attacker with the target user's smart card**. This attacker rests on the Assumption 1 and 2.
(III) **Attacker with the target user's password**. This attacker rests on the Assumption 1 and 3.
(IV) **Attacker with the target user's biometric**. This attacker rests on the Assumption 1 and 4.
(V) **Attacker with her own smart card and password**. This attacker rests on the Assumption 1–3, being a malicious insider.
(VI) **Attacker with the target user's smart card and password**. This attacker rests on the Assumption 1–3.
(VII) **Attacker with the target user's smart card and biometric**. This attacker rests on the Assumption 1, 2 and 4.
(VIII) **Attacker with the target user's password and biometric**. This attacker rests on the Assumption 1, 3 and 4.
(IX) **Attacker with her own smart card, password and biometric**. This attacker rests on the Assumption 1–4, being a malicious insider.

It is evident that the *basic attacker* is with the fewest capabilities, while the eight remaining attackers are all realistic according to the aforementioned discussions. More specifically, any two-factor scheme aiming for practical use shall be able to withstand the first four attackers, while any three-factor scheme aiming for practical use shall be able to withstand all the nine attackers. All the five multi-factor schemes examined in this work are claimed to be secure under the above assumptions, but as we will show, this is not the case. For example, Wu et al. (2017) stated that their scheme is secure "even if the attacker steals the smart card of the user U_i and extracts the parameters $\{Z_i, V_i, B_i, H_i, b, h(\cdot), h(y)\}$ stored in the smart card by some methods". However, contrary to their claims, we will show that this scheme still provides no truly two-factor security or user anonymity.

3. Cryptanalysis of Xu et al. scheme

We first review Xu et al. scheme Xu et al. (2017) proposed at ICICS'17, and then show that it is subject to a damaging security flaw: anyone can impersonate any legitimate user without guessing the victim's password or obtaining the victim's device.

3.1. A brief review of Xu et al. scheme

Xu et al. scheme Xu et al. (2017) is composed of four phases. For simplicity, the notations employed throughout this paper are listed in Table 1; We will comply with the abbreviations in Xu et al. scheme closely.

Server Registration Phase. This phase proceeds as follows:

Step 1. $S_j \Rightarrow RC$: $\{e_j, n_j, SID_j\}$. S_j computes $n_j = p_j \times q_j$, $\phi(n_j) = (p_j - 1)(q_j - 1)$ where both p_j and q_j are large prime numbers, then chooses a public key $e_j (1 < e_j < \phi(n_j))$ where $\gcd(\phi(n_j), e_j) = 1$, and computes $d_j \equiv e_j^{-1} \bmod \phi(n_j)$ as its private key.
Step 2. $RC \Rightarrow S_j$: $\{Cer_j\}$. RC computes $Cer_j = h(e_j \parallel SID_j \parallel n_j)^d$.

User Registration Phase. This phase proceeds as follows:

Step 1. $U_i \Rightarrow RC$: $\{ID_i\}$.
Step 2. $RC \Rightarrow U_i$: $\{d_i\}$. RC computes $d_i = h(ID_i)^d \bmod n_j$.

Login and authentication phase. This phase proceeds as follows:

Step 1. $U_i \rightarrow S_j$: a random number T_i.
Step 2. $S_j \rightarrow U_i$: $\{e_j, n_j, Cer_j, A_j\}$ where $A_j = h(T_i)^{d_j}$.
Step 3. $U_i \rightarrow S_j$: $\{PID_i, R_i, S_i, x\}$. If $Cer_j^e \bmod n_j$ equals $h(SID_j \| e_j \| n_j)$ and $A_j^{e_j}$ equals $h(T_i)$, U_i computes $PID_i = (ID_i \oplus a_i \parallel a_i)^{e_j} \bmod n_j$, $R_i = h(PID_i)^r \bmod n_j$, $x = h(m, R_i)$ and $Si = d_i^{r-x}$ where a_i, r, m are three random numbers.
Step 4. S_j computes $S_i^e = h(ID_i)^{r-x}$, $PID_i^{d_j} \bmod n_j = ID_i \oplus a_i \| a_i$, $ID_i' = ID_i \oplus a_i \| a_i$. If $S_i^e h(ID_i')^x$ equals Ri, S_j authenticates U_i.

3.2. Flaws in Xu et al. scheme

3.2.1. User impersonation attack.

Xu et al. claimed that their "proposed scheme can provide proper mutual authentication", but we show that this is not the case: Anyone can impersonate any legitimate user without guessing password or accessing the victim's device:

Step 1. $\mathcal{M}$ chooses a random number $T_i \in_R (1, n_j]$;
Step 2. $\mathcal{M}$ receives $\{e_j, n_j, Cer_j, A_j\}$ that comes from S_j;
Step 3. $\mathcal{M}$ chooses a random number $X \in_R (1, n_j]$;
Step 4. $\mathcal{M}$ sets $S_i = X$, $R_i = X^e h(ID_i)^X$;
Step 5. $\mathcal{M} \rightarrow S_j$: $\{PID_i, R_i, S_i, x = X\}$, where PID_i is intercepted.

Note that the above attack will succeed, because $\{PID_i, R_i, S_i, x = X\}$ will be accepted by the service server S_j. More specifically, according to attack Step 4, we have $S_i^e h(ID_i)^x = X^e h(ID_i)^X$, which equals R_i and passes Step 4 of login phase. This demonstrates that even a Type-I attacker (see Section 2) can completely break the scheme, suggesting that this scheme is fundamentally flawed.

3.2.2. Poor repairability.

In Xu et al. scheme, there should be times that a user suspects (or realizes) that her smart card might be power analysed

Table 2
Summary of our cryptanalysis results and the underlying vulnerabilities (info=information).

Scheme	Weaknesses	Vulnerabilities exploited
Xu et al. (2017) (IFP-based)	User impersonation	Logic flaw in the design.
	Poor repairability	Overlooking the property.
Wu *et al.* (Chaotic-based)	Smart card loss attack II	When using Elgamal-like encryption (including chaotic-based and ECDLP-based), at least two exponentiations are needed at the user side to construct the login message Wang et al. (2015b).
	Temporary info leakage attack	Temporary info itself is explicitly used to conceal/encrypt long term secret(s).
Leu-Hsieh Leu and Hsieh (2014) (Hash-based)	Smart card loss attack I	There is an explicit password verification verifier stored in the smart card, violating the "security-usability trade-off principle" in Ma et al. (2014); Wang et al. (2015a).
	Smart card loss attack II	No public-key technique is used, violating the "public-key technique principle" in Ma et al. (2014)
	No user anonymity	No public-key technique is used, violating the "public-key principle for user anonymity" in Wang and Wang (2014).
Zhou *et al.* Zhou et al. (2018) (DLP-based)	Smart card loss attack I	There is an explicit password verification verifier stored in the smart card, violating the "security-usability trade-off principle" in Ma et al. (2014); Wang et al. (2015a).
	Temporary info leakage attack	Temporary info itself is explicitly used to conceal/encrypt long term secret(s).
	Poor repairability	Overlooking the property.
Roy *et al.* Roy et al. (2019) (Pairing-based)	Smart card loss attack I	There is an explicit password verification verifier stored in the smart card, violating the "security-usability trade-off principle" in Ma et al. (2014); Wang et al. (2015a).
	Smart card loss attack II	No public-key technique used at the user side, violating the "public-key technique principle" in Ma et al. (2014).
	No forward secrecy	No public-key technique used at the user side, violating the "public-key technique principle" in Ma et al. (2014).

and the secret $d_i = h(ID_i)^d \bmod n$ has been leaked. However, even if U_i has detected this abnormality and changes her password to a new one, no means can be employed to deter $\mathcal{M}$ from using the master secret d_i to login the server S_j. In other words, U_i cannot be easily repaired (Wang et al., 2015a). More Specifically, since $d_i = h(ID_i)^d \bmod n$ is uniquely defined by U_i's identity ID_i and RC's long-term private key d, RC is unable to update d_i for U_i unless either ID_i or d is updated. Nevertheless, because d is usually utilized for all legitimate users of the entire system rather than only one user U_i, it would be irrational and inefficient to change d to restore the security of a single user, i.e. U_i. Furthermore, since ID_i is typically bound with U_i in many application systems, it is also unreasonable to change ID_i to address the problem. In summary, the repairability of Xu et al. scheme constitutes a realistic issue.

4. Cryptanalysis of Wu et al. scheme

Here we first review Wu et al. scheme (Wu et al., 2017). This scheme is an improvement over existing schemes that aims to attain user anonymity lacked in Irshad et al. (2016) and Zhu (2015). Wu et al. scheme does preserve user anonymity, yet we observe that it still remains feasible for an attacker to break the "truly two-factor security". In addition, the scheme cannot provide sound repairability.

4.1. A brief review of Wu et al. scheme

Wu et al. scheme Wu et al. (2017) is composed of four phases: initialization, registration, login and authentication, and one activity: password change. The notations and initial system parameters employed in Wu et al. scheme are the same as those employed in the scheme of Xu et al. (see Table 1).

Initialization phase. Let $k_{xy}(1 \le x, y \le n, x \ne y)$ be the common secret key of each pair of servers $(S_x, S_y)(x \ne y)$, s be their common parameter, and k_y be the secret key of S_y.

User Registration. This phase proceeds as follows:

Step 1. $U_i \Rightarrow S_y$: $\{ID_i, HPW_i\}$, where $HPW_i = h(PW_i \| b_i)$.
Step 2. $S_y \Rightarrow U_i$: $\{PID_i, B_1, B_2, s, h(\cdot)\}$. S_y selects PID_i, then computes: $B_{01} = h(PID_i \| k_y \| ID_{S_y})$, $B_1 = B_{01} \oplus HPW_i$, $B_{02} = h(ID_i \| k_y \| ID_{S_y})$ and $B_2 = B_{02} \oplus h(ID_i \| HPW_i)$, and stores ID_i.
Step 3. U_i inputs $(PID_i, B_1, B_2, B_3, s, h(\cdot))$ into mobile device, where $B_3 = b_i \oplus h(ID_i \| PW_i)$.

