Journal of Systems Architecture 118 (2021) 102206

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Understanding security failures of anonymous authentication schemes for cloud environments

Meijia Xu [a,b], Ding Wang [a,b,*], Qingxuan Wang [a,b], Qiaowen Jia [c]

[a] College of Cyber Science, Nankai University, Tianjin 300350, China
[b] Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China
[c] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Keywords:
Cloud service
Password-based authentication
Smart card loss attack
Cryptographic analysis
Forward secrecy

ABSTRACT

Currently, password-based remote authentication mechanism has become an essential procedure to ensure users access the resources of the cloud server securely. Dozens of password-based multi-factor authentication schemes have been successively proposed recently. Unfortunately, most of them are vulnerable to various known attacks. The key to designing a secure and privacy-preserving authentication scheme is drawing some lessons from the security failures of existing schemes. In this work, we investigate three anonymous multi-factor authentication schemes based on passwords for cloud environments (i.e., Karuppiah et al.'s scheme at MONET'19, Lin's scheme at IEEE Syst J'19, Rajamanickam et al.'s scheme at IEEE Syst J'20), and demonstrate that these three schemes all suffer from off-line guessing attacks and are short of an important property (i.e., forward secrecy). We also propose several effective countermeasures to remedy these weaknesses. Our analysis shows that none of these three protocols can achieve their security goals. Furthermore, we make a summary of the causes of the flaws, and reveal that the vulnerabilities of these schemes are caused by violating the basic design principles for a secure protocol (e.g., Ma et al.'s principles at IJCS'14). In addition, we investigate whether dozens of recently proposed schemes follow the design principles of Ma et al..

1. Introduction

Nowadays, in the wake of the progress of information technology, various technologies related to the cloud are developing rapidly [1]. Cloud computing has been widely used in all walks of life because of its advantages such as broadband interconnection, shared resource pool, and flexible configuration. For enterprises, cloud computing can help reduce the maintenance costs of computing and storage. For individuals, deploying data storage and computing on cloud servers can eliminate constraints caused by limited storage resources. However, in cloud computing, due to user's data are stored in the remote cloud server, users will lose control of the data, which means that it cannot guarantee the security of the data [2,3]. Therefore, it is necessary to design the corresponding security mechanism for cloud computing to protect the confidentiality, integrity and availability of data on the cloud server. In order to achieve the above goal, Zhang et al. proposed a quantum-safe identity-based data outsourcing scheme with public integrity verification in cloud storage (DOPIV) [4]. However, sooner after that, Wang et al. [5] reveal that Zhang et al.'s mechanism is invalid because they only pay attention to the security of the data but ignore the identity authentication, which is the first line of defense to ensure the security of information systems [6,7]. Therefore, the failure of Zhang et al. proved that the identity authentication is a necessary security mechanism to ensures the security of users' data. Meanwhile, since the cloud service has the characteristics of openness and resource sharing, we also need to protect the cloud server's service to prevent unauthorized users from abusing the resources of the server. There have been a number of attacks related to the security issues mentioned above, which allow attackers to impersonate legitimate users to steal users' data or misuse cloud servers' resources [8]. Solving the security issues of cloud computing is the top priority for the continued development of the cloud computing. Consequently, the identity authentication scheme is an indispensable part of cloud services. it is essential to ensure malicious attackers and unauthorized users cannot access the services of remote cloud servers in open environments [9].

1.1. Related works

As mentioned before, the cloud environment faces the threats of malicious attackers who can eavesdrop, interrupt, intercept and modify

* Correspondence to: Ding Wang, College of Cyber Science, Nankai University, No. 38 Tongyan Road, Jinnan Distric, Tianjin 300350, China.
E-mail addresses: xumeijia@mail.nankai.edu.cn (M. Xu), wangding@nankai.edu.cn (D. Wang), wangqingxuan@mail.nankai.edu.cn (Q. Wang), jiaqw@ios.ac.cn (Q. Jia).

https://doi.org/10.1016/j.sysarc.2021.102206
Received 3 March 2021; Received in revised form 23 April 2021; Accepted 3 June 2021
Available online 8 June 2021

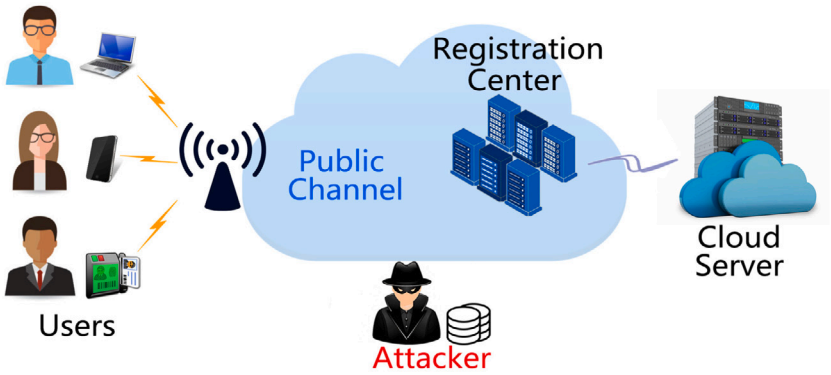

Fig. 1. Password-based authentication for the cloud services environment.

messages in the open channel, which accords to the standard Dolev-Yao model [10]. Accordingly, the designer is confronted with an impressive list of security goals (e.g., the resistance of impersonation attacks, replay attacks, and insider attacks). Most of the existing remote authentication schemes (e.g., [11,12]) for cloud environments are built on the basis of generic two-factor authentication schemes like [13,14].

In 1981, the authentication scheme based on passwords was first suggested by Lamport [15]. In this scheme, the user who owns the correct identity and password can pass the server's authentication. Since then, a series of password-based authentication protocols [16–21] sprang up. In 1991, Chang et al. [22] proposed the first smart-card-based two-factor password authentication scheme, which introduces a new authentication factor to improve the security of the authentication. However, these password-based schemes (e.g., [16,22]) are considered vulnerable after they were proposed. More specifically, they need to maintain a table to store the user's password or private parameters on the server side, which leads to stolen-verifier attacks or off-line dictionary attacks. In 1999, Yang et al. [23] proposed a new authentication scheme to overcome the above flaws. This scheme is the first two-factor scheme without a private parameters table on the server side. The security of these schemes are under the tamper-resistance assumption of the smart card.

Unfortunately, some research results have demonstrated that the security parameters stored in smart cards could be revealed or partially extracted by the state-of-the-art reverse engineering techniques [24] and side-channel attacks [25] (e.g., differential power analysis). To overcome this issue, in 2009, Xu et al. [26] proposed a two-factor authentication scheme based on smart card. They stated the scheme's security under the non-tamper resistance assumption. However, in 2010, Sood et al. [27] found that Xu et al.'s scheme cannot resist user impersonation attacks. After that, Song et al. [28] also found the weakness of Xu et al.'s scheme as mentioned before and proposed an improved scheme to fix the flaw in Xu et al.'s scheme [26]. In 2012, Chen et al. [29] revealed that both Sood et al.'s scheme [27] and Song et al.'s scheme [28] are still vulnerable. For nearly a decade, the history of the two-factor authentication has been a repeat of "break-fix-break-fix" [13]. Most scheme designers only pay attention to the modification and enhancement of a single protocol, but few designers consider the general design principles of a certain type of authentication schemes. In 2014, Ma et al. [30] proposed three vital design principles for the secure two-factor scheme based on smart cards, which has made a profound impact on the development of designing a secure scheme. Based on the design principles proposed by Ma et al. several designers have enhanced the security, efficiency, and anonymity of the two-factor authentication schemes (see Wang et al.' scheme at IEEE TDSC'18 [13], Li et al.'s scheme at IEEE TDSC' 20 [20]).

Most of the secure schemes mentioned above are suitable for single-server architecture of the cloud computing. This means that when the user log-in to different cloud servers, she needs to memorize corresponding different users identities and passwords [31] for different cloud servers, which causes a burden for users. To reduce the burden of user's memory, in 2013, yang et al. [32] proposed a two-factor authentication scheme for multi-server cloud environments. After that, a series of schemes [31,33–38] were presented to solve the usability problem of authentication in the multi-server cloud environments recent years. As shown in Fig. 1, different from single-server schemes, in addition to users and cloud servers, the register center (also called the center server) is also a part of participants. Both the servers and the users need to register in the register center. Therefore, the user only needs to memorize one identity and corresponding password to access all the servers under the control of the register center [32]. Compared with single-server architecture schemes for cloud environments, these schemes for the multi-server architecture reduce the user's memory burden [31].

