Hindawi
Security and Communication Networks
Volume 2017, Article ID 1619741, 14 pages
https://doi.org/10.1155/2017/1619741

WILEY | Hindawi

Research Article

Cryptanalysis of Three Password-Based Remote User Authentication Schemes with Non-Tamper-Resistant Smart Card

Chenyu Wang and Guoai Xu

School of CyberSpace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Chenyu Wang; 2579005740@qq.com

Received 8 February 2017; Accepted 29 March 2017; Published 17 July 2017

Academic Editor: Alessandro Barenghi

Copyright © 2017 Chenyu Wang and Guoai Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Remote user authentication is the first step to guarantee the security of online services. Online services grow rapidly and numerous remote user authentication schemes were proposed with high capability and efficiency. Recently, there are three new improved remote user authentication schemes which claim to be resistant to various attacks. Unfortunately, according to our analysis, these schemes all fail to achieve some critical security goals. This paper demonstrates that they all suffer from offline dictionary attack or fail to achieve forward secrecy and user anonymity. It is worth mentioning that we divide offline dictionary attacks into two categories: (1) the ones using the verification from smart cards and (2) the ones using the verification from the open channel. The second is more complicated and intractable than the first type. Such distinction benefits the exploration of better design principles. We also discuss some practical solutions to the two kinds of attacks, respectively. Furthermore, we proposed a reference model to deal with the first kind of attack and proved its effectiveness by taking one of our cryptanalysis schemes as an example.

1. Introduction

These days an increasing number of online services (E-Health, E-Banking, and E-Shopping) have been provided for people's daily life with the rapid development of the Internet. Moreover, modern terminal equipment, like smartphones, smartwatches, and Google's Project Glass glasses, has become widespread. The growth of online services and terminal equipment makes the authentication process more important and difficult. Remote authentication is an essential part to guarantee both the claimed user and server are legitimate. In other words, authentication ensures that only the legitimate users can access the resources on the target server. And authentication protocols have been widely used for various fields, including cloud computing, E-Health, and wireless sensor [1–4].

In 1981, Lamport [5] designed the first authentication scheme based on password, while this scheme was pointed out as being insecure shortly: (1) the server having to maintain a password table and (2) high hash overhead. Therefore, many advanced schemes [6–8] were proposed with a lower overhead for the hash function to improve the computing performance of Lamport's scheme, while most of them still require a verification table.

To tackle this problem, Hwang et al. [9] developed a noninteractive password authentication scheme which discards the verification table but using smart card instead in 1990. The main drawback lies in the hardship of changing password. Because the password is related to the ID, for the sake of security, the ID has to be changed once the password is changed. However, it is not easy to change the ID. In 1991, Chang and Wu [10] also developed a scheme using smart card for storing sensitive information to help the authentication. Since then, smart cards have been applied to user authentication schemes widely, and some notable ones include [11–14]. Furthermore, these years many schemes used biometrics characteristic as an additional factor to provide the authentication [15–17].

From 1990 to 2004, numerous remote user authentication schemes with smart card were designed, while almost all were proved to be flawed. However after these years of research, remote user authentication has made great progress: on the one hand, the problem of maintaining the verification table was almost settled, and smart cards got widely used; on

the other hand, the authentication schemes became more sophisticated to withstand the increasing new attacks or to meet more requirements (setting and changing the password freely, no verification table, etc.). Furthermore, ID protection was regarded as an important attribute to be noticed by researchers around 2000. In fact, in this period, most of the proposed schemes used a static user identity in the open communication channel, thus resulting in the ID theft problem. To deal with this problem, in 2004, Das et al. [18] designed a dynamic ID-based scheme, which became a landmark in the history of remote user authentication. The dynamic ID technique is able to conceal the real ID by using random numbers to generate a pseudo identity. As a good and new method to deal with ID theft, Das's scheme draws much attention. However, from then on, many authors raised concerns [19, 20] about Das's scheme and devised a variety of improved schemes. In 2005, Chien and Chen [21] criticized Das's scheme of its incapability of preserving user anonymity and proposed an enhanced one. In 2009, Wang et al. [22] also revealed that Das's scheme was completely insecure for its incapability of password-dependent goal, mutual authentication, and resistance of impersonation attack. In this period, most of the schemes (before or around 2004) assumed the smart cards are tamper resistant; that is, the parameters in the smart cards are unaccessible to adversaries.

Later, however, researchers demonstrated that the message stored in smart card can be easily extracted by reverse engineering techniques [23, 24] and power analysis [25, 26], which becomes another important landmark in remote user authentication area. Since then, most of schemes prefer to use non-tamper-resistant smart cards.