Login and authentication phase. This phase proceeds as follows:

Step 1. $U_i \rightarrow S_x$: $M_1 = \{PID_i, C_1, C_2, C_3, C_5, SID_j, ID_{S_y}\}$. U_i inputs ID_i and PW_i, then the device calculates $b_i = B_3 \oplus h(ID_i \| PW_i)$, $HPW_i = h(PW_i \| b_i)$, $C_1 = T_{r_U}(s)$, $C_2 = B_1 \oplus HPW_i \oplus N_U$, $C_3 = h(N_U) \oplus ID_i$, $C_4 = B_2 \oplus h(ID_i \| HPW_i)$ and $C_5 = h(C_1 \| N_U \| C_4)$, where r_U and N_U are two randomly chosen nones.
Step 2. $S_x \rightarrow S_y$: $M_2 = \{PID_i, C_1, C_2, C_3, C_5, C_6, C_7, SID_j\}$. S_x computes $C_6 = T_{r_{S_x}}(s)$ and $C_7 = h(C_6 \| k_{xy} \| SID_j)$, where r_{S_x} is a nonce.
Step 3. There are further messages flows $S_y \rightarrow S_x$: $M_3 = \{C_8, C_9, C_{10}, C_{11}\}$ and $S_x \rightarrow U_i$: $M_4 = \{C_6, C_9 \sim C_{12}\}$, but they have little relevance to our discussions and are omitted.

4.2. Flaws in Wu et al. scheme

We now show the flaws of Wu et al. scheme Wu et al. (2017). Recall that the first three assumptions listed in Section 2 are also explicitly made when Wu et al. analyzing (Irshad et al., 2016; Zhu, 2015) schemes.

4.2.1. Smart card loss attack

Based on Wu et al. own security assumptions (i.e., the first three assumptions in Section 2), we now cryptanalyze the security provisions of their scheme. More specifically, in what follows we assume that $\mathcal{M}$ can extract the private data $\{B_1, B_2, B_3, h(\cdot)\}$ kept in U_i's smart card, and can also eavesdrop the messages $\{PID_i, C_1, C_2, C_3, C_5, SID_j, ID_{S_y}\}$ exchanged between the parties. $\mathcal{M}$ obtains U_i's password PW_i as follows:

Step 1. Guesses the value of ID_i to be ID_i^* from dictionary space $\mathcal{D}_{id}$ and the value of PW_i to be PW_i^* from dictionary space $\mathcal{D}_{pw}$;
Step 2. Computes $b_i^* = B_3 \oplus h(ID_i^* \| PW_i^*)$, where B_3 is revealed from U_i's card;
Step 3. Computes $N_u^* = C_2^* \oplus B_1 \oplus h(PW_i \| b_i^*)$, where C_2 is intercepted from the channel and B_1 is revealed from U_i's card;
Step 4. Computes $C_3^* = h(N_u^*) \oplus ID_i^*$;
Step 5. Verifies the correctness of (ID_i^*, PW_i^*) by checking if C_3^* equals the intercepted C_3;

Step 6. Repeats Step 1 ~ 5 until the right (ID_i^*, PW_i^*) is found.

The time complexity of the attack is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * 3T_H)$, where T_H is the running time for Hash operation. Recently, it has been found that user-chosen password follow the Zipf's law and the dictionary size is very restricted, e.g., $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 10^6$ (Wang et al., 2017). Further, regarding the timings in Table 5 of Wang et al. (2015a), $\mathcal{A}$ may figure out the password within 24.6 days on a common PC; Or, $\mathcal{A}$ may cost $30.36 and spend 16.37 h by using the Amazon EC2 `C4.4X-large` cloud computing service (Amazon elastic compute cloud Amazon EC2, 2018). The above attack means that, once the smart card factor is breached, then the password factor will also be compromised. *This indicates that truly two-factor security cannot be achieved in Wu* et al.*scheme*.

4.2.2. Temporary information leakage attack

As session-specific information is generally of large volume and deemed less sensitive than long-term secret keys, the former will be much less well protected than the latter. Thus, session-specific information can be more easily leaked (e.g., through improper erasing Burmester, 1994, memory leakage Heartbleed – openssl zero-day bug leaves millions of websites vulnerable, 2014 or even poor implementations Caudill, 2016; Volgers, 2016). Therefore, it is desirable that the security impact of the leakage of such session-specific information can be limited to just session-specific secret keys, but not the long-term secret keys.

Yet, in Wu et al. scheme Wu et al. (2017), the leakage of session-specific information will put the long-term secret key at risk:

Step 1. $\mathcal{M}$ somehow obtains the session-specific N_u during one session;
Step 2. Computes $B_{01} = C_2 \oplus N_u = h(PID_i \parallel k_y \parallel ID_{S_y})$.

Note that, $B_{01} = h(PID_i \parallel k_y \parallel ID_{S_y})$ is just U_i's long-term authenticator. After obtaining B_{01}, $\mathcal{M}$ can guess U_i's passwords:

Step 1. Computes $ID_i = C_3 \oplus h(N_u)$, where C_3 is from open channel;
Step 2. Guesses the value of PW_i to be PW_i^* from space $\mathcal{D}_{pw}$;
Step 3. Computes $C_2^* = B_{01} \oplus h(PW_i^* \| b_i^*) \oplus N_u$, where B_{01} is obtained as shown above;
Step 4. Verifies the correctness of PW_i^* by comparing if C_2^* equals C_2;
Step 5. Repeats Step 1 ~ 4 until the right value of PW_i^* is found.

The time complexity is $\mathcal{O}(|\mathcal{D}_{pw}| * 2T_H)$, which can be completed in 1.39s on a common PC according to the timings that $T_H \approx 0.693\mu$ s (see Table 5 of Wang et al. (2015a)). That is, the leakage of session-specific information will lead to the leakage of user identity and passwords. This is rather dangerous.

5. Cryptanalysis of leu-Hsieh's scheme

In 2014, Leu and Hsieh (2014) pointed out that Lee et al. scheme Lee et al. (2011) actually fails to attain truly two-factor security. In order to overcome the revealed pitfalls, Leu and Hsieh (2014) suggested an improved scheme. However, contrary to their claims, Leu–Hsieh's scheme is still subject to serval serious security flaws.

5.1. Review of Leu–Hsieh's scheme

To be self-contained, in this section we concisely describe Leu–Hsieh's scheme Leu and Hsieh (2014) that is suggested in 2014. As with many other schemes, Leu–Hsieh's scheme also includes four phases.

Initialization. RC compute $h(x\|y)$ and $h(y)$ where x is the master secret key and y is the secret number, and then shares $h(x\|y)$ and $h(y)$ with S_j.

Registration phase. This phase proceeds as follows:

Step 1. $U_i \Rightarrow RC$: $\{ID_i, h(b \oplus PW_i)\}$.
Step 2. $RC \Rightarrow U_i$: a smart card with $\{Z_i, V_i, B_i, H_i, h(\cdot), h(y)\}$. RC computes $T_i = h(R_i\|x)$, $Z_i = R_i \oplus ID_i \oplus h(b \oplus PW_i)$, $Vi = T_i \oplus h(ID_i \| h(b \oplus PW_i))$, $Bi = h(b \oplus PW_i) \oplus ID_i \oplus h(h(b \oplus PW_i \oplus R_i) \| h(x\|y))$, $H_i = h(T_i)$ where R_i is a random number for U_i.
Step 3. U_i enters b into the card.

Login and authentication phase. This phase proceeds as follows:

Step 1. $U_i \rightarrow S_j$: $\{CID_i, P_{ij}, Q_i, N_i\}$. U_i inputs ID_i and PW_i, then the smart card computes $R_i = Z_i \oplus ID_i \oplus h(b \oplus PW_i)$, $T_i = V_i \oplus h(ID_i \| h(b \oplus PW_i))$, $H_i' = h(T_i)$. If H_i' equals H_i, the card computes $O_i = h(b \oplus PW_i) \oplus D_i \oplus B_i = h(h(b \oplus PW_i \oplus R_i) \| h(x\|y))$, $A_i = h(T_i \| h(y) \| N_i))$, $CID_i = h(b \oplus PW_i \oplus R_i) \oplus h(T_i \| A_i \| N_i)$, $P_{ij} = T_i \oplus h(h(y) \| N_i \| \mathrm{SID}_j)$, $Q_i = h(O_i \| A_i \| N_i)$. Otherwise, end the session.
Step 2. $S_j \rightarrow U_i$: $\{M_{ij}, N_j\}$. S_j computes $T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, $A_i = h(T_i \| h(y) \| N_i))$. Next, S_j computes $h(b \oplus PW_i \oplus R_i) = CID_i \oplus h(T_i \parallel A_i \| N_i)$ and $O_i = h(h(b \oplus PW_i \oplus R_i) \| h(x\|y))$. If Q_i equals $h(O_i \| A_i \| N_i)$, S_j computes $M_{ij} = h(O_i \| N_i \| A_i \| SID_j)$ where N_j is a nonce. Otherwise, end the session.
Step 3. $U_i \rightarrow S_j$: $\{M_{ij}''\}$. U_i computes: $M_{ij}' = h(O_i \| N_i \| A_i \| SID_j)$. If M_{ij}' equals M_{ij}, U_i computes $M_{ij}'' = h(O_i \| N_j \| A_i \| SID_j)$. Otherwise, end the session.
Step 4. If $h(O_i \| N_j \| A_i \| SID_j)$ equals M_{ij}'', S_j authenticates U_i successfully, and they share a session key $SK = h(O_i \| N_i \| N_j \| A_i \| SID_j)$.