1.2. Motivation and contributions

In spite of authentication schemes for cloud environments have been developed for more than 10 years, it is still far from satisfactory. The history of the research area is a designer first proposes a scheme, then other researchers reveal this scheme is insecure to various attacks and propose several possible countermeasures to fix this scheme. The research rhythm is monotonous and some common problems are repeated. The whole field is struggling to move forward. Identifying the common problems and finding ways to solve them is of great importance to the development of the field.

Recently, Karuppiah et al. [39] demonstrate that there are some security limitations in Kaul-Awasithi et al.'s scheme [40] proposed in 2016. Therefore, in 2019, Karuppiah et al. [39] proposed an improved scheme to overcome Kaul-Awasithi et al.'s [40] flaws. They claim that the proposed scheme can resist the known attacks with lower computing and storage than other related schemes and the scheme performs better compared with other related schemes. In addition, Lin [37] and Rajamanickam et al. [38] proposed two different schemes for multi-server cloud environments. Lin [37] proposes a scheme simultaneously fulfilling preauthentication and postauthentication, which is a novel feature for the authentication scheme. Meanwhile, Lin proves his scheme's security under the random oracle model. Different from Lin [37], Rajamanickam et al. [38] pay more attention to insider attacks, which is a long-standing vulnerability for cloud environments. They proposed a novel authentication protocol for insider attacks based on a robust cryptographic algorithm, ECC. In addition to insider attacks, they claim that their scheme also can resist other known attacks.

In this work, we investigate three different schemes for cloud environments recently presented by Karuppiah et al. [39], Lin [37] and Rajamanickam et al. [38]. Each of them claims to achieve their own ambitious design goals. Unfortunately, we reveal that none of them can achieve the security goals they claimed. We find all of they suffer from off-line guessing attack and cannot achieve forward secrecy. To

Table 1
Capabilities of the adversary.

C-01	$\mathcal{A}$ can enumerate the space of $\|D_{id}\| \times \|D_{pw}\|$.
C-02	$\mathcal{A}$ can obtain the victim U_i's identity ID_i.
C-1	$\mathcal{A}$ can full control of the public channel used for communication.
C-2	$\mathcal{A}$ can either (i) use malicious card readers to learn the user's password, or (ii) use side-channel attacks to extract the secret data in the card.
C-3	$\mathcal{A}$ can learn the previous session key(s) between the user and the server.
C-4	$\mathcal{A}$ can obtain the server's long-time private key(s) and other data stored in the server only when analyzing the forward secrecy or the root cause of the server's failure (e.g., known key attacks).

learn some lessons, we discuss the causes of the pitfalls and propose several possible countermeasures to overcome them. Last but not least, we analyze many different schemes proposed recently [37–47] and discuss the three basic principles of designing a secure scheme at Ma et al. [30]. We believe this work will spark interest for the designers to propose a password-based authentication scheme that achieved all essential security goals.

1.3. Organization

The rest of the paper is organized as follows: The capabilities of the adversary $\mathcal{A}$ are introduced in Section 2. The cryptanalysis of the scheme of Karuppiah et al. [39] is given in Section 3. And the cryptanalysis of the scheme of Lin [37] is given in Section 4. Section 5 gives the cryptanalysis of Rajamanickam et al.'s [38] scheme. The Section 6 discusses the causes of the pitfalls and Ma et al.'s three basic principles for protocol design. And Section 7 gives a conclusion.

2. Adversary models

The adversary models we adopted are presented in [13], which is harsh but reasonable and widely accepted (see [48–51]). We also summarize the capabilities of the adversary $\mathcal{A}$ in Table 1. We give a simple explanation of the capabilities.

As usual, the user's identity can be obtained in public channels, which always has a unified fixed format or chosen by the user randomly. Meanwhile, recently, Wang et al. [52,53] reveal that the password chosen by the user follows the Zipf's law. That means the space of passwords $|D_{pw}|$ is extremely limited in practice. They usually assume that $|D_{id}| \leq |D_{pw}| \leq 2^{20} \approx 10^6$. Therefore, it is reasonable to set the capability C-01 that $\mathcal{A}$ can enumerate the space of identity and password effectively in polynomial time.

Specially, we suppose the capability C-02 separately that $\mathcal{A}$ can determine the victim U_i's identity ID_i. A scheme is usually required to provide user anonymity, which means that $\mathcal{A}$ neither knows which user this intercepted message comes from nor whether two messages are from the same user [54]. However, if $\mathcal{A}$ desires to launch the impersonation attack, she can obtain the victim U_i's identity ID_i by non-cryptographic methods (e.g., shoulder surfing). Thus, it is not contradictory for our supposed C-02 and the security goal of user anonymity.

Since the transmission of the message is in the insecure channel, C-1 is a common capability for $\mathcal{A}$. She can eavesdrop, intercept, insert, delete, and modify any transmitted messages over the public channel [55,56], which accords the Dolev–Yao model [10]. What is more, "Two-factor security" means that the users who have to know both two factors can be authenticated successfully. Therefore, the capability C-2 is usually used to analyze smart-card-based two-factor authentication protocols.

Last but not least, the capabilities C-3 and C-4 are presented to analyze the forward secrecy. It is common that $\mathcal{A}$ can know the previous session key because of the insecure behaviors of insiders and users (e.g., delete incorrectly). What is more, when we evaluate the notion of forward secrecy, it is a convention that supposing $\mathcal{A}$ can obtain the server's long-time private keys [13,57,58]. The capabilities of adversary $\mathcal{A}$ are indeed reasonable. Our analysis of three schemes will be based on these capabilities.

Table 2
Notations and abbreviations.

Symbol	Description
U_i	The ith user
S	Cloud server
$\mathcal{A}$	Malicious attacker
ID_i	The identity of U_i
PW_i	The password of U_i
x_s	The private key of cloud server
T_h	The running time of hash function
T_{xor}	The running time of bitwise XOR operation
$\oplus$	The XOR operation
$\otimes$	The NOR operation
$\|\|$	The concatenation operation
$h(\cdot)$	One-way hash function
$\rightarrow$	The public channel
$\Rightarrow$	The secure channel

3. Cryptanalysis of Karuppiah et al.'s scheme

Karuppiah et al.'s [39] scheme is an enhancement over Kaul et al.'s scheme [40]. Karuppiah et al. prove that their scheme is invulnerable to various attacks by using both rigorous formal and informal security proof. However, after our analysis, it is clear that the scheme cannot achieve its security goals.

3.1. Review of Karuppiah et al.'s scheme

We will briefly review Karuppiah et al.'s scheme [39]. This scheme consists of five phases: initialization, registration, log-in, authentication and password change. We simplify the initialization phase in the registration phase. For ease of description, some intuitive notations and abbreviates are listed in Table 2 and will be used through-out this paper.

3.1.1. Registration phase

First, the parameters $\{g, y, p\}$ are public parameters. And the cloud server has her own private key x_s. Then S gets public key $y = g^{x_s} \pmod p$ using the generator g.

(1) User U_i freely selects identity ID_i and password PW_i. Then the user selects a random nonce k and uses k to compute $HPW = h(PW_i \| k)$.

(2) $U_i \Rightarrow S : \{HPW, ID_i\}$.

(3) Then the server S uses received message $\{HPW, ID_i\}$ to compute $A_i = h(ID_i \oplus x_s)$ and $B_i = A_i \oplus h(ID_i \| HPW)$. And the cloud server S uses a smart card SC to store parameters $\{B_i, g, y, p, h(\cdot)\}$.

(4) $S \Rightarrow U_i$: SC.

(5) After receiving the SC, U_i computes $N_i = k \oplus h(ID_i \oplus PW_i)$ and $N_t = k \otimes ID_i \otimes PW_i$. Then she stores $\{N_i, N_t\}$ in the SC. Finally, the smart card contains $\{B_i, g, y, p, h(\cdot), N_i, N_t\}$.

3.1.2. Log-in phase

(1) User U_i enters her ID_i and corresponding PW_i.

(2) Then smart card SC retrieves $k = N_i \oplus h(ID_i \oplus PW_i)$ and use k to get $N_t^* = k \otimes ID_i \otimes PW_i$.

(3) After that, the SC gets $A_i = B_i \oplus h(ID_i \| h(PW_i \| k))$ and $W_i = h(T_u \oplus A_i) \oplus (r_i \oplus k)$ where r_i is a random number . Then the SC can also obtain $C_i = g^{(r_i \oplus k)}$, $DID = ID_i \oplus h(y^{(r_i \oplus k)})$. And the smart card SC uses the present timestamp T_u to computes $V_i = h(ID_i \| A_i \| W_i \| (r_i \oplus k) \| T_u)$.