In 2010, Li et al. [27] proposed a password-authenticated key agreement scheme, while Tasi et al. [28] demonstrated it cannot be resistant to desynchronization attack and thus developed a new one. Unfortunately, in 2015 Wang et al. [29] showed Tsai's scheme suffers from smart card loss attack. Song [30] in 2010 revealed that the scheme [31] of Xu et al.'s is vulnerable to impersonation attack and thus designed a new one using symmetric key cryptosystem. Sandeep et al. [32], in the same year, also proved that Xu et al.'s scheme is not resistant to impersonation attack and offline dictionary attack and then devised a new enhanced one. Shortly after, however, Chen et al. [33] found that both the schemes of Song and Stood et al. are not secure: the scheme of Song cannot be resistant to smart card loss attack and offline dictionary attack; the scheme of Stood et al. fails to achieve mutual authentication. So Chen et al. designed an enhanced remote user authentication scheme. While this scheme was also proved by Kumari and Khan [34] it suffered from insider attack and impersonation attack. Li et al. [35], in 2013, reanalyzed Chen et al.'s scheme and then indicated that it cannot promise forward secrecy.

Till recent years, remote user authentication schemes display several distinctive features:

(1) Some attacks, including parallel session attacks, have stolen verifier attacks, and replay attacks are rarely mentioned, which means most schemes can resist these attacks.

Table 1: Security requirements.

1	Denial of Service (DoS) attack
2	Forgery attack (impersonation attack)
3	Replay attack
4	Stolen verifier attack
5	Parallel session attack
6	Password guessing attack
7	Smart card loss attack
8	Reflection attack
9	Insider attack

(2) Smart card loss attack and offline dictionary attack draw more and more attentions:

(i) Ma et al. [36] showed that the public key algorithm is required to resist offline dictionary attack (also called offline-password guessing attack). It is worth mentioning that we will show the following in later section: here the method is specifically applied to the offline dictionary attack using the verification from the open channel, while it is not applied to the offline dictionary attack using the verification from the smart card;

(ii) Wang et al. [29] demonstrated that there is an unavoidable trade-off between changing password locally and resisting smart card loss attack (including offline-password attack). As shown in [37], here the offline dictionary attack should be specific to the offline dictionary attack using the verification from the smart card, but not to offline dictionary attack using the verification from the open channel;

(iii) in [38], Wang gave an analysis to offline dictionary attack and proposed several security models.

(3) User anonymity and forward secrecy attract many discussions: Ma et al. [36] proved that public key algorithm is necessary to protect user anonymity; to achieve forward secrecy, the server side needs to conduct two exponentiation operations at least [36].

Although numerous user remote schemes were proposed, people are still confused about how to assess which scheme is better or whether a scheme is secure enough. Thus Madhusudhan and Mittal [39] tried to answer the question by giving nine security requirements and ten desirable attributes of a sound smart card-based authentication scheme, which we think is another landmark in the history of remote user authentication. Those security requirements and desirable attributes are shown in Tables 1 and 2. They have become an important criterion of an ideal remote authentication scheme. Most of remote user authentication schemes [4, 40–42] are designed and evaluated according to them, while none of the schemes could actually satisfy them simultaneously. Therefore, many researchers begin to pay more attention to

Table 2: Desirable attributes.

1	No password reveal
2	Password dependent
3	No verification table
4	Freely chosen password by the users
5	Forward secrecy
6	User anonymity
7	Mutual authentication
8	Efficiency for wrong password login
9	Smart-card revocation
10	Session key agreement

exploring the design principles and assessment criteria of authentication schemes. The most recent one is from Wang et al. [29, 37]. These two papers explored the relationship between the security requirements and desirable attributes and gave two significant tables to show the relationships. However, how to assess an authentication scheme is still an unsettled issue. Furthermore, in [11], D. Wang and P. Wang for the first time integrated "honeywords" and "fuzzy-verifiers" to settle a long-standing security-usability conflict (i.e., the trade-off between changing password locally and resisting smart card loss attack). It is a remarkable breakthrough in this area, and we will give more details in later section.

Throughout the history of two-factor authentication, it is easy to find the following: although there have been dozens of works endeavored to construct practical remote user authentication schemes, no one has succeeded in withstanding various attacks or satisfying various desirable attributes. The main reason is the chaos of some essential issue, for example, the sound assessment criterion, the reasonable classification, and definition of attacks in smart card-based scheme. Our work tries to give some inspiration on exploring better proposals.

1.1. Our Contributions. Most recently, Yeh [43] proved Chang et al.'s scheme [20] is vulnerable to replay attack, user impersonation attack, and so on and therefore proposed a new authentication scheme with user untraceability. In 2016, Kang et al. [44] showed that Djellali et al.'s scheme [45] suffers from offline dictionary attack, impersonation attack, and replay attack and then developed an enhanced scheme that achieves user anonymity with a Markov chain; and Kaul et al. [46] also designed an improved authentication scheme based on Kumari et al.'s scheme [34]. These schemes all claim to be resistant to various attacks, such as offline dictionary attack and impersonation attack. Unfortunately, according to our analysis, they fail to withstand those attacks as claimed. We summarize our contributions as follows:

(1) This paper demonstrates that the three schemes all suffer from offline dictionary attack, man-in-the-middle attack, and impersonation attack, as well as failing to preserve user anonymity or forward secrecy.