5.2. Flaws in Leu–Hsieh's scheme

We now point out the flaws of Leu–Hsieh's dynamic-ID based scheme. Note that the three assumptions about an adversary's capabilities presented in Section 2 are also clearly stated in Leu–Hsieh's work Leu and Hsieh (2014) as they analyze Lee et al. scheme Lee et al. (2011) and their owns. However, we find Leu–Hsieh's scheme fails to achieve three important goals: (1) User anonymity; (2) Resistance against smart card loss attack; and (3) Forward secrecy.

5.2.1. No user anonymity

With the concern of user privacy rising rapidly nowadays, user anonymity is becoming a primary feature to be considered in the design of authentication protocols, especially in wireless environments (Das, 2009; Wang and Wang, 2014; Wang et al., 2013). In Leu–Hsieh's scheme, the exchanged messages are different in every session due to the use of fresh random nonces, and user identity is dynamic in every session by hiding the true identity ID_i into shadow identities CID_i. In this way, anonymity service is claimed to be provided in Leu and Hsieh (2014) by arguing that an attacker $\mathcal{M}$ "cannot distinguish between different sessions corresponding to a certain user and cannot obtain any clue to the real identity." However, Leu–Hsieh's scheme fails to consider that the attacker $\mathcal{M}$ may be a malicious insider (i.e., a legitimate but malicious user – a type IV attacker, see Section 2). In the following we show that such a type IV attacker $\mathcal{M}$ is able to breach U_i's untraceability:

Step 1. $\mathcal{M}$ eavesdrops a login request $\{P_{ij}, N_i\}$ sent by U_i;
Step 2. $\mathcal{M}$ calculates $T_i = P_{ij} \oplus h(h(y) \| N_i \| SID_j)$, where $h(y)$ is shared among all users and service servers.

Note that, $T_i = h(R_i\|x)$ is specific to U_i and static in all of user U_i's login sessions, and thus it can be used to link the different sessions participated by U_i, breaching the user untraceability.

The above procedure shows that, a type IV attacker (see Section 2) is capable of disclosing the activity of any legitimate user in the system without the sensitive information from user's smart card. Instead, $\mathcal{M}$ only needs the sensitive information from her own knowledge. This is a much weaker condition as compared with the condition that $\mathcal{M}$ needs the sensitive information from U_i's smart card. This is contrary to Leu–Hsieh's claim that $\mathcal{M}$ "cannot obtain any clue to the real identity." Thus, their scheme cannot preserve user anonymity and is not a true dynamic-ID based scheme. Our attack highlights the seriousness of threat arising from malicious insiders.

5.2.2. Smart card loss attack I

We now show that a Type-I attacker $\mathcal{M}$ can breach the two-factor security of Leu–Hsieh's scheme. Assume $\mathcal{M}$ somehow obtains U_i's smart card and extracts $\{V_i, H_i, h(\cdot)\}$ from U_i's smart card, $\mathcal{M}$ can obtain U_i's password PW_i as follows:

Step 1. $\mathcal{M}$ picks a candidate PW_i^* from the password dictionary $\mathcal{D}_{pw}$, and a candidate ID_i^* from the identity dictionary $\mathcal{D}_{id}$.
Step 2. $\mathcal{M}$ computes $T_i^* = V_i \oplus h(ID_i^*\|h(b\|PW_i^*))$;
Step 3. $\mathcal{M}$ computes $H_i^* = h(T_i^*)$;
Step 4. $\mathcal{M}$ examines the validity of PW_i^* by comparing if the computed H_i^* is equal to H_i which is extracted from the card memory.
Step 5. $\mathcal{M}$ goes to Step 2 until the right PW_i is obtained.

The time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (3T_H + T_X))$, where T_X denotes bit-wise XOR operation. Based on the results in Wang et al. (2015a), this attack is able to be carried out in a few days on a common computer, for in practice the size of identity space $\mathcal{D}_{id}$ and the size of dictionary space $\mathcal{D}_{id}$ are rather limited and $\mathcal{M}$ could try all the possible passwords through an offline method (Wang et al., 2016b). This attack has been given extensive attention in the literature (Wang et al., 2016a; 2015a). As shown in Wang et al. (2018), Maitra et al. scheme (Maitra et al., 2016), an improvement of Leu–Hsieh's work scheme Leu and Hsieh (2014), suffers exactly the same issue.

5.2.3. Smart card loss attack II

We further show that a Type-I attacker $\mathcal{M}$ can obtain U_i's password PW_i via another attacking procedure. In this attack, $\mathcal{M}$ not only needs to obtain the parameters $\{B_i, Z_i, V_i, b, h(\cdot)\}$ from U_i's smart card by side channel attacks (Amiel et al., 2007; Messerges et al., 2002), but also eavesdrops the messages $\{N_i, Q_i\}$ exchanged between the parties. $\mathcal{M}$ obtains U_i's password PW_i as follows:

Step 1. $\mathcal{M}$ picks a candidate PW_i^* from the password dictionary $\mathcal{D}_{pw}$, and a candidate ID_i^* from the identity dictionary $\mathcal{D}_{id}$.
Step 2. $\mathcal{M}$ computes $R_i^* = Z_i \oplus ID_i^* \oplus h(b\|PW_i^*)$;
Step 3. $\mathcal{M}$ computes $O_i^* = h(b\|PW_i^*) \oplus ID_i^* \oplus R_i^*$;
Step 4. $\mathcal{M}$ computes $A_i^* = h(T_i\|h(y)\|N_i)$, where N_i is from the open channel and $h(y)$ from a legitimate yet curious user/server;
Step 5. $\mathcal{M}$ computes $Q_i^* = h(O_i^*\|A_i^*\|N_i)$;
Step 6. $\mathcal{M}$ examines the validity of PW_i^* by comparing if the computed Q_i^* equals Q_i which is intercepted from the open channel.
Step 7. $\mathcal{M}$ goes to Step 2 until the right PW_i is obtained.

The time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (4T_H + 3T_X))$. It can be carried out in a few days on a common computer according to the timings in Wang et al. (2015a). Note that, our attack involves the parameter $h(y)$ but not $h(x\|y)$, which is different from Maitra et al. attack (see Section 6.3 of Maitra et al., 2016). When compared with the above "smart card loss attack I", this attack is less effective as it requires that $\mathcal{M}$ colludes with a malicious insider. Still, this attack invalidates the claim of achieving truly two-factor security in Leu and Hsieh (2014).

5.2.4. No forward secrecy

A scheme providing forward secrecy is able to assure that, even if one participant's long-term key (e.g., the server's master key) has been exposed through improper erasing or leakage, the session keys that have been formerly established will not be compromised. It can be seen as the last line of defense for an authentication protocol in case the long-term secret key is (accidentally) leaked. This security feature is becoming more and more desirable as zero day attacks are prevailing (e.g., heartbleed Heartbleed – openssl zero-day bug leaves millions of websites vulnerable, 2014 and shellshock Cerrudo, 2014). As a result, new security standards like WiFi WPA3 Rescorla (2018) and TLS 1.3 Kastrenakes (2018) begin requiring forward secrecy as a feature of authentication (and key exchange) protocols.

When analyzing their scheme, Leu and Hsieh did not consider (mention) forward secrecy. We now show that this desirable property cannot be preserved: Supposing an attacker $\mathcal{M}$ manages to obtain the long-term keys $h(y)$ and $h(x\|y)$ from a compromised/malicious service server and eavesdrops the messages $\{CID_i, P_{ij}, Q_i, N_i, N_j\}$ exchanged during U_i and S authentication process from the public channel. For convenience of presentation, assume it is U_i's mth login. $\mathcal{M}$ can calculate U_i and S's session key during the jth communication as follows:

Step 1. $\mathcal{M}$ calculates $T_i = P_{ij} \oplus h(h(y)\|N_i\|SID_j)$, $A_i = h(T_i\|h(y)\|N_i))$, where $\{P_{ij}, N_i\}$ is from the open channel.
Step 2. $\mathcal{M}$ computes $h(b \oplus PW_i \oplus R_i) = CID_i \oplus h(T_i\|A_i\|N_i)$ and $O_i = h(h(b \oplus PW_i \oplus R_i)\|h(x\|y))$, where $\{CID_i, N_i\}$ is intercepted.
Step 3. $\mathcal{M}$ calculates $SK^m = h(O_i\|N_i\|N_j\|A_i\|SID_j)$, where $\{N_i, N_j\}$ is from the open channel.

Once the session key SK^m is leaked, the entire mth communication will be leaked to $\mathcal{M}$. Maitra et al. scheme Maitra et al. (2016) suffers from exactly the same issue.

6. Cryptanalysis of Zhou et al. scheme

To let the number of parameters in smart card increase linearly with the number of servers without the help of register center, Zhou et al. proposed a DLP (and bilinear-map) based authentication scheme and proved its security via Burrows–Abadi–Needham logic. However, when revisiting their scheme, we find some security flaws, including failing to withstand smart card lost attack and user/server impersonation attack. So this section will cryptanalyze Zhou et al. scheme (Zhou et al., 2018).

6.1. Review of Zhou et al. scheme

Initialization phase. Let G with a generator g and G_T be two bilinear groups of prime order p, $g_1 = e(g, g)$, $g_1 \in G_T$. RC chooses secure hash functions H_1, H_2 satisfying $H_1 : \{0, 1\}^* \to Z_p^*$, H_2: $G_T \to \{0, 1\}^*$ and a master key $SK = s (s \in Z_p^*)$, computes $PK = g^s$.

Registration. To server S_j who wants to join in the networks, it can send identity SID_j to RC, then RC will respond a private key $SK_j = g^{1/(H_1(SID_j)+s)}$. To a new user, she can register as follows:

Step 1. $U_i \Rightarrow RC$: $\{ID_i, W\}$, where $F_i = h(Bio_i)$, $W = H_1(PW_i\|F_i)$ and Bio_i is the user's biometrics.
Step 2. $RC \Rightarrow U_i$: $\{Z_i, P_i\}$.
RC computes: $k_i = g^{1/(H_1(ID_i)+s)}$, $Z_i = k_i \oplus W$, $P_i = H_1(k_i\|ID_i\|W)$.