(4) $SC \rightarrow S : \{C_i, W_i, V_i, DID, T_u\}$.

3.1.3. Authentication phase

(1) After receiving the message, S uses her timestamp T_s to verify $(T_s - T_u) \leq \Delta T$. If T_u is invalid, then the S rejects the message.

(2) Then, S computes V_i^* and verifies $V_i^* \stackrel{?}{=} V_i$. If the verification is true, S authenticates U_i. S computes $h(ID_i \| (A_i \oplus (r_i \oplus k))) = G_i$. Then S uses G_i to get $M_i = h(G_i \| T_s)$.

(3) $S \rightarrow U_i : \{M_i, T_s\}$.

(4) Upon receiving the message $\{M_i, T_s\}$, time stamp T_s need to be verified. SC computes $G_i^* = h(ID_i \| (A_i \oplus (r_i \oplus k)))$. The SC uses G_i^* to compute $M_i^* = h(G_i^* \| T_s)$. SC verifies $M_i^* \stackrel{?}{=} M_i$. If it holds, U_i authenticates S.

(5) Next, the smart card and the server calculate $SK_u = h(ID_i \| A_i \| (r_i \oplus k) \| (T_u \oplus T_s)) = SK_s$ separately for later communications.

3.1.4. Password change phase

(1) The ID_i, PW_i and the new password PW_i^n are entered by U_i and then the user initiates a request to change the password.

(2) SC computes $k = N_i \oplus h(ID_i \oplus PW_i)$ and $HPW_i = h(k \| PW_i)$. Then it computes $N_t^* = k \otimes ID_i \otimes PW_i$.

(3) SC checks whether $N_t^* \stackrel{?}{=} N_t$. If $N_t^* \neq N_t$, SC ends the session; Otherwise, the smart card gets $B_i^n = B_i \oplus h(ID_i \| h(PW_i \| k)) \oplus h(ID_i \| h(PW_i^n \| k)) = A_i \oplus h(ID_i \| HPW_i^n)$. Then the SC computes $N_i^n = N_i \oplus h(ID_i \oplus PW_i) \oplus h(ID_i \oplus PW_i^n) = k \oplus h(ID_i \oplus PW_i^n)$, $N_t^n = k \otimes ID_i \otimes PW_i^n$. Lastly, SC replaces $\{B_i, N_i, N_t\}$ with $\{B_i^n, N_i^n, N_t^n\}$.

3.2. Cryptanalysis of Karuppiah et al.'s scheme

Karuppiah et al. [39] claim that their scheme can provide the session key (SK) security and resist the known attacks with lower computing and storage than other related schemes. However, we reveal that the scheme cannot achieve forward secrecy. What is more, it is also vulnerable to smart card loss attacks and insider attacks.

3.2.1. Smart card loss attack

Firstly, we suppose that $\mathcal{A}$ can obtain U_i's smart card and extract the parameters $\{B_i, g, y, p, h(\cdot), N_i, N_t\}$ using side-channel attack [59,60]. What is more, the adversary can also intercept U_i's log-in message $\{C_i, W_i, V_i, DID, T_u\}$, then she can obtain PW_i as follows:

Step 1. $\mathcal{A}$ chooses ID_i^*, PW_i^* from $\mathcal{D}_{id} \times \mathcal{D}_{pw}$.
Step 2. $\mathcal{A}$ computes $k^* = N_i \oplus h(ID_i^* \oplus PW_i^*)$.
Step 3. $\mathcal{A}$ computes $A_i^* = B_i \oplus h(ID_i^* \| HPW^*) = B_i \oplus h(ID_i^* \| h(PW_i^* \| k^*))$.
Step 4. $\mathcal{A}$ computes $(r_i \oplus k)^* = W_i \oplus h(T_u \oplus A_i^*)$.
Step 5. $\mathcal{A}$ computes $V_i^* = h(ID_i^* \| A_i^* \| W_i \| (r_i \oplus k)^* \| T_u)$.
Step 6. $\mathcal{A}$ check whether $V_i^* = V_i$ to verify the correctness of (ID_i^*, PW_i^*) pair.
Step 7. $\mathcal{A}$ repeats steps 1–6 until above equation holds, then the right values are found.

The time complexity of the attack procedure is $\mathcal{O}((5T_h + 5T_{xor}) \times |\mathcal{D}_{id}| \times |\mathcal{D}_{pw}|)$. It is clear that the time for $\mathcal{A}$ to recover U_i's password is a linear function of $|\mathcal{D}_{pw}| \times |\mathcal{D}_{id}|$. Nowadays, to meet user-friendliness, the users are allowed to choose ID and PW by themselves in most authentication schemes (e.g., [44,61]). Karuppiah et al.'s scheme [39] also like the above principle. However, users usually choose the ID and PW which are low entropy and easy-to-remember. Since the password follows the Zipf's law [62] and the dictionary size is very restricted, usually $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| \leq 2^{20} \approx 10^6$. It is reasonable that $\mathcal{A}$ can offline enumerate all the (ID, PW) pairs in polynomial time.

As it is known that the inherent cause of off-line password guessing attack is that $\mathcal{A}$ can determine the correctness of the guess parameters by using a verifier. Specially, for the adversary $\mathcal{A}$, the verifier only contains one unknown parameter (i.e., password) which means all unknown parameters of the verifier can be derived from the password and known parameters [63]. Ma et al. [30] proposed that the public-key techniques are necessary to resist off-line password guessing attack. It is obvious that the verifier is V_i in the above attack. Thus, we set $R_1 = y^{r_1} \bmod p$, $R_0 = g^{r_1} \bmod p$, where y is server's public key and r_1 is a random number . We reset $V_i = h(ID_i \| A_i \| W_i \| (r_i \oplus k) \| T_u \| R_1)$. Then the user sends message $\{C_i, W_i, V_i, DID, T_u, R_0\}$ to the server. When the server receives the message, she uses her secret key x_s to calculate $R_1^* = R_0^{x_s} = g^{r_1 x_s} \bmod p$. Then the server gets other parameters as the original scheme. She can authenticate user's identity by calculating $V_i^* = h(ID_i \| A_i \| W_i \| (r_i \oplus k) \| T_u \| g^{r_1 x_s})$.

What is more, after the server authenticates use's identity, she will calculate G_i, M_i. The M_i in original Karuppiah et al.'s scheme [39] can also be used to apply off-line password guessing attacks. The attack steps are similar to the above attack. Therefore, we also reset $G_i = h(ID_i \| (A_i \oplus (r_i \oplus k)) \| R_2)$ and $M_i = h(G_i \| T_s)$, where r_2 is a random number and $R_2 = g^{r_1 r_2} \bmod p$. Then the server gets $R_3 = g^{r_2} \bmod p$ and sends $\{M_i, T_s, R_3\}$ to the user. Then the SC computes $G_i^* = h(ID_i \| (A_i \oplus (r_i \oplus k)) \| R_3^{r_1})$, $M_i^* = h(G_i^* \| T_s)$. The SC ensures server's identity by verifying $M_i^* \stackrel{?}{=} M_i$.

After that, if the adversary $\mathcal{A}$ desire to use off-line guessing attack to candidate user's password. She needs to get a verifier to confirm the correctness of the password [63]. However, the attacker cannot use (ID, PW) she guessed to represent V_i or M_i. The adversary cannot obtain r_1, r_2 by R_0, R_3 because it is a computational Diffie–Hellman problem [64] which cannot be solved in polynomial time. Consequently, our improvement measure is available and secure.

3.2.2. Insider attack

Suppose $\mathcal{A}$ is a malicious server administrator. $\mathcal{A}$ can intercept U_i's registration message $\{ID_i, HPW\}$, and she can also stole or pick up U_i's smart card and extract $\{B_i, g, y, p, h(\cdot), N_i, N_t\}$. The malicious administrator can compute U_i's password by the following steps:

Step 1. $\mathcal{A}$ chooses PW_i^* from $\mathcal{D}_{pw}$.
Step 2. $\mathcal{A}$ computes $k^* = N_i \oplus h(ID_i \oplus PW_i^*)$.
Step 3. $\mathcal{A}$ computes $HPW^* = h(PW_i^* \| k^*)$.
Step 4. $\mathcal{A}$ verifies whether $HPW^* = HPW$.
Step 5. $\mathcal{A}$ repeats the steps $1 \sim 4$ until above equation holds.

In fact, many websites and applications are attacked because of malicious server administrators, which causes the disclosure of user personal information. Therefore, we can reasonably assume that attackers can intercept or internally extract U_i's registration information. The time complexity of this attack is $\mathcal{O}((2T_h + 2T_{xor}) \times |\mathcal{D}_{pw}|)$. The attacker can achieve this attack in polynomial time.