Step 3. U_i stores $\{Z_i, P_i\}$ in the card.

Login and authentication phase. This phase proceeds as follows:

Step 1. $U_i \rightarrow S_j$: $\{ID_{RC}, C, Q, A, T\}$. U_i inputs ID_i, PW_i and $Bio_i^{'}$, then the smart card computes: $F_i^{'} = H_1(Bio_i^{'})$, $k_i^{'} = Z_i \oplus H_1(PW_i^{'}||F_i^{'})$. If $P_i \neq H_1(k_i^{'}||ID_i^{'}||H_1(PW_i^{'}||F_i^{'}))$, exit the session.
Otherwise, the smart card selects two random integers $x \in Z_p^*$ and $a \in Z_p^*$, and computes $A = g^a$, $MRU_i = (ID_{RC}||ID_i||A)$, $r_1 = g_1^x$, $C = MRU_i \oplus H_2(r_1)$, $h = H_1(MRU_i||H_2(r_1))$, and $Q = k_i^{x+h}$, $T = g^{rH_1(SID_j)}PK^r$. Note that the parameter A is not sent to S_j in Zhou et al. paper, while according to the following steps, S_j shall know this parameter.

Step 2. $S_j \rightarrow U_i$: $\{SID_j, D, B\}$.
S_j computes $r_1 = e(T, SK_j)$, $MRU_i = C \oplus H_2(r_1)$, $h = H_1(MRU_i||H_2(r_1))$. If $r_1 \neq e(Q, g^{H_1(ID_i)}PK)g_1^{-h}$, exit.
Otherwise, S_j computes: $B = g^b$, $D = H_1(A)$ and $SK_{ij} = A^b = g^{ab}$, where $b \in Z_p^*$.

Step 3. U_i checks whether $D \stackrel{?}{=} H_1(A)$, if not, terminates the session; otherwise, computes $SK_{ij} = A^b = g^{ab}$.

6.2. Flaws in Zhou et al. scheme

According to our analysis, Zhou et al. scheme is not as secure as they claimed. Here we show how an attacker conducts attacks to acquire user's password and impersonate a legitimate user/server.

6.2.1. Smart card loss attack

Once $\mathcal{M}$ obtains U_i's smart card and biometrics Bio_i, $\mathcal{M}$ can guess U_i's password PW_i as follows:

Step 1. $\mathcal{M}$ extracts $\{Z_i, P_i\}$ from U_i's smart card.
Step 2. $\mathcal{M}$ picks a candidate PW_i^* from the password dictionary $\mathcal{D}_{pw}$, and a candidate ID_i^* from the identity dictionary $\mathcal{D}_{id}$.
Step 3. $\mathcal{M}$ computes $F_i^{'} = H_1(Bio_i^{'})$;
Step 4. $\mathcal{M}$ computes $k_i^* = Z_i \oplus H_1(PW_i^*||F_i^{'})$;
Step 5. $\mathcal{M}$ computes $P_i' = H_1(k_i^*||ID_i^*||H_1(PW_i^*||F_i^{'}))$
Step 6. $\mathcal{M}$ examines the validity of PW_i^* by comparing if the computed P_i' is equal to P_i.
Step 7. $\mathcal{M}$ goes to Step 2 until the right PW_i is obtained.

The time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (3T_H + T_X))$. Based on the results in Wang et al. (2015a), this attack is able to be carried out in a few days on a common computer. This attack has been given extensive attention in the literature (Wang et al., 2016a; 2015a). Once $\mathcal{M}$ gets the password, then she can compute k_i, then further impersonate U_i.

6.2.2. Temporary information leakage attack

As said in Section 4.2.2, session-specific information is generally of large volume and deemed less sensitive than long-term secret keys, the former will be much less well protected than the latter and thus more easily leaked (e.g., through improper erasing Burmester, 1994, memory leakage Heartbleed – openssl zero-day bug leaves millions of websites vulnerable, 2014 or even poor implementations Caudill, 2016; Volgers, 2016). Consequently, it is desirable that the security impact of the leakage of such session-specific information can be limited only to session-specific secret keys, but not the long-term secret keys.

However, in Zhou et al. scheme Zhou et al. (2018), the leakage of session-specific information will put the long-term secret key k_i in danger:

Step 1. $\mathcal{M}$ somehow obtains the session-specific random number x during one session;
Step 2. Computes $r_1 = g_1^x \bmod p$.
Step 3. Computes $MRU_i = C \oplus H_2(r_1)$, where C is from open channel;
Step 4. Computes $h = H_1(MRU_i||H_2(r_1))$;
Step 5. Computes $k_i = Q^{\frac{1}{x+h}} \bmod p$.

The time complexity is about $\mathcal{O}(2T_E + 3T_H)$, which can be completed in 2.340 ms on a common PC according to the timings that $T_E \approx$ 1.169 ms and $T_H \approx 0.693\mu$ s (see Table 5 of Wang et al., 2015a). That is, the leakage of session-specific information will lead to the leakage of user identity and passwords. This is rather dangerous.

6.2.3. Poor repairability.

Zhou et al. scheme has the same repairability issue with Xu et al. scheme (see Section 3.2.2), because the user U_i's secret $k_i = g^{1/(H_1(ID_i)+s)}$ is uniquely defined by U_i's identity ID_i and RC's long-term private key d. RC is unable to update k_i for U_i when k_i is insecure, unless either ID_i or d is updated. On the other hand, even if k_i is updated to k_i', the old secret k_i will still be accepted by every service server unless the identity ID_i is blocked, because at the server side, S_j only checks if user U_m's key k_m is of the form $g^{\frac{1}{ID_m+s}}$ through verifying $r_1 \stackrel{?}{=} e(Q, g^{H_1(ID_i)}PK)g_1^{-h}$. To block ID_i, S_j shall check whether ID_i is currently valid whenever U_i logs in. To this end, S_j either needs to query the registration center RC, or maintains an updated revocation list of all revoked identities. This defeats the original goal of Zhou *et al.* to avoid interacting with the RC and maintaining/managing a large list.

7. Cryptanalysis of Roy et al. scheme

In 2019, Roy et al. (2019) proposed a provably secure user authentication scheme to achieve fine-grained data access control for cloud computing environment. Unfortunately, we reveal that their scheme suffers from the smart card loss attack and cannot provide forward secrecy.

7.1. Review of Roy et al. scheme

Initialization phase. RC selects two multiplicative cyclic groups G_1 with generator g and G_T with prime order p and a bilinear map e: $G_1 \times G_1 \rightarrow G_T$. For servers S_j, RC randomly chooses t_{aj} from $\mathbb{Z}_p$ for each attribute $a \in \mathcal{I}_j$. Then, RC chooses a random number $y_j \in \mathbb{Z}_p$, where $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, computes $Y_{S_j} = e(g,g)^{y_j} (\text{mod } p)$, and $T_{1j} = g^{t_{1j}} (\text{mod } p)$, $T_{2j} = g^{t_{2j}} (\text{mod } p)$,..., $T_{|\mathcal{I}_j|j} = g^{t_{\mathcal{I}j}} (\text{mod } p)$. Finally, RC selects random secret key K_j for each S_j, stores $\{ID_{S_j}, K_j, Y_{S_j}, \{T_{aj} = g^{t_{aj}} (\text{mod } p)\}_{\forall a \in |\mathcal{I}_j|}\}$ in S_j's database.

User Registration. This phase proceeds as follows:

Step 1. $U_i \Rightarrow RC$: $\{ID_i, (m \oplus NBPW_i\}$, where $(\eta_i, \mu_i) = Generation(Bio_i)$, $NBPW_i = h(h(\eta_i||ID_i||PW_i)||\rho)$.

Step 2. $RC \Rightarrow U_i$: $\{XId_{U_i}, (ID_{S_j}, N_{ij}, NId_{S_j}, \mathcal{P}_{ij}, \mathcal{UK}_{U_i,S_j}) | 1 \leq j \leq s\}$.
RC generates a random number x_{ij} for each pair of U_i and S_j, and computes $M_{ij} = h(h(x_{ij} \oplus ID_i)||K_j)$, $N_{ij} = M_{ij} \oplus NBPW_i$, and $NId_{S_j} = h(ID_{S_j}||K_j)$. RC selects a random number XId_{U_i} and an access structure $\mathcal{P}_{ij}$, and compute $\mathcal{UK}_{U_i,S_j}$ for each (U_i, S_j). Starting from the root node r of $\mathcal{P}_{ij}$ in a top-down manner. Then, RC generates a random polynomial q_x of degree $d_x - 1$ for each node $x \in \mathcal{P}_{ij}$, where d_x is the degree of a node x. For each non-root node $x \in \mathcal{P}_{ij}$, sets $q_x(0) = q_{parent(x)}(index(x))$, where $parent(x)$ is the parent of x, and x is the $index(x)^{th}$ child. Note that $q_r(0) = y_j$ and $\mathcal{UK}_{U_i,S_j} = \langle \{D_{S_{j,x}} = g^{\frac{q_x(0)}{t_{aj}}}\}_{x \in \mathcal{L}} \rangle$, where $\mathcal{L}$ is the set of leaf nodes and t_{aj} is the matching data attribute of S_j.

RC sends U_i a smart card with $\{XId_{U_i}, (ID_{S_j}, N_{ij}, NId_{S_j}, \mathcal{P}_{ij}, \mathcal{UK}_{U_i,S_j})|1 \le j \le s\}$ and stores (ID_i, SN_i) where SN_i is the identity of the card.

Step 3. U_i computes $P_i^1 = h(\eta_i||PW_i) \oplus \rho$, $P_i^2 = h(\rho||\eta_i||PW_i||ID_i)$, and $N_{ij}^{'} = N_{ij} \oplus m = M_{ij} \oplus h(ID_i||h(PW_i||\eta_i||\rho))$, $AID_{ij} = XId_{U_i} \oplus h(N_{ij}^{'}||ID_i)$ and $NId_{S_j}^{'} = NId_{S_j} \oplus h(\rho||\eta_i)$ for $1 \le j \le n$. Finally, U_i stores $\mu_i, P_i^1, P_i^2, N_{ij}^{'}s, AID_{ij}s$ and $NId_{S_j}^{'}$s into the card, and deletes N_{ij}s, XId_{U_i} and NId_{S_j}s.

Login and authentication phase. This phase proceeds as follows:

Step 1. $U_i \rightarrow S_j$: $\{Z_1, XId_{U_i}^*, T_{m_i}, H_1\}$. U_i inputs ID_i, PW_i and $Bio_i^{'}$, then the smart card computes: $\eta_i = Reproduction(Bio_i^{'}, \mu_i)$, $\rho^{'} = P_i^1 \oplus h(\eta_i||PW_i)$. If $P_i^2 \neq h(\rho^{'}||\eta_i||PW_i||ID_i)$, exit the session.
Otherwise, the smart card selects a random numbers RDN_i, computes: $NBPW_i = h(h(\eta_i||ID_i||PW_i)||\rho^{'})$ and $M_{ij} = N_{ij}^{'} \oplus RPBW_i$, $Z_1 = M_{ij} \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus RDN_i = h(h(x_{ij} \oplus ID_i)||K_j) \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus RDN_i$, $H_1 = h(Z_1||ID_i||T_{m_i}||RDN_i)$, $NId_{S_j} = NId_{S_j}^{'} \oplus h(\rho^{'}||\eta_i)$, $XId_{U_i} = AID_{ij} \oplus h(N_{ij}^{'}||ID_i)$, $XId_{U_i}^* = XId_{U_i} \oplus h(T_{m_i}||NId_{S_j})$, where T_{m_i} is the timestamp.

Step 2. $S_j \rightarrow U_i$: $\{Z_2, E_{S_j}, C_{ij}, H_3, T_{m_j}\}$. S_j first checks the valid of T_{m_i}, then computes $NId_{CS_j} = h(ID_{S_j}||K_j)$, $XId_{U_i} = XId_{U_i}^* \oplus h(NId_{CS_j}||T_{m_i})$, $Q_{ji} = h(h(x_{ij} \oplus ID_i)||K_j)$, $X_1 = Z_1 \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus Q_{ji} = M_{ij} \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus RDN_i \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus Q_{ji} = RDN_i$, $H_2 = h(Z_1||ID_i||T_{m_i}||X_1)$. If $H_2 \neq H_1$, exit.
Otherwise, S_j stores $\{ID_i, T_{m_i}, RDN_i\}$ to defend replay attack, and computes: $E_{S_j} = \{T_{aj}^{\alpha}\}_{\forall a \in |\mathcal{I}_j|}$, $C_{ij} = \mathcal{KS}_{ij} Y_{S_j}^{\alpha}$, where $a \in \mathbb{Z}_p$ is a random number. Then, S_j computes $Z_2 = Q_{ji} \oplus T_{m_j} \oplus RDN_j \oplus ID_i$, $SK_{S_j,U_i} = h(ID_i||ID_{S_j}||\mathcal{KS}_{ij}||Q_{ji}||X_1||RDN_j||T_{m_i}||T_{m_j})$, $H_3 = h(ID_i||E_{S_j}||C_{ij}||T_{m_i}||X_1||SK_{S_j,U_i}||T_{m_j}||RDN_j)$, where T_{m_j} is the timestamp and RDN_j is a random number.

Step 3. U_i checks T_{m_j} and computes: $X_2 = Z_2 \oplus M_{ij} \oplus ID_i \oplus T_{m_j} = Q_{ji} \oplus T_{m_j} \oplus RDN_j \oplus ID_i \oplus M_{ij} \oplus ID_i \oplus T_{m_j} = RDN_j$. Then, U_i obtains $\mathcal{KS}_{ij}$ from C_{ij} as $\mathcal{KS}_{ij} = C_{ij}(Y_{S_j}^{\alpha})^{-1}(mod p)$, then computes: $SK_{U_i,S_j} = h(ID_i||ID_{S_j}||\mathcal{KS}_{ij}||M_{ij}||RDN_i||X_2||T_{m_i}||T_{m_j})$, $H_4 = h(ID_i||E_{S_j}||C_{ij}||T_{m_i}||RDN_i||SK_{U_i,S_j}||T_{m_j}||X_2)$. If H_4 equals H_3, the authentication finishes successfully. Otherwise, U_i terminates the session.

7.2. Flaws in Roy et al. scheme

Though Roy et al. scheme Roy et al. (2019) meet the demand for fine-grained data access control, their scheme is quite complicated. What's more, their scheme does not achieve truly multi-factor authentication according to our following analysis.

7.2.1. Smart card loss attack I

An attacker $\mathcal{M}$, who obtains the smart card and biometrics, can conduct a smart card loss attack to guess the third authentication factor (i.e., password) as follows:

Step 1. $\mathcal{M}$ extracts $\{\mu_i, P_i^1, P_i^2, N_{ij}^{'}, AID_{ij}s, NId_{S_j}^{'}s, (ID_{S_j}, \mathcal{P}_{ij}, \mathcal{UK}_{U_i,S_j})|1 \le j \le s\}$ from U_i's smart card.
Step 2. $\mathcal{M}$ picks a candidate PW_i^* from the password dictionary $\mathcal{D}_{pw}$, and a candidate ID_i^* from the identity dictionary $\mathcal{D}_{id}$.
Step 3. $\mathcal{M}$ computes $\eta_i = Reproduction(Bio_i^{'}, \mu_i)$;
Step 4. $\mathcal{M}$ computes $\rho^{'} = P_i^1 \oplus h(\eta_i||PW_i^*)$
Step 5. $\mathcal{M}$ computes $P_i^{2*} = h(\rho^{'}||\eta_i||PW_i^*||ID_i^*)$;
Step 6. $\mathcal{M}$ tests the validity of PW_i^* by checking if P_i^{2*} is equal to P_i^2.
Step 7. $\mathcal{M}$ goes to Step 2 until the right PW_i is obtained.

The time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (2T_H + T_X + T_B))$, where T_B is the time for fuzzy extractor.

7.2.2. Smart card loss attack II

We further show that a Type-I attacker $\mathcal{M}$ can obtain U_i's password via another attacking procedure. In this attack, $\mathcal{M}$ not only needs to extract $\{\mu_i, P_i^1, P_i^2, N_{ij}^{'}, AID_{ij}s, NId_{S_j}^{'}s, (ID_{S_j}, \mathcal{P}_{ij}, \mathcal{UK}_{U_i,S_j})|1 \le j \le s\}$ from U_i's smart card, but also eavesdrops the messages $\{T_{m_i}, XId_{U_i}^*\}$ exchanged between the parties.

Step 1. $\mathcal{M}$ picks a candidate PW_i^* from the password dictionary $\mathcal{D}_{pw}$, and a candidate ID_i^* from the identity dictionary $\mathcal{D}_{id}$.
Step 2. $\mathcal{M}$ computes $\eta_i = Reproduction(Bio_i^{'}, \mu_i)$;
Step 3. $\mathcal{M}$ computes $\rho^{'} = P_i^1 \oplus h(\eta_i||PW_i^*)$;
Step 4. $\mathcal{M}$ computes $RBPW_i = h(ID_i^*||h(PW_i^*||\eta_i||\rho^{'}))$;
Step 5. $\mathcal{M}$ computes $M_{ij} = N_{ij}^{'} \oplus RPBW_i$;
Step 6. $\mathcal{M}$ computes $NId_{S_j} = NId_{S_j}^{'} \oplus h(\rho^{'}||\eta_i)$;
Step 7. $\mathcal{M}$ computes $XId_{U_i} = AID_{ij} \oplus h(N_{ij}^{'}||ID_i^*)$;
Step 8. $\mathcal{M}$ computes $XId_{U_i}^{'} = XId_{U_i} \oplus h(T_{m_i}||NId_{S_j})$, where T_{m_i} is from open channel;
Step 9. $\mathcal{M}$ examines the validity of PW_i^* by comparing if the computed $XId_{U_i}^{'}$ equals intercepted $XId_{U_i}^*$.
Step 10. $\mathcal{M}$ goes to Step 2 until the right PW_i is obtained.

The time complexity is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * (6T_H + 4T_X + T_B))$. When compared with the above "smart card loss attack I", this attack is less effective as it requires more computations. Nevertheless, this attack invalidates the claim of achieving truly multi-factor security in Roy et al. (2019).

7.2.3. No forward secrecy

As underlined in Wang and Wang (2018), Rescorla (2018) and Kastrenakes (2018), forward secrecy is a crucial feature for authentication schemes to limit the impact of the system breach, in situations where the attacker manages to obtain (e.g., stealing, hacking or social engineering) the long-term secret key(s) of one communication entity.