In this scheme, the verifier HPW contains two parameters PW_i and k. Unfortunately, k can be computed using the parameters N_i. A possible countermeasure to resist this attack is that the user updates k after registration. When the server S sends smart card to the U_i, the U_i can select a new nonce k'. Then updating $HPW' = h(PW_i \| k')$ and $B_i = A_i \oplus h(ID_i \| HPW')$. What is more, $N_i = k' \oplus h(ID_i \oplus PW_i)$ and $N_t = k' \otimes ID_i \otimes PW_i$. Then the user stores $\{N_i, N_t\}$ in the SC.

When we adapt this countermeasure, the malicious administrator cannot launch above attack. The malicious insider could get N_i and HPW as mentioned before. If she desires to verify the correctness of PW_i, she needs to get a parameter k. And then she calculates $HPW^* = h(PW_i^* \| k^*)$. However, in our countermeasure, we update new parameter k' to calculate N_i after the user's registration. Therefore, the adversary cannot get parameters k by using the PW_i she guessed. The attacker cannot get the user's password or other related parameters, so our countermeasure can resist insider attack.

Table 3
Notations and abbreviations.

Symbol	Description
SA	The system authority
UID_i	The username of ith user
ST	The security token
x_s	The private key of cloud server
Y_s	The public key of cloud server
K_{sa}	The master key of system authority
T_me	The running time of modular exponentiation
$H(\cdot)$	One-way hash function
$F(\cdot)$	One-way hash function
$E(\cdot)/D(\cdot)$	Symmetric encryption/decryption function

3.2.3. No forward secrecy

We claim that $\mathcal{A}$ has the capabilities C-02 and C-4. When analyzing the forward security of the protocol, we suppose that $\mathcal{A}$ knows S's long-time secret key x_s. And $\mathcal{A}$ can obtain the messages $\{C_i, W_i, V_i, DID, T_u\}$ and $\{M_i, T_s\}$. According to the following steps, $\mathcal{A}$ can get the session key.

Step 1. $\mathcal{A}$ computes $A_i = h(ID_i, x_s)$.
Step 2. $\mathcal{A}$ computes $(r_i \oplus k) = W_i \oplus h(T_u \oplus A_i)$.
Step 3. $\mathcal{A}$ computes
$SK = h(ID_i \parallel A_i \parallel (r_i \oplus k) \parallel (T_u \oplus T_s))$.

The time complexity of this attack is $\mathcal{O}((3T_h + 3T_{xor}) \times |\mathcal{D}_{pw}|)$. Moreover, in the above attack, we suppose that U_i's ID_i can be known by some non-cryptographic methods, because the user's ID_i is regarded as a kind of public information on some websites (e.g., telephone number, social security account). Thus, it is reasonable that $\mathcal{A}$ can calculate the session key in polynomial time. In a word, Karuppiah et al.'s scheme [39] cannot attain forward secrecy.

Following Ma et al.'s principle [30], to resist this attack, the public key technique and two modular exponentiations are required on the server side. We can set $R_1 = y^{r_1} \bmod p$ and $R_0 = g^{r_1} \bmod p$, where r_1 is randomly selected by the user. Then the user replaces message $\{C_i, W_i, V_i, DID, T_u\}$ by message $\{C_i, W_i, V_i, DID, T_u, R_0\}$ and sends to server. The server computes $R_2 = g^{r_2 r_1} \bmod p$ and $R_3 = g^{r_2} \bmod p$, where r_2 is a random number. She sends $\{M_i, T_s, R_3\}$ to the user. Lastly, the users computes $SK_u = h(ID_i \parallel A_i \parallel (r_i \oplus k) \parallel (T_u \oplus T_s) \parallel {R_3}^{r_1})$. And the server computes $SK_s = h(ID_i \parallel A_i \parallel (r_i \oplus k) \parallel (T_u \oplus T_s) \parallel R_2)$. As such, the adversary cannot compute r_1 or r_2 within polynomial time (it is a computational Diffie–Hellman problem). Then the attacker cannot get the session key. Therefore, our method is useful to protect the scheme's forward secrecy.

4. Cryptanalysis of Lin's scheme

Lin [37] proposes an anonymous two-factor authentication scheme. Lin claims that his scheme allows a user to choose a pseudo-identity for remote authentication in cloud environments. What is more, he claims the scheme can achieve user anonymity. Although the scheme is different from traditional schemes, we also find it is vulnerable to smart card loss attack and cannot achieve forward secrecy.

4.1. Review of Lin's scheme

We briefly review Lin's [37] scheme, which involves four phases: setup, user registration, authentication and password update. We simplify the setup phase in the registration phase. the intuitive abbreviations are listed in Table 2 and some additional ones in Table 3. In this scheme, some used notations are different. The username UID_i is randomly selected by the user (e.g., a nickname) and ID_i is a fixed format number used to represent the user's identity.

4.1.1. Registration phase

Initially, the system authority SA owns a master secret key K_{sa}. She first generates p, q and a generator g of order q. The client U_i's identity is ID_i, which is usually a fixed parameter assigned by the system. U_i has her own username UID_i and corresponding password PW_i. Then the user U_i will register to S through the system authority SA.

(1) U_i computes $e_i = H(PW_i \oplus r_i, UID_i, ID_{cs})$ and $\delta_i = H(PW_i \oplus r_i, UID_i, ID_i)$, where r_i is a nonce selected from Z_p.
(2) $U_i \Rightarrow SA : \{ID_i, e_i\}$.
(3) If ID_i has not registered before, the SA generates a pseudo-identity for ID_i as $PID_i = E_{K_{sa}}(ID_i \oplus Y_{cs})$.
(4) $SA \Rightarrow S : \{PID_i, e_i\}$.
(5) Then S computes $k_i = F(PID_i, x_s)$, $\sigma_i = k_i \parallel Y_s$, $\tau_i = \sigma_i \oplus e_i$, $\eta_i = PID_i \oplus e_i \oplus H(x_s)$, and stores (η_i, Y_s, τ_i) in a security token hardware ST_i.
(6) $S \Rightarrow U_i : ST_i$.
(7) Finally, U_i further stores (r_i, δ_i) into ST_i.

4.1.2. Authentication phase

(1) U_i first enters UID_i and PW_i to the security token ST_i, then ST_i calculates $e_i = H(PW_i \oplus r_i, UID_i, ID_{cs})$, $\sigma_i = \tau_i \oplus e_i = k_i \parallel Y_s'$, and checks if $Y_s' = Y_s$. Otherwise, it aborts.
(2) Next, ST_i chooses a random number $a \in_R Z_q$. Then ST_i uses a to compute $\beta = (Y_{cs}')^a \bmod p$, $CID_i = \eta_i \oplus e_i$, $c_1 = \beta \oplus CID_i$, $c_0 = g^a \bmod p$, $d_i = H(CID_i, c_0, \beta, T_1)$, and $c_2 = d_i \oplus k_i$.
(3) $U_i \rightarrow S : \{c_0, c_1, c_2, T_1\}$.
(4) S first checks whether T_1 is fresh or valid. S further computes $\beta' = {c_0}^{x_s} \bmod p$, $CID_i' = \beta' \oplus c_1$, $d_i' = H(CID_i', c_0, \beta', T_1)$, $k_i' = F(CID_i', x_s)$, $c_2' = d_i' \oplus k_i'$ and verifies if $c_2' = c_2$; Else, S denies the log-in request. Then, S chooses $b \in_R Z_q$ to compute $VK = H(d_i', k_i' \oplus b, ID_{cs})$, $c_3 = VK \oplus \beta'$.
(5) $S \rightarrow U_i : \{c_3, b\}$.
(6) U_i derives $VK' = H(d_i, k_i \oplus b, ID_{cs})$ and verifies the equation of $c_3 = VK' \oplus \beta$.

4.1.3. Password-update

The user U_i first enters ID_i, UID_i along with passwords (PW_i, PW_i') to the security token ST_i. After that, ST_i computes $e_i' = H(PW_i \oplus r_i, UID_i, ID_{cs})$, $e_i^* = H(PW_i' \oplus r_i, UID_i, ID_{cs})$, and $\delta_i^* = H(PW_i \oplus r_i, UID_i, ID_i)$. Then ST_i checks whether $\delta_i^* = \delta_i$; Next, ST_i can update the stored (τ_i, δ_i) as $\tau_i' = \tau_i \oplus e_i \oplus e_i^*$, $\delta_i' = H(PW_i' \oplus r_i, UID_i, ID_i)$.