However, we find that Roy et al. scheme Roy et al. (2019) dose not provide such a security feature. Supposing an attacker $\mathcal{M}$ manages to obtain the system keys ID_{S_j}, K_j, Y_{S_j}, $\{T_{aj} = g^{t_{aj}} (\text{mod } p)\}_{\forall a \in |\mathcal{I}_j|}, ID_i, XId_{U_i}, x_{ij}\}$ from a compromised/malicious server and eavesdrops the messages $\{Z_1, XId_{U_i}^*, T_m, H_1\}$ and $\{E_{S_j}, C_{ij}\}$, then $\mathcal{M}$ can compute previous session key as follows:

Step 1. $\mathcal{M}$ calculates $NId_{CS_j} = h(ID_{S_j}||K_j)$;
Step 2. $\mathcal{M}$ calculates $XId_{U_i} = XId_{U_i}^* \oplus h(NId_{CS_j}||T_{m_i})$;
Step 3. $\mathcal{M}$ calculates $Q_{ji} = h(h(x_{ij} \oplus ID_i)||K_j)$;
Step 4. $\mathcal{M}$ calculates $X_1 = Z_1 \oplus h(ID_{S_j}) \oplus T_{m_i} \oplus Q_{ji} = RDN_i$;
Step 5. $\mathcal{M}$ calculates $X_2 = Z_2 \oplus M_{ij} \oplus ID_i \oplus T_{m_j} = RDN_j$
Step 6. $\mathcal{M}$ obtains $Y_{S_j}^{\alpha}$ as the algorithm, and computes $\mathcal{KS}_{ij} = C_{ij}(Y_{S_j}^{\alpha})^{-1}(mod p)$;
Step 7. $\mathcal{M}$ calculates the session key as $SK_{S_j,U_i} = hh((ID_i||ID_{S_j}||\mathcal{KS}_{ij}||Q_{ji}||X_1||RDN_j||T_{m_i}||T_{m_j})$.

The time complexity is $\mathcal{O}(6T_H + 8T_X + T_E)$.

8. Some lessons learned

It has been recognized that user authentication is one of the most critical cryptographic missions in cryptography research, and

the development of lightweight and secure protocols (sometimes with user anonymity) has gained extensive attention and been proved to be difficult. And the illustration of serious revealing pitfalls in previous schemes can help the research community to avoid common mistakes. As summarized in Table 2, we have shown that five representative protocols investigated are all unable to achieve the claimed important goals, and some lessons can be learned herein.

It is worth noting that, the "temporary information leakage attack" illustrated in Sections 4.2.2 and 6.2.2 seriously degrades the security of the whole system (i.e., leaking the long-term secret keys) by leaking some session-specific information. As session-specific data is generally of large volume and deemed less sensitive than long-term secret keys, the former will be much less well protected than the latter and thus more easily leaked. Therefore, this attack is rather realistic. This attack is indeed serious, because such an attacker neither needs the sensitive data in user's smart card nor needs to compromise the server, all she needs to do is to obtain some session-specific temporary information. This threat has not received sufficient attention and we highlight it here.

The attack illustrated in Section 5.2.1 highlights another serious threat, i.e., a legitimate but malicious service server – a type IV attacker (see Section 2). There are some secret parameters stored on each service server. If these parameters are all service-server-specific, the attack will not be successful. However, often it is possible for an attacker to obtain some system-wide parameters by colluding with a malicious service server. For example, the system-wide parameters $h(x\|y)$ and $h(y)$ in Leu-Hsieh's scheme (Leu and Hsieh, 2014) can be extracted by $\mathcal{M}$ from a malicious server, and with these two system-wide parameters, $\mathcal{M}$ can link the login sessions from a given user. We note that actually, the same flaw also exists in the origin of Leu–Hsieh's scheme (i.e., Lee et al. scheme Lee et al., 2011) and other schemes like (Fan et al., 2011; Lu et al., 2015a; Mishra et al., 2014; Tsai, 2008; Xu et al., 2009). Thus, the threat of type IV attackers shall not be overlooked.

In 2014, Wang et al. rigorously proved that, under the non-tamper resistance assumption of smart cards, public-key techniques are indispensable for achieving user untraceability. In the meantime, Ma et al. (2014) suggested three general principles for constructing a secure and practical two-factor authentication scheme, namely, the public-key principle, the security-usability tradeoff principle and the forward secrecy principle. The first principle states that only symmetric-key primitives are insufficient to attain two-factor security; The second principle says that there is an inevitable trade-off when providing the usability feature "local password update" and the security goal "resistance against password guessing", and a viable countermeasure is to employ the "fuzzy-verifier + honeywords" techniques as proposed in Wang and Wang (2018); The last principle suggests that forward secrecy might be achieved only when at least two exponentiations (or point multiplications) are performed at the server side (and at least one exponentiation or point multiplication is performed at the user/sender side).

It is not difficult to observe that: 1) The smart card loss attack I in Sections 5.2.2 and 7.2.1 are mainly due to the violation of Ma et al. "security-usability" trade-off principle (Ma et al., 2014); 2) The smart card loss attack II in Section 5.2.1 is mainly due to the violation of Ma et al. public-key principle Ma et al. (2014); 3) The user anonymity vulnerability reported in Section 5.2.1 is mainly due to the non-compliance with Wang et al. "publici-key principle" (Wang and Wang, 2014), because the studied scheme attempt to achieve user untraceability while only using some symmetric-key techniques; 4) The vulnerabilities reported in Sections 5.2.4 and 7.2.3 are mainly due to the non-compliance with Ma et al. "forward secrecy principle" in Ma et al. (2014), because less than two modular exponentiations are performed on the server side or less than one modular exponentiation is performed on the user side. On the other hand, it is interesting to see that all these schemes (see Luo et al., 2018; Wang et al., 2018; Wang et al., 2019b; Xiong et al., 2017) that can attain both truly multi-factor security and forward secrecy comply with these four principles. Particularly, they all employ the fuzzy-verifier technique proposed in Ma et al. (2014) and Wang et al. (2015a) and their user side carries out at least three modular exponentiations, point multiplications or Chebyshev polynomial computations.

Our results show the importance of being consistent with some basic protocol design principles when developing complex security protocols like multi-factor authentication schemes. Note that, all the four principles mentioned here are generic. This means that, besides schemes (e.g., Amin et al., 2018a; Amin et al., 2018b; Barman et al., 2019; Guo et al., 2018; Hsiang and Shih, 2009; Hsiang and Shih, 2009; Li et al., 2012; Lu et al., 2015a; Mishra et al., 2014; Yeh et al., 2013) for multi-server environments, schemes for other environments that violates these principles are also doomed to fail. For example, all the multi-factor schemes for wireless sensor networks (e.g., Amin et al., 2016; Gupta et al., 2019; Jiang et al., 2017; Liao and Wang, 2009; Ostad-Sharif et al., 2019; Shuai et al., 2019; Srinivas et al., 2017; Wazid et al., 2017; Wu et al., 2017a; Wu et al., 2017b; Xue et al., 2013) that only use some symmetric-key techniques are inherently unable to provide truly multi-factor security, forward secrecy and user anonymity.

As best illustrated in Fig. 2, how to design secure and efficient multi-factor schemes for multi-server environments is really a hard topic. We acknowledge that this paper mainly focuses on "what went wrong", and it constitutes a first step towards "how this can be done well". We only provide some guidelines at principles level. As for concrete countermeasures/schemes, this is another important research topic and we leave it as our future work.

9. Conclusion

In the past decades, considerable efforts have been spent on designing an efficient, secure and privacy-preserving multi-factor authentication scheme for multi-server environments under the security assumption that smart cards can be extracted and biometric can be spoofed. Very recently, Xu et al., Wu et al., Leu–Hsieh, Zhou et al. and Roy et al. made five new attempts. However, through systematic evaluation we reveal that all of them are still subject to various serious defects. Most importantly, our attacks underscore some new challenges (e.g., attacks arising from the leakage of session-specific information and from malicious insiders) in devising a practical multi-factor authentication scheme for multi-server environments. We also draw some lessons learned from the cryptanalysis results, and believe that they would be helpful for the community to avoid common downfalls.

Declaration of Competing Interest

None.

Acknowledgments

We thank the anonymous reviewers for their invaluable comments. This research was supported by the National Natural Science Foundation of China under Grants Nos. 61802006 and 61672059, and by China Postdoctoral Science Foundation under Grants nos. 2018M640026 and 2019T120019, and by the National Key Research and Development Plan of China under Grant no. 2017YFB1200700.

References

Amazon elastic compute cloud Amazon EC2, 2018. https://aws.amazon.com/ec2/pricing/.

Ali, R., Pal, A.K., 2018. An efficient three factor–based authentication scheme in multiserver environment using ECC. Int. J. Commun. Syst. 31 (4). e3484:1–22

Amiel, F., Feix, B., Villegas, K., 2007. Power analysis for secret recovering and reverse engineering of public key algorithms. In: Proc. SAC 2007, pp. 110–125.

Amin, R., Islam, S., Khan, M.K., Karati, A., Giri, D., Kumari, S., 2017. A two-factor RSA-based robust authentication system for multiserver environments. Secur. Commun. Netw. 2017. 5989151:1–15

Amin, R., Islam, S.H., Biswas, G.P., Khan, M.K., Leng, L., Kumar, N., 2016. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Comput. Netw. 101 (C), 42–62.

Amin, R., Islam, S.H., Gope, P., Choo, K.-K.R., Tapas, N., 2018. Anonymity preserving and lightweight multi-medical server authentication protocol for telecare medical information system. IEEE J. Biomed. Health. Inf. 23 (4), 1749–1759.

Amin, R., Kumar, N., Biswas, G., Iqbal, R., Chang, V., 2018. A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment. Future Gener. Comput. Syst. 78, 1005–1019.

Balasch, J., Gierlichs, B., Verdult, R., Batina, L., Verbauwhede, I.,. Power analysis of atmel cryptomemory–recovering keys from secure EEPROMs. In: Proc. CT-RSA 2012, pp. 19–34.

Barman, S., Shum, H.P., Chattopadhyay, S., Samanta, D., 2019. A secure authentication protocol for multi-server-based E-healthcare using a fuzzy commitment scheme. IEEE Access 7, 12557–12574.

Bellare, M., Pointcheval, D., Rogaway, P.,. Authenticated key exchange secure against dictionary attacks. In: Proc. EUROCRYPT 2000, pp. 139–155.

Burmester, M., 1994. On the risk of opening distributed keys. In: Proc. CRYPTO 1994, pp. 308–317.

Burrows, M., Abadi, M., Needham, R.M., 1989. A logic of authentication. Proc. R. Soc. Lond. A 426 (1871), 233–271.

Caudill, A., 2016. PL/SQL developer: nonexistent encryption. https://adamcaudill.com/2016/02/02/plsql-developer-nonexistent-encryption/.

Cerrudo, C., 2014. Why the shellshock bug is worse than heartbleed. https://www.technologyreview.com/s/531286/why-the-shellshock-bug-is-worse-than-heartbleed/.

Chang, C.-C., Hwang, S.-J., 1993. Using smart cards to authenticate remote passwords. Comput. Math. Appl. 26 (7), 19–27.

Chang, C.-C., Laih, C., 1992. Comment on remote password authentication with smart cards. IEE Proc.-E 139 (4), 372.

Chang, C.C., Le, H.D., 2016. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. IEEE Trans. Wirel. Commun. 15 (1), 357–366.

Chang, C.-C., Nguyen, N.-T., 2016. An untraceable biometric-based multi-server authenticated key agreement protocol with revocation. Wirel. Pers. Commun. 90 (4), 1695–1715.

Chang, C.C., Wu, T.C., 1991. Remote password authentication with smart cards. IEE Proc.-Comput. Digit. Tech. 138 (3), 165–168.

Chatterjee, S., Roy, S., Das, A.K., Chattopadhyay, S., Kumar, N., Vasilakos, A.V., 2018. Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment. IEEE Trans. Depend. Secur. Comput. 15 (5), 824–839.

Das, M.L., 2009. Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. 8 (3), 1086–1090.

Dolev, D., Yao, A., 1983. On the security of public key protocols. IEEE Trans. Inf. Theory 29 (2), 198–208.

Fan, R., He, D., Pan, X., Ping, L., 2011. An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks. J. Zhejinag Univ.-Sci. C 12 (7), 550–560.

Goodin, D., 2013. Anatomy of a hack: How crackers ransack passwords like "qeadzcwrsfxv1331". http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-out-of-your-passwords/.

Goodin, D., 2018. Lenovo fixes hard coded password and weak crypto in fingerprint manager/. https://arstechnica.com/information-technology/2018/01/lenovo-fixes-hard-coded-password-and-weak-crypto-in-fingerprint-manager/.

Guo, H., Chen, C., Gao, Y., Li, X., Jin, J., 2018. A secure three-factor multiserver authentication protocol against the honest-but-curious servers. Wirel. Commun. Mobile Comput. 2018, 1–14.

Gupta, A., Tripathi, M., Shaikh, T.J., Sharma, A., 2019. A lightweight anonymous user authentication and key establishment scheme for wearable devices. Comput. Netw. 149, 29–42.

Hackett, 2018. 150 million MyFitnessPal accounts have been hacked, under armour says. http://fortune.com/2018/03/29/myfitnesspal-password-under-armour-data-breach/.

Hackett, R., 2017. Yahoo raises breach estimate to full 3 billion accounts, by far biggest known. http://fortune.com/2017/10/03/yahoo-breach-mail/.

Hao, F., 2010. On robust key agreement based on public key authentication. In: Proc. FC 2010. In: LNCS, 6052, pp. 383–390.

He, D., Kumar, N., Khan, M.K., Wang, L., Shen, J., 2018. Efficient privacy-aware authentication scheme for mobile cloud computing services. IEEE Syst. J. 12 (2), 1621–1631.

He, D., Wang, D., 2015. Robust biometrics-based authentication scheme for multiserver environment. IEEE Syst. J. 9 (3), 816–823.

He, D., Zeadally, S., Kumar, N., Wu, W., 2016. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Trans. Inf. Forensics Secur. 11 (9), 2052–2064.

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

Heim, P., 2016. Resetting passwords to keep your files safe. https://blogs.dropbox.com/dropbox/2016/08/resetting-passwords-to-keep-your-files-safe/.

Hsiang, H.-C., Shih, W.-K., 2009. Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. Comput. Stand. Interfaces 31 (6), 1118–1123.

Huang, X., Chen, X., Li, J., Xiang, Y., Xu, L., 2014. Further observations on smart–card-based password-authenticated key agreement in distributed systems. IEEE Trans. Parallel Distrib. Syst. 25 (7), 1767–1775.

Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.H., 2011. A generic framework for three-factor authentication: preserving security and privacy in distributed systems. IEEE Trans. Parallel Distrib. Syst. 22 (8), 1390–1397.

Irshad, A., Ahmad, H.F., Alzahrani, B.A., Sher, M., Chaudhry, S.A., 2016. An efficient and anonymous chaotic map based authenticated key agreement for multi--server architecture. KSII Trans. Internet Inf. Syst. 10 (12), 5572–5595.

Irshad, A., Chaudhry, S.A., Sher, M., Alzahrani, B.A., Kumari, S., Li, X., Wu, F., 2018. An anonymous and efficient multiserver authenticated key agreement with offline registration centre. IEEE Syst. J. 13 (1), 436–446.

Islam, S., 2016. Design and analysis of an improved smartcard-based remote user password authentication scheme. Int. J. Commun. Syst. 29 (11), 1708–1719.

Jiang, Q., Ma, J., Li, G., Li, X., 2014. Improvement of robust smart-card-based password authentication scheme. Int. J. Commun. Syst.s 28 (2), 383–393.

Jiang, Q., Zeadally, S., Ma, J., He, D., 2017. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. IEEE Access 5, 3376–3392.

Kalra, S., Sood, S., 2013. Advanced remote user authentication protocol for multi--server architecture based on ECC. J. Inf. Secur. Appl. 18 (2–3), 98–107.

Karuppiah, M., Das, A.K., Li, X., Kumari, S., Wu, F., Chaudhry, S.A., Niranchana, R., 2018. Secure remote user mutual authentication scheme with key agreement for cloud environment. Mob. Netw. Appl. 1–17.

Kastrenakes, J., 2018. Wi-Fi security is starting to get its biggest upgrade in over a decade. https://www.theverge.com/circuitbreaker/2018/6/26/17501594/wpa3-wifi-security-certification.

Kim, S., Lim, S., Won, D., 2002. Cryptanalysis of flexible remote password authentication scheme of ICN'01. Electr. Lett. 38 (24), 1519–1520.

Kokalitcheva, Kia, 2016. Fingerprint spoofing is much easier than you think. http://fortune.com/2016/02/24/fingerprint-spoofing-easy/.

Kumari, S., Khan, M.K., Li, X., 2014. An improved remote user authentication scheme with key agreement. Comput. Elect. Eng. 40 (6), 97–112.

Lee, C.-C., Lin, T.-H., Chang, R.-X., 2011. A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards. Expert Syst. Appl. 38 (11), 13863–13870.

Leu, J.-S., Hsieh, W.-B., 2014. Efficient and secure dynamic id-based remote user authentication scheme for distributed systems using smart cards. IET Inf. Secur. 8 (2), 104–113.

Li, X., Niu, J., Kumari, S., Liao, J., Liang, W., 2015. An enhancement of a smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. 80 (1), 175–192.

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

Li, X., Yang, D., Zeng, X., Chen, B., Zhang, Y., 2018. Comments on "provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model". IEEE Trans. Inf. Forensics Secur. doi:10.1109/TIFS.2018.2866304.

Liao, Y.-P., Hsiao, C.-M., 2013. A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. Future Gener. Comput. Syst. 29 (3), 886–900.

Liao, Y.-P., Wang, S.-S., 2009. A secure dynamic id based remote user authentication scheme for multi-server environment. Comput. Stand. Interfaces 31 (1), 24–29.

Lin, I.-C., Hwang, M.-S., Li, L.-H., 2003. A new remote user authentication scheme for multi-server architecture. Future Gener. Comput. Syst. 19 (1), 13–22.

Lu, Y., Li, L., Peng, H., Yang, Y., 2015. A biometrics and smart cards-based authentication scheme for multi-server environments. Secur. Commun. Netw. 8 (17), 3219–3228.

Lu, Y., Li, L., Yang, X., Yang, Y., 2015. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS One 10 (5), e0126323.

Luo, M., Sun, A., He, D., Li, X., 2018. An efficient and secure 3-factor user-authentication protocol for multiserver environment. Int. J. Commun. Syst. 31 (14), e3734.

Lwamo, N.M., Zhu, L., Xu, C., Sharif, K., Liu, X., Zhang, C., 2019. Suaa: a secure user authentication scheme with anonymity for the single & multi-server environments. Inf. Sci. 477, 369–385.

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

Maitra, T., Islam, S.H., Amin, R., Giri, D., Khan, M.K., Kumar, N., 2016. An enhanced multi-server authentication protocol using password and smart-card: cryptanalysis and design. Secur. Commun. Netw. 9 (17), 4615–4638.

Mangard, S., Oswald, E., Popp, T., 2007. Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer-Verlag.

Memon, N., 2017. How biometric authentication poses new challenges to our security and privacy [in the spotlight]. IEEE Signal Process. Mag. 34 (4), 194–196.