4.2. Cryptanalysis of Lin's scheme

Lin [37] shows that the proposed scheme is invulnerable to various attacks together with attacks observed in the analyzed scheme and proves the security of the proposed scheme in the random oracle model. However, we use reasonable and widely accepted adversary models to analyze this scheme and find that it cannot resist smart card loss attacks and cannot achieve forward secrecy.

4.2.1. Smart card loss attack

Suppose $\mathcal{A}$ has obtained user U_i's security token hardware ST_i and uses side-channel attacks to extract $(\eta_i, Y_s, \tau_i, r_i, \delta_i)$. Then she can obtain PW_i following the steps blow:

Step 1. $\mathcal{A}$ chooses UID_i^*, PW_i^* from $\mathcal{D}_{uid} \times \mathcal{D}_{pw}$.
Step 2. $\mathcal{A}$ computes $\delta_i^* = H(PW_i^* \oplus r_i, UID_i^*, ID_i)$.
Step 3. $\mathcal{A}$ verifies whether $\delta_i^* = \delta_i$.
Step 4. $\mathcal{A}$ repeats step 1–3 until equation holds.

In this scheme, user's ID_i is a fixed number represented user's identity (e.g., ID Card number). As usual, we assume that ID_i is a known parameter because $\mathcal{A}$ can get user's identity through public channels, such as hacker websites, etc. The time complexity of this attack is $\mathcal{O}(|D_{uid}| \times |D_{pw}| \times (T_h + T_{xor}))$. Therefore, $\mathcal{A}$ can lunch above attack in polynomial time.

The possible countermeasure is that we reset $\delta_i = H(PW_i \oplus r_i, UID_i, ID_i) \pmod{n_0}$, where n_0 is an integer and $n_0 \in (2^4, 2^8)$. After that, $\mathcal{A}$ can candidate $|D_{uid} \times D_{puw}|/n_0 \approx 2^{32}$ pairs of (UID_i, PW_i), which satisfy the equation $\delta_i = \delta_i^*$. However, only one pair of (UID_i, PW_i) is correct. In order to get the correct one, the adversary has to initiate the log-in request to the server, which will be detected and rejected by the server. Because the number of incorrect user identity and password is limited (generally 5–10 times). Thus, $\mathcal{A}$ usually cannot get the correct (UID_i, PW_i) from 2^{32} pairs of (UID_i, PW_i) within 10 times, so she cannot launch smart loss attacks in this scheme.

4.2.2. No forward secrecy

Suppose $\mathcal{A}$ can obtain the cloud server ID_{cs}'s long-time secret key x_s. She also intercepts user log-in message $\{c_0, c_1, c_2, T_1\}$ and server response $\{c_3, b\}$. Then she can obtain previous session key VK as follows:

Step 1. $\mathcal{A}$ computes $\beta' = c_0^{x_s} \pmod{p}$.
Step 2. $\mathcal{A}$ computes $VK = c_3 \oplus \beta'$.

Suppose $\mathcal{A}$ obtains previous session key VK_i, then she can compute the jth session key VK_j.

Step 1. $\mathcal{A}$ computes $\beta_i = c_{3i} \oplus VK_i$.
Step 2. $\mathcal{A}$ computes $CID_i = CID_j = c_{1i} \oplus \beta_i$.
Step 3. $\mathcal{A}$ computes $\beta_j = CID_j \oplus c_{1j}$.
Step 41. $\mathcal{A}$ computes $VK_j = \beta_j \oplus c_{3j}$.

The time complexity of attack ∞ is $\mathcal{O}(T_{me} + T_{xor})$. And the time complexity of attack 2 is $\mathcal{O}(4T_{xor})$. Despite Lin's scheme [37] has paid attention to forward secrecy issue, it still violates the "PFS principle" suggested in Ma et al. [30]. There is only one exponentiation operation on the server side. Accordingly, similar to Section 3.2.3, we assume r_1, r_2 are two random numbers selected separately by the server and the user. The user computes $c_4 = g^{r_1} \bmod p$ and the server gets $c_5 = g^{r_2} \bmod p$. Then they send c_4, c_5 to each other. The session key VK can be recalculated as $VK' = H(d_i, k_i \oplus b, ID_{cs}, g^{r_1 r_2})$. After slight modification, Lin's scheme [37] can provide forward secrecy and resist above attack successfully.

5. Cryptanalysis of Rajamanickam et al.'s scheme

In 2020, Rajamanickam et al. [38] propose a lightweight password-based single-factor authentication scheme for Internet applications, which claimed to resist insider attacks and other security issues. However, we reveal that their scheme is not actually resistant to insider attacks. What is more, it also cannot achieve forward secrecy and user anonymity.

5.1. Review of Rajamanickam et al.'s scheme

We will describe Rajamanickam's scheme proposed in 2020 [38], which claims to resist insider attack and achieve their claimed security goals. The scheme includes three main phases: initial setup, registration, log-in and authentication. For ease of description, some intuitive notations and abbreviates are listed in Tables 4 and 2.

5.1.1. Initial setup

SPS first sends its identity ID_{SPS} to the PMS through an insecure channel. Then PMS generates a master secret key MSK_j, gets public key $P_{pub} = MSK_j \cdot P$ and sends MSK_j to the SPS in a secure channel. Then PMS stores (ID_{SPS}, MSK_j) in a table encrypted by PMS's secret key s_{PMS}. And the corresponding public key of PMS's secret key is $pub_{PMS} = s_{PMS} \cdot P$.

Table 4
Notations and abbreviations.

Symbol	Description
SPS	Service provider server
PMS	Password management server
ID_{SPS}	The identity of SPS
MSK_j, P_{pub}	Master secret key, public key of SPS
pub_{PMS}, s_{PMS}	Public key, secret key of PMS
T_{mul}	The running time of ECC multiplication
$T_e cc$	The running time of decryption in ECC
$E(\cdot)$	ECC encryption
$AES(\cdot)$	AES encryption

5.1.2. Registration phase

When the client U_i desires to enjoy the services of the SPS, she need to register first. (1) U_i chooses the username ID_i and the password PW_i. Then she enters (ID_i, PW_i) to the web browser. The browser gets $E_{pub_{PMS}}(ID_i, PW_i)$ by using PMS's public key pub_{PMS}.

(2) $U_i \rightarrow$PMS : $E_{pub_{PMS}}(ID_i, PW_i)$.

(3) When receiving the message, the PMS decrypts and computes $A_i = h((h(ID_i) \oplus h(PW_i)) \bmod n)$ and $B_i = h((h(ID_{SPS}) \oplus h(A_i)) \bmod n)$. Then she encrypts A_i and B_i by MSK_j to make ciphertext $C_i = AES_{MSK_j}(A_i, B_i, TID_i)$, where $TID_i = h(ID_i)$. The PMS maintains different tables for different SPSs. Every table stores data $TID_i = h(ID_i)$, A_i and $E_{s_{PMS}}(PW_i)$.

(4) PMS $\rightarrow$SPS : $\{C_i, n\}$.

(5) PMS $\rightarrow U_i$: a confirmation message .

(6) The SPS decrypts message to get A_i, B_i and TID_i. She computes $B_i^* = h((h(ID_{SPS}) \oplus h(A_i)) \bmod n)$. After that, SPS checks $B_i^* \stackrel{?}{=} B_i$. If it do not holds, the SPS will end the conversation. Otherwise, the SPS stores TID_i and A_i in a secure manner.

5.1.3. Log and authentication phase

(1) U_i first provides ID_i and PW_i to the web browser. Then the browser gets $A_i^* = h((h(ID_i) \oplus h(PW_i)) \bmod n)$, $TID_i^* = h(ID_i)$. After that, the web browser chooses the nonce r_1 and computes $D_1 = r_1 \cdot P$, $E_i = h(ID_{SPS} \parallel D_1 \parallel A_i \parallel T_1)$, where T_1 is a timestamp. Moreover, U_i encrypts TID_i^* by SPS's public key P_{pub} and computes $F_i = E_{P_{pub}}(TID_i^*)$, $G_i = TID_i^* \oplus r_1$.

(2) $U_i \rightarrow$SPS: $\{E_i, F_i, G_i\}$.

(3) The SPS verifies $| T_1 - T_2 | \leq \Delta T$ firstly. If it holds, the SPS computes TID_i^* by using MSK_j. Then the SPS searches the table and obtains A_i. Then the SPS computes $r_1^* = G_i \oplus TID_i^*$, $D_1^* = r_1^* \cdot P$, $E_i^* = h(ID_{SPS} \parallel D_1^* \parallel A_i \parallel T_1)$, and checks whether $E_i^* \stackrel{?}{=} E_i$. If it holds, the SPS chooses a nonce r_2. Then SPS gets $R_i = r_1 \oplus r_2$, $CK = r_2 \cdot D_1^*$ and $K_i = h(ID_{SPS} \parallel CK \parallel A_i \parallel T_2)$.

(4) SPS$\rightarrow U_i$: $\{K_i, R_i, T_2\}$.

(5) The user U_i first checks the timestamp, then U_i get $r_2^* = R_i \oplus r_1$ and computes $CK^* = r_2^* \cdot D_1$, $K_i^* = h(ID_{SPS} \parallel CK^* \parallel A_i \parallel T_2)$. After that, U_i checks whether $K_i^* = K_i$. Then, the session key of the client U_i and the SPS is $CK = CK^* = r_1 \cdot r_2^* \cdot P$.

5.2. Cryptanalysis of Rajamanickam et al.'s scheme

Rajamanickam et al. [38] claim that their scheme can resist insider attacks. However, after our analysis, we reveal that the scheme proposed by Rajamanickam et al. [38] cannot resist insider attacks. And it can neither provide forward secrecy nor user anonymity.

5.2.1. Insider attack

According to the capabilities of the attacker in Section 2, suppose $\mathcal{A}$ is a malicious SPS administrator, she can obtain the message $\{A_i, TID_i\}$ stored in SPS. And $\mathcal{A}$ can also intercept $\{C_i, n\}$ and perform offline password guess. We show this attack as follows:

Step 1. $\mathcal{A}$ chooses a pair (ID_i^*, PW_i^*) from $D_{id} \times D_{puw}$.

Step 2. $\mathcal{A}$ computes $A_i^* = h((h(ID_i^*) \oplus h(PW_i^*)) \bmod n)$.

Step 3. $\mathcal{A}$ verifies the correctness of (ID_i^*, PW_i^*) by checking whether $A_i^* = A_i$.

Step 4. $\mathcal{A}$ repeats the steps $1 \sim 3$ until the correct values are found.

In this scheme, they attempt to provide off-line password guess attack resistance by employing "fuzzy-verifier" technique [13]. However, we find they use the wrong "fuzzy verifier". It is invalid and even makes the situation worse. Specifically, in the above attack, $\mathcal{A}$ can find $(|D_{id}| \times |D_{pw}|)/n$ pair of (ID_i^*, PW_i^*) satisfied equation $A_i = h((h(ID_i^*) \oplus h(PW_i^*)) \bmod n)$. As it is known that, if the $\mathcal{A}$ desire to pass the authentication from the server, she has to know the legitimate user's ID and PW. Unfortunately, A_i is the key parameter to help the server authenticate user, which means $\mathcal{A}$ can use $(|D_{id}| \times |D_{pw}|)/n$ pair of (ID_i^*, PW_i^*) to impersonate user pass the authentication of the server. In a word, their method does not only increase the difficulty of attack, but also make the attack easier. The time complexity of the attacking procedure is $\mathcal{O}((|D_{id}| \times |D_{pw}|)/n) \times (3T_h + T_{xor} + T_{mod}))$.

In their scheme, there is no verifier in the log-in phase. That means their scheme cannot provide a mechanism to check the validity of passwords in real time and detect the malicious attackers. We propose a possible countermeasure that using Wang et al. [13]'s "fuzzy-verifier" technique to solve this flaw. Firstly, the web browser computes $V_i = h(h(ID_i) \oplus h(PW_i \parallel a) \bmod n)$. Then the browser stores it in a table. After that, if the user desires to log-in, she needs to entire her ID_i^*, PW_i^*. Then the web browser computes V_i^* and checks whether $V_i^* \stackrel{?}{=} V_i$. Last but not least, the web browser provides timely detection mechanism for mistyped pair of (ID_i, PW_i), which will resist off-line password guessing attacks.

5.2.2. No forward secrecy

Unfortunately, Rajamanickam et al.'s [38] scheme cannot provide forward security. In case the SPS's master secret key MSK_j has been compromised by an adversary $\mathcal{A}$. $\mathcal{A}$ can intercept the messages $\{E_i, F_i, G_i\}$ and $\{K_i, R_i, T_2\}$. With the extracted MSK_j and message, $\mathcal{A}$ can obtain the session key. We show the steps as follows:

Step 1. $\mathcal{A}$ extracts TID_i^* from F_i using the master secret key MSK_j.

Step 2. $\mathcal{A}$ computes $r_1 = G_i \oplus TID_i^*$.

Step 3. $\mathcal{A}$ computes $r_2 = R_i \oplus r_1$.

Step 4. $\mathcal{A}$ computes $CK = r_1 \cdot r_2 \cdot P$.

The time complexity of the attack is $\mathcal{O}(T_{ecc}+2T_{xor}+2T_{mul})$. The result means that the above attack is quite efficient. Generally speaking, to achieve forward secrecy, the SK usually consists of two fresh random numbers which cannot be obtained in the open channel. However, in the above attack, $\mathcal{A}$ can obtain two random numbers r_1 and r_2. There is not simple solution to resist this attack, which means the designers need to redesign the scheme fundamentally.

5.2.3. No user anonymity

In this scheme, Rajamanickam et al. [38] claim that the proposed U_i's ID is never disclosed to the SPS. However, with our analysis, this protocol cannot achieve this goal. The user's identity can be extracted by the following steps. Suppose $\mathcal{A}$ is a server administrator.

Step 1. $\mathcal{A}$ chooses a ID_i^* from $\mathcal{D}_{id}$, where $\mathcal{D}_{id}$ represents the identity space.

Step 2. $\mathcal{A}$ extracts TID_i from server.

Step 3. $\mathcal{A}$ computes $TID_i^* = h(ID_i^*)$.

Step 4. $\mathcal{A}$ check if $TID_i^* = TID_i$. If not, $\mathcal{A}$ repeat step 1–3 until find correct ID_i.

The time complexity of above attack is $\mathcal{O}(|D_{id}| \times T_h)$. The above attack can be effectively implemented in polynomial time, thus, $\mathcal{A}$ can exploit the user's identity, which means that this scheme cannot realize their claimed goal.

6. Some lessons learned

It has proved that despite decades of development, designing a secure password-based authentication scheme is still hard work. Most protocol designers only think about the modification and enhancement of a single protocol, but few people consider the general design principles of a certain type of protocol. In 2014, Ma et al. [30] proposed three vital design principles for secure two-factor schemes based on smart cards, which has made a profound impact on the development of designing a secure scheme.

(1) The public key principle

In 1999, Halevi-Krawczyk [56] proposed a principle that password-based single-factor authentication scheme cannot resist off-line guess attack by only using symmetric cryptographic primitives. Based on this principle, Ma et al. [30] proposed and proved that two-factor authentication scheme based on smart card cannot resist off-line password guessing attacks if they do not employ public-key techniques. Moreover, when we consider n-factor security, we usually suppose that adversary can get $n-1$ factors which means n-factor security schemes will become a traditional two-factor security scheme. Therefore, Ma et al.'s principle [30] is also suitable for multi-factor authentication.

(2) The security-usability trade-off principle

The smart card storing a verifier to achieve "local password update" will lead to off-line password guessing attacks. That means when the designers have to make trade-offs when they choose to provide the function of "local password update" or achieve the property of "resistance against off-line password guess attack" in their schemes.

(3) The forward secrecy principle

When we analyze the forward secrecy, we usually suppose the adversary can eavesdrop server's long-term secret key. Ma et al. [30] proposed that forward secrecy cannot be achieved without the public-key technique in the schemes and the server should conduct at least two exponential operations or point multiplication.

Though these three principles are proposed for two-factor schemes, it is also suitable for password-based multi-factor authentication schemes. Furthermore, they are also used in various environments such as cloud services, industrial IoT, WSN. Besides, there are still other principles of protocol design (e.g., [13,54,67]). Accordingly, we find that the reason why the three schemes [37–39] are vulnerable to various attacks is that they do not consider the protocol design principles. We have analyzed that the three schemes mentioned above are vulnerable to various attacks and cannot achieve security goals. To learn some lessons, we summarize the flaws of the three schemes and discuss the principles they violated in Table 5. For instance, the smart card attack in Section 4.2.1 is mainly due to violating Ma et al.'s "security-usability" trade-off principle [30] and it also does not use the "fuzzy-verifier" technique [13] to solve this situation. What is more, the vulnerabilities reported in Sections 3.2.3, 4.2.2 and 5.2.2 are mainly due to the violation of Ma et al.'s "forward secrecy principle" [30]. It also reflects the importance of meeting the requirements of the three basic principles for designing a secure protocol. Therefore, we also reveal a series of two-factor or multi-factor schemes proposed recently [37–47] and show in Table 6 that most of them are mostly typical invalid protocols which do not respect the three principles mentioned above and cannot achieve indeed security.