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

Mishra, D., 2016. Design and analysis of a provably secure multi-server authentication scheme. Wirel. Pers. Commun. 86 (3), 1095–1119.
Mishra, D., Das, A.K., Mukhopadhyay, S., 2014. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst. Appl. 41 (18), 8129–8143.
Moon, J., Lee, D., Jung, J., Won, D., 2017. Improvement of efficient and secure smart card based password authentication scheme. Int. J. Netw. Secur. 19 (6), 1053–1061.
Odelu, V., Das, A.K., Goswami, A., 2015. A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Trans. Inf. Forensics Secur. 10 (9), 1953–1966.
Ostad-Sharif, A., Arshad, H., Nikooghadam, M., Abbasinezhad-Mood, D., 2019. Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme. Future Gener. Comput. Syst. 100, 882–892.
Pippal, R.S., Jaidhar, C., Tapaswi, S., 2013. Robust smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. 72 (1), 729–745.
Rescorla, E., 2018. The transport layer security (TLS) protocol version 1.3. https://datatracker.ietf.org/doc/rfc8446/?include_text=1.
Roy, S., Chatterjee, S., Das, A.K., Chattopadhyay, S., Kumar, N., Vasilakos, A.V., 2017. On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. IEEE Access 5, 25808–25825.
Roy, S., Das, A.K., Chatterjee, S., Kumar, N., Chattopadhyay, S., Rodrigues, J.J., 2019. Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. IEEE Trans. Ind. Inf. 15 (1), 457–468.
Shuai, M., Yu, N., Wang, H., Xiong, L., 2019. Anonymous authentication scheme for smart home environment with provable security. Comput. Secur. 86 (1), 132–146.
Song, J., Li, G.-s., Xu, B.-r., Ma, C.-g., 2018. A novel multiserver authentication protocol with multifactors for cloud service. Secur. Commun. Netw. 2018, 1–13.
Srinivas, J., Mukhopadhyay, S., Mishra, D., 2017. Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. Ad Hoc Netw. 54 (C), 147–169.
Stobert, E., Biddle, R., 2018. The password life cycle. ACM Trans. Privacy Secur. 21 (3), 1–32.
Tsai, J.-L., 2008. Efficient multi-server authentication scheme based on one-way hash function without verification table. Comput. Secur. 27 (3–4), 115–121.
Tsaur, W.-J., 2001. A flexible user authentication scheme for multi-server internet services. In: Proc. ICN 2001. Springer, pp. 174–183.
Tsaur, W.-J., Wu, C.-C., Lee, W.-B., 2005. An enhanced user authentication scheme for multi-server internet services. Appl. Math. Comput. 170 (1), 258–266.
Volgers, R., 2016. Exploiting two buggy SRP implementations. https://www.computest.nl/blog/exploiting-two-buggy-srp-implementations/?utm_content=buffercb32b&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer.
Wang, C., Xu, G., Li, W., 2018. A secure and anonymous two-factor authentication protocol in multiserver environment. Secur. Commun. Netw. doi:10.1155/2018/9062675.
Wang, D., Cheng, H., Wang, P., Huang, X., Jian, G., 2017. Zipf's law in passwords. IEEE Trans. Inf. Forensics Secur. 12 (11), 2776–2791.
Wang, D., Gu, Q., Cheng, H., Wang, P., 2016. The request for better measurement: a comparative evaluation of two-factor authentication schemes. In: Proc. ACM ASIACCS 2016, pp. 475–486.
Wang, D., He, D., Wang, P., Chu, C.-H., 2015. Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans. Depend. Secur. Comput. 12 (4), 428–442.
Wang, D., Wang, N., Wang, P., Qing, S., 2015. Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. Inf. Sci. 321, 162–178.
Wang, D., Wang, P., 2016. On the implications of zipf's law in passwords. In: Proc. ESORICS 2016, pp. 111–131.
Wang, D., Wang, P., 2014. On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions. Comput. Netw. 73 (C), 41–57.
Wang, D., Wang, P., 2018. Two birds with one stone: two-factor authentication with security beyond conventional bound. IEEE Trans. Depend. Secur. Comput. 15 (4), 708–772.
Wang, D., Wang, P., Wang, C., 2019. Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNS. ACM Trans. Cyber-Phys. Syst. 1–25. doi:10.1145/3325130.
Wang, D., Zhang, Z., Wang, P., 2016. Targeted online password guessing: an underestimated threat. In: Proc. ACM CCS 2016, pp. 1242–1254.
Wang, F., Xu, G., Wang, C., Peng, J., 2019. A provably secure biometrics-based authentication scheme for multiserver environment. Secur. Commun. Netw. doi:10.1155/2019/2838615.
Wang, G.-L., Yu, J.-S., Xie, Q., 2013. Security analysis of a single sign-on mechanism for distributed computer networks. IEEE Trans. Ind. Inf. 9 (1), 294–302.
Wang, P., Zhang, Z., Wang, D., 2018. Revisiting anonymous two-factor authentication schemes for multi-server environment. In: Proc. ICICS 2018. Springer, pp. 805–816.
Wang, R.-C., Juang, W.-S., Lei, C.-L., 2009. User authentication scheme with privacy-preservation for multi-server environment. IEEE Commun. Lett. 13 (2), 157–159.
Wang, S.-J., Chang, J.-F., 1996. Smart card based secure password authentication scheme. Comput. Secur. 15 (3), 231–237.
Wang, Y., Peng, X., 2015. Cryptanalysis of two efficient password-based authentication schemes using smart cards. Int. J. Netw. Secur. 17 (1), 315–322.
Wang, Y.G., 2012. Password protected smart card and memory stick authentication against off-line dictionary attacks. In: Proc. IFIP SEC 2012, pp. 489–500.
Wazid, M., Das, A.K., Kumar, N., Rodrigues, J.J., 2017. Secure three-factor user authentication scheme for renewable-energy-based smart grid environment. IEEE Trans. Ind. Inf. 13 (6), 3144–3153.
Wazid, M., Das, A.K., Odelu, V., Kumar, N., Susilo, W., 2017. Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans. Depend. Secur. Comput. doi:10.1109/TDSC.2017.2764083.
Wu, F., Xu, L., Kumari, S., Li, X., 2017. A new and secure authentication scheme for wireless sensor networks with formal proof. Peer Peer Netw. Appl. 10 (1), 16–30.
Wu, F., Xu, L., Kumari, S., Li, X., Shen, J., Choo, K.K.R., Wazid, M., Das, A.K., 2017. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. J. Netw. Comput. Appl. 89, 72–85.
Wu, F., Xu, L., Li, X., 2017. A new chaotic map-based authentication and key agreement scheme with user anonymity for multi-server environment. In: Proc. FC 2017, pp. 335–344.
Xie, Q., Wong, D.S., Wang, G., et al., 2017. Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Trans. Inf. Forensics Secur. 12 (6), 1382–1392.
Xiong, H., Tao, J., Yuan, C., 2017. Enabling telecare medical information systems with strong authentication and anonymity. IEEE Access 5, 5648–5661.
Xu, J., Zhu, W., Feng, D., 2009. An improved smart card based password authentication scheme with provable security. Comput. Stand. Interfaces 31 (4), 723–728.
Xu, Z., He, D., Huang, X., 2017. Secure and efficient two-factor authentication protocol using RSA signature for multi-server environments. In: Proc. ICICS 2017, pp. 595–605.
Xue, K., Hong, P., Ma, C., 2014. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J. Comput. Syst. Sci. 80 (1), 195–206.
Xue, K., Ma, C., Hong, P., Ding, R., 2013. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. J. Netw. Comput. Appl. 36 (1), 316–323.
Yang, G., Wong, D., Wang, H., Deng, X., 2008. Two-factor mutual authentication based on smart cards and passwords. J. Comput. Syst. Sci. 74 (7), 1160–1172.
Yao, H., Wang, C., Fu, X., Liu, C., Wu, B., Li, F., 2019. A privacy-preserving rlwe-based remote biometric authentication scheme for single and multi-server environments. IEEE Access doi:10.1109/ACCESS.2019.2933576.
Yeh, K.-H., Tsai, K.-Y., Hou, J.-L., 2013. Analysis and design of a smart card based authentication protocol. J. Zhejiang Uni. Sci. C 14 (12), 909–917.
Yu, J., Wang, G., Mu, Y., Gao, W., 2014. An efficient generic framework for three-factor authentication with provably secure instantiation. IEEE Trans. Inf. Forensics Secur. 9 (12), 2302–2313.
Zhang, R., Xiao, Y., Sun, S., Ma, H., 2017. Efficient multi-factor authenticated key exchange scheme for mobile communications. IEEE Trans. Depend. Secur. Comput. doi:10.1109/TDSC.2017.2700305.
Zhou, S., Gan, Q., Wang, X., 2018. Authentication scheme based on smart card in multi-server environment. Wirel. Netw. doi:10.1007/s11276-018-1828-7.
Zhou, Y., Yu, Y., Standaert, F., Quisquater, J., 2013. On the need of physical security for small embedded devices: a case study with COMP128-1 implemen- tations in SIM cards. In: Proc. FC 2013. In: LNCS, 7859, pp. 230–238.
Zhu, H., 2015. Flexible and password-authenticated key agreement scheme based on chaotic maps for multiple servers to server architecture. Wirel. Pers. Commun. 82 (3), 1697–1718.

Ding Wang received his Ph.D degree in Information Security at Peking University in 2017. He is currently supported by the "Boya Postdoctoral Fellowship" in Peking University, China. As the first author, he has published more than 40 papers at venues like ACM CCS, Usenix Security, NDSS, IEEE DSN, ESORICS, ACM ASIACCS, ACM TCPS, IEEE TDSC and IEEE TIFS. Seven of them are recognized as "ESI highly cited papers". His Ph.D thesis receives the "ACM China Doctoral Dissertation Award" and "China Computer Federation (CCF) Outstanding Doctoral Dissertation Award". He has been involved in the community as a TPC member for over 40 international conferences. His research interests focus on password, authentication and provable security.

Xizhe Zhang is an Undergraduate Student in School of EECS, Peking University. Currently, he is in the Intelligent Computing and Sensing Laboratory under the supervision of Professor Ping Wang and Doctor Ding Wang. His research interests focus on password authentication.

Zijian Zhang received her B.E. Degree in School of EECS from Peking University, in 2016. Now he is pursuing his Ph.D. degree in information Security at Peking University, Beijing, China. His work was covered by worldwide medias including Dailmail, Forbes, Naked security, Science Daily, ACM Technews, Microsoft, theSun, theRegister, Alphr, SCMagazine, etc. His research interests focus on password authentication.

Ping Wang received his Ph.D. degree in Computer Science from the University of Massachusetts, USA in 1996. He is currently a professor at Peking University, China. Dr. Wang has authored over 50 papers in journals or proceedings such as IEEE TDSC, ACM CCS, IEEE ICWS and IEEE CloudCom. His research interests include information security and distributed computing.