7. Conclusion

In this work, we analyze three different authentication schemes for cloud environments presented by Karuppiah et al. [39], Lin [37] and Rajamanickam et al. [38], which are designed for cloud environments. We demonstrate that they all suffer from security attacks (e.g., smart card loss attack, insider attack) and fail to achieve forward secrecy. In order to draw some lessons, we summarize the causes of these flaws and propose possible countermeasures (e.g., "fuzzy-verifier", update parameter after login) to overcome these pitfalls. Meanwhile, we reveal

Table 5
Summary of our cryptanalysis results and the underlying vulnerabilities.

Scheme	Weakness	Vulnerabilities exploited
Karuppiah et al. [39]	Smart card loss attack (See Section 3.2.1)	The user does not correctly use public keycryptography to construct the log-in message [30].
	Insider attack (See Section 3.2.2)	When the attacker obtains user's smart card, the security of user's password is affected [65,66].
	No forward secrecy (See Section 3.2.3)	At least two exponentiations are needed at the server side to achieve forward security [30].
Lin [37]	Smart card loss attack (See Section 4.2.1)	The smart card stores an explicit password verificationverifier which violates the "security- usability trade-offprinciple" in [9,30].
	No forward secrecy (See Section 4.2.2)	At least two exponentiations are needed at the serverside to achieve forward security [30].
Rajamanickam et al. [38]	Smart card loss attack (See Section 5.2.1)	The authentication server should not store sensitiveparameters of the user to avoid insider attacks [30].
	No forward secrecy (See Section 5.2.2)	At least two exponentiations are needed at the serverside to achieve forward security [30].

Table 6
A summary of the violation of three protocol design principles in recent schemes.

Principle	Kaul et al. [40]	Karuppiah et al. [39]	Lin [37]	Rajamanickam et al. [38]	Roy et al. [41]	Guo et al. [42]	Ostad-Sharif et al. [43]	Kaura et al. [44]	Gope et al. [45]	Chaudhry et al. [46]	Ayub et al. [47]
Public key principle [30]	✓	✓	✓	✓	×	×	×	✓	×	✓	✓
Security-usability trade-off principle [30]	✓	×	✓	✓	×	×	×	✓	×	×	×
Forward secrecy principle [30]	×	×	×	×	×	×	×	×	×	✓	✓

Note that "✓" means achieving the corresponding principle, while "×" not.

that it is necessary for protocol designers to follow the three principles of protocol design (i.e. the public key principle, the security-usability trade-off principle and the forward secrecy principle) proposed by Ma et al. [30]. We also analyzed whether a dozen recently proposed schemes follow the above three principles. After our analysis, there are still many protocol designers who do not pay attention to the importance of the principle of designing protocol, which leads to attacks on the protocol. How to design a truly two-factor secure authentication scheme under the protocol design principles is still a huge challenge. In the future work, we will pay more attention to this.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the Program of 100 Young Academic Leaders of Nankai University.

References

[1] R. Gilad-Bachrach, K. Laine, K.E. Lauter, P. Rindal, M. Rosulek, Secure data exchange: A marketplace in the cloud, in: Proc. ACM CCS 2019, pp. 117–128.
[2] T. Wang, Y. Lu, J. Wang, H. Dai, X. Zheng, W. Jia, EIHDP: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems, IEEE Trans. Comput. (2021) http://dx.doi.org/10.1109/TC.2021.3060484.
[3] T. Wang, Y. Mei, X. Liu, J. Wang, Z. Wang, Edge-based auditing method for data security in resource-constrained Internet of Things, J. Syst. Archit. (2020) http://dx.doi.org/10.1016/j.sysarc.2020.101971.
[4] X. Zhang, J. Zhao, C. Xu, H. Wang, Y. Zhang, DOPIV: Post-quantum secure identity-based data outsourcing with public integrity verification in cloud storage, IEEE Trans. Serv. Comput. (2019) 1, http://dx.doi.org/10.1109/TSC.2019.2942297.
[5] Q. Wang, C. Cheng, R. Xu, J. Ding, Z. Liu, Analysis and enhancement of a lattice-based data outsourcing scheme with public integrity verification, IEEE Trans. Serv. Comput. (2020) 1, http://dx.doi.org/10.1109/TSC.2020.3041324.
[6] D. He, Y. Zhang, D. Wang, K.K.R. Choo, Secure and efficient two-party signing protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography, IEEE Trans. Depend. Secur. Comput. 17 (5) (2020) 1124–1132, http://dx.doi.org/10.1109/TDSC.2018.2857775.
[7] Q. Feng, D. He, H. Wang, D. Wang, X. Huang, Multi-party key generation protocol for the identity-based signature scheme in the IEEE P1363 standard for public key cryptography, IET Inf. Secur. 14 (2020) 724–732.
[8] J. Srinivas, A.K. Das, N. Kumar, J.J.P.C. Rodrigues, Cloud centric authentication for wearable healthcare monitoring system, IEEE Trans. Depend. Secur. Comput. 17 (5) (2020) 942–956.
[9] D. Wang, D. He, P. Wang, C.-H. Chu, Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment, IEEE Trans. Depend. Secur. Comput. 12 (4) (2015) 428–442.
[10] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.
[11] V. Odelu, A.K. Das, S. Kumari, X. Huang, M. Wazid, Provably secure authenticated key agreement scheme for distributed mobile cloud computing services, Future Gener. Comput. Syst. 68 (2017) 74–88.
[12] Q. Jiang, M.K. Khan, X. Lu, J. Ma, D. He, A privacy preserving three-factor authentication protocol for e-health clouds, J. Supercomput. 72 (10) (2016) 3826–3849.
[13] D. Wang, P. Wang, Two birds with one stone: Two-factor authentication with security beyond conventional bound, IEEE Trans. Depend. Secur. Comput. 15 (4) (2018) 708–772.
[14] G. Yang, D.S. Wong, H. Wang, X. Deng, Two-factor mutual authentication based on smart cards and passwords, J. Comput. System Sci. 74 (7) (2008) 1160–1172.
[15] L. Lamport, Password authentication with insecure communication, Commun. ACM 24 (11) (1981) 770–772.
[16] A. Shimizu, A dynamic password authentication method using a one-way function, Syst. Comput. Japan 22 (7) (1991) 32–40.
[17] E. Bresson, O. Chevassut, D. Pointcheval, New security results on encrypted key exchange, in: Proc. PKC 2004, Vol. 2947, Springer, 2004, pp. 145–158.
[18] J. Becerra, D. Ostrev, M. Skrobot, Forward secrecy of SPAKE2, IACR Cryptol. EPrint Arch. 2019 (2019) 351.
[19] S. Shieh, W. Yang, H. Sun, An authentication protocol without trusted third party, IEEE Commun. Lett. 1 (3) (1997) 87–89.
[20] Z. Li, D. Wang, E. Morais, Quantum-safe round-optimal password authentication for mobile devices, IEEE Trans. Depend. Secur. Comput. (2020) http://dx.doi.org/10.1109/TDSC.2020.3040776.
[21] C. Wang, D. Wang, G. Xu, D. He, Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0, Sci. China Inf. Sci. (2020) http://dx.doi.org/10.1007/s11432-020-2975-6.

[22] C.C. Chang, T.C. Wu, Remote password authentication with smart cards, IEE Proc.-Comput. Digit. Tech. 138 (3) (1991) 165–168.
[23] W. Yang, S. Shieh, Password authentication schemes with smart cards, Comput. Secur. 18 (8) (1999) 727–733.
[24] K. Nohl, D. Evans, Starbug, H. Plötz, Reverse-engineering a cryptographic RFID tag, in: P.C. van Oorschot (Ed.), Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA, USENIX Association, 2008, pp. 185–194.
[25] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552, http://dx.doi.org/10.1109/TC.2002.1004593.
[26] J. Xu, W. Zhu, D. Feng, An improved smart card based password authentication scheme with provable security, Comput. Stand. Interfaces 31 (4) (2009) 723–728.
[27] S.K. Sood, A.K. Sarje, K. Singh, An improvement of Xu et al.'s authentication scheme using smart cards, in: Proceedings of the Third Annual ACM Bangalore Conference, in: COMPUTE'10, ACM, New York, NY, USA, 2010, pp. 1–5.
[28] R. Song, Advanced smart card based password authentication protocol, Comput. Stand. Interfaces 32 (5) (2010) 321–325.
[29] B.-L. Chen, W.-C. Kuo, L.-C. Wuu, Robust smart-card-based remote user password authentication scheme, Int. J. Commun. Syst. S 27 (2) (2014) 377–389.
[30] C. Ma, D. Wang, S. Zhao, Security flaws in two improved remote user authentication schemes using smart cards, Int. J. Commun. Syst. 27 (10) (2014) 2215–2227.
[31] P. Wang, Z. Zhang, D. Wang, Revisiting anonymous two-factor authentication schemes for multi-server environment, in: Proc. ICICS 2018, pp. 805–816.
[32] J. Yang, Y. Chang, C. Huang, A user authentication scheme on multi-server environments for cloud computing, in: ICICS, IEEE, 2013, pp. 1–4.
[33] Y. Park, K. Park, Y. Park, Secure user authentication scheme with novel server mutual verification for multiserver environments, Int. J. Commun. Syst. 32 (7) (2019).
[34] Q. Feng, D. He, S. Zeadally, H. Wang, Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment, Future Gener. Comput. Syst. 84 (2018) 239–251.
[35] A. Irshad, S.A. Chaudhry, M. Sher, B.A. Alzahrani, S. Kumari, X. Li, F. Wu, An anonymous and efficient multiserver authenticated key agreement with offline registration centre, IEEE Syst. J. 13 (1) (2018) 436–446.
[36] S. Zhou, Q. Gan, X. Wang, Authentication scheme based on smart card in multi-server environment, Wirel. Netw. 26 (2) (2020) 855–863.
[37] H. Lin, Traceable anonymous authentication and key exchange protocol for privacy-aware cloud environments, IEEE Syst. J. 13 (2) (2019) 1608–1617.
[38] S. Rajamanickam, S. Vollala, R. Amin, N. Ramasubramanian, Insider attack protection: Light- weight password-based authentication techniques using ECC, IEEE Syst. J. 14 (2) (2020) 1972–1983.
[39] M. Karuppiah, A.K. Das, X. Li, S. Kumari, F. Wu, S.A. Chaudhry, N. Radhakrishnan, Secure remote user mutual authentication scheme with key agreement for cloud environment, Mob. Netw. Appl. 24 (3) (2019) 1046–1062.
[40] S.D. Kaul, A.K. Awasthi, Security enhancement of an improved remote user authentication scheme with key agreement, Wirel. Pers. Commun. 89 (2) (2016) 621–637.
[41] S. Roy, A.K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, J.J. Rodrigues, Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications, IEEE Trans. Ind. Inform. 15 (1) (2019) 457–468.
[42] H. Guo, C. Chen, Y. Gao, X. Li, J. Jin, A secure three-factor multiserver authentication protocol against the honest-but-curious servers, Wirel. Commun. Mob. Comput. 2018 (2018) 1–14.
[43] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, D. Abbasinezhad-Mood, Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme, Future Gener. Comput. Syst. 100 (2019) 882–892.
[44] D. Kaura, D. Kumar, Cryptanalysis and improvement of a two-factor user authentication scheme for smart home, J. Inf. Secur. Appl. 58 (2021) 102787.
[45] P. Gope, Enhanced secure mutual authentication and key agreement scheme with user anonymity in ubiquitous global mobility networks, J. Inf. Secur. Appl. 35 (2017) 160–167.
[46] S.A. Chaudhry, T. Shon, F. Al-Turjman, M.H. Alsharif, Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems, Comput. Commun. 153 (2020) 527–537.
[47] M.F. Ayub, S. Shamshad, K. Mahmood, S.H. Islam, R.M. Parizi, K.R. Choo, A provably secure two-factor authentication scheme for USB storage devices, IEEE Trans. Consumer Electron. 66 (4) (2020) 396–405.
[48] J. Zhang, N. Lu, J. Ma, C. Yang, Universally composable secure geographic area verification without pre-shared secret, Sci. China Inf. Sci. 62 (3) (2019) 32113:1–32113:15.
[49] H. Xiong, Z. Kang, J. Chen, J. Tao, S. Kumari, A novel multiserver authentication scheme using proxy resignature with scalability and strong user anonymity, IEEE Syst. J. (2020) http://dx.doi.org/10.1109/JSYST.2020.2983198.
[50] P. Gope, A.K. Das, N. Kumar, Y. Cheng, Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks, IEEE Trans. Ind. Inform. 15 (9) (2019) 4957–4968.
[51] X. Li, J. Peng, M.S. Obaidat, F. Wu, M.K. Khan, C. Chen, A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems, IEEE Syst. J. 14 (1) (2020) 39–50.
[52] D. Wang, Z. Zhang, P. Wang, Targeted online password guessing: An underestimated threat, in: Proc. ACM CCS 2016, pp. 1242–1254.
[53] D. Wang, P. Wang, On the implications of Zipf's law in passwords, in: Proc. ESORICS 2016, pp. 111–131.
[54] D. Wang, P. Wang, On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions, Comput. Netw. 73 (C) (2014) 41–57.
[55] M. Bellare, D. Pointcheval, P. Rogaway, Authenticated key exchange secure against dictionary attacks, in: Proc. EUROCRYPT 2000, pp. 139–155.
[56] S. Halevi, H. Krawczyk, Public-key cryptography and password protocols, ACM Trans. Inf. Syst. Secur. 2 (3) (1999) 230–268.
[57] D. Wang, X. Zhang, Z. Zhang, P. Wang, Understanding security failures of multi-factor authentication schemes for multi-server environments, Comput. Secur. 88 (2020) 101619.
[58] M. Xu, Q. Dong, M. Zhou, C. Wang, Y. Liu, Security analysis on "anonymous authentication scheme for smart home environment with provable security", Wirel. Commun. Mob. Comput. 2020 (2020) 8838363:1–8838363:4.
[59] T. Kasper, D. Oswald, C. Paar, Side-channel analysis of cryptographic RFIDs with analog demodulation, in: Proc. RFIDSec 2012, pp. 61–77.
[60] A. Bogdanov, I. Kizhvatov, Beyond the limits of DPA: Combined side-channel collision attacks, IEEE Trans. Comput. 61 (8) (2012) 1153–1164.
[61] M. Fotouhi, M. Bayat, A.K. Das, H.A.N. Far, S.M. Pournaghi, M. Doostari, A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT, Comput. Netw. 177 (2020) 107333.
[62] D. Wang, H. Cheng, P. Wang, X. Huang, G. Jian, Zipf's law in passwords, IEEE Trans. Inform. Foren. Secur. 12 (11) (2017) 2776–2791.
[63] X. Huang, X. Chen, J. Li, L. Xiang, Further observations on smart-card-based password-authenticated key agreement in distributed systems, IEEE Trans. Parallel Distrib. Syst. 25 (7) (2014) 1767–1775.
[64] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory 22 (6) (1976) 644–654.
[65] D. Wang, Q. Gu, H. Cheng, P. Wang, The request for better measurement: A comparative evaluation of two-factor authentication schemes, in: Proc. ACM ASIACCS 2016, pp. 475–486.
[66] R. Madhusudhan, R. Mittal, Dynamic ID-based remote user password authentication schemes using smart cards: A review, J. Netw. Comput. Appl. 35 (4) (2012) 1235–1248.
[67] D. Wang, N. Wang, P. Wang, S. Qing, Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity, Inform. Sci. 321 (2015) 162–178.

Meijia Xu received the B.S. degree in information security from the China University of Mining and Technology, P. R. China, in Jun. 2020. She is currently working toward the M.S. degree from the College of Cyber Science, Nankai University, Tianjin, P. R. China. Her research interests include applied cryptography and password-based authentication.

Ding Wang received his Ph.D. degree in Information Security at Peking University in 2017. He is currently a Professor at College of Cyber Science, Nankai University, and also serves as the deputy director of the Tianjin key laboratory of Network and Data Security Technology. As the first author or corresponding author, he has published more than 60 papers at venues like ACM CCS, Usenix Security, NDSS, IEEE DSN, ESORICS, ACM TCPS, IEEE TDSC and IEEE TIFS. Six 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, AEC member or PC Chair for over 50 international conferences such as Usenix Security, PETS, ACSAC, ASIACCS, SEC, ISC and ACISP. His research interests include password, multi-factor authentication and provable security.

Qingxuan Wang received the M.S. degree in information security from the China University of Geosciences, Wuhan, P. R. China, in Jun. 2020. He is currently working toward the Ph.D. degree from the College of Cyberspace Security, Nankai University, Tianjin, P.R. China. His research interests include applied cryptography and password-based authentication.

Qiaowen Jia is currently a Ph.D. candidate in Institute of Software, University of Chinese Academy of Sciences. Her research interest includes concurrent program and software verification.