(2) Furthermore, we for the first time divide offline dictionary attacks into two categories: (1) the ones using the verification from smart cards and (2) the ones using the verification from the open channel. The second is more complicated and intractable than the first type. We show that treating them with no difference arouses confusion and misleads the related research. Such distinction which benefits the exploration of better design principles is requisite and significant.

(3) Remarkably, we explore the solution to such two kinds of attacks and propose a reference model to settle the offline dictionary attack using the verification from the open channel and then use Yeh's scheme to check the effectiveness of our reference model; the result shows that our reference model actually works.

The remainder of this paper is organized as follows: in Section 2, the system architecture and the capacities of adversary are explained. In Section 3, we give a cryptanalysis of Yeh's scheme. We review Kang et al.'s scheme in Section 4 and Kaul et al.'s scheme in Section 5. Section 6 analyzes the two kinds of offline-password guessing attacks. And Section 7 gives a conclusion.

2. System Architecture and the Capacities of Adversary

In this section, we first list the notations used in the three schemes and then briefly introduce the system architecture and the capacities of the adversary in the schemes.

2.1. Notations and Abbreviations. The notations in the three schemes are shown in Notations and Abbreviations at the end of the paper.

2.2. System Architecture. Like many other smart card-based authentication methods, the three schemes involve a set of users and a single server. Users access the resources by mutual authentication with server. The authentication usually includes four basic phases: registration, login, authentication, and password change. Firstly, a user submits personal information to the server to register. Then the server issues the user a smart card with security parameters. The registration phase is only performed once unless the user reregisters for special reasons. After that, in the login phase, the user will send the access request. Then the server and the user authenticate each other in verification phase to finish the authentication. The phases of login and verification usually will be carried out many times. A sound two-factor authentication schemes should ensure that only the user who owns the smart card and submits the corresponding password can access the server successfully. As a realistic problem, the password change phase attracts more and more attention these years where the user can change his/her password locally or remotely.

2.3. The Capacities of Adversary. In the cryptanalysis of the two-factor authentication schemes, the adversary $\mathcal{A}$ is also supposed to have the following capacities [29, 47–49]:

(1) $\mathcal{A}$ can fully control the open communication channel; that is, $\mathcal{A}$ can modify, intercept, delete, and resend the eavesdropped messages over an open channel.

User/smart card	Channel	Server S
Choose $(\mathrm{ID}_i, \mathrm{PW}_i)$ Generate random number r_1, r_2	$(\mathrm{ID}_i, \mathrm{PW}_i, r_1, r_2)$ *Smart card with* $(N_i, M_i, r_1, h(\cdot))$	Compute $M_i = h(y \parallel r_2) \oplus h(\mathrm{PW}_i \parallel r_1)$ $N_i = h(\mathrm{ID}_i \parallel x) \oplus h(\mathrm{PW}_i \parallel r_1)$ Store $h(h(y \parallel r_2))$ in table T
Input $(\mathrm{ID}_i', \mathrm{PW}_i')$ *Smart card* Generate random number r_3 Compute $A = M_i \oplus h(\mathrm{PW}_i' \parallel r_1) = h(y \parallel r_2)$ $D = r_3 \oplus h(A)$ $B = N_i \oplus h(\mathrm{PW}_i' \parallel r_1) = h(\mathrm{ID}_i' \parallel x)$ $N_i' = N_i \oplus h(h(A \parallel r_3))$ $\mathrm{CID}_i = \mathrm{ID}_i' \oplus h(N_i \parallel h(A) \parallel r_3)$ $C = h(N_i \parallel h(A) \parallel B \parallel r_3)$	$(D, \mathrm{CID}_i, N_i', C)$	For each $h(h(y \parallel r_2))^*$ in table T Compute $r_3^* = D \oplus h(h(y \parallel r_2))^*$ $N_i^* = N_i' \oplus h(h(h(y \parallel r_2))^* \parallel r_3^*)$ $\mathrm{ID}_i^* = \mathrm{CID}_i \oplus h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel r_3^*)$ $B^* = h(\mathrm{ID}_i^* \parallel x)$ $C^* = h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel B^* \parallel r_3^*)$ Check $C^* =? C$
Check the freshness of r_4 Compute $a' = h(B \parallel h(A) \parallel r_4)$ Check $a' =? a$ If it holds, U_i authenticate S	(a, r_4)	Check the freshness of r_3 Generate a random number r_4 Compute $a = h(B^* \parallel h(h(y \parallel r_2))^* \parallel r_4)$

Figure 1: The scheme of Yeh et al.

(2) $\mathscr{A}$ can enumerate all the items in $\mathscr{D}_{\mathrm{pw}} * \mathscr{D}_{\mathrm{id}}$ in polynomial time, where $\mathscr{D}_{\mathrm{pw}}$ and $\mathscr{D}_{\mathrm{id}}$ denote the password space and the identity space, respectively.

(3) $\mathscr{A}$ can acquire the password of a legitimate user by a malicious card reader or get the parameters in smart card but cannot achieve both.

(4) When evaluating forward secrecy, $\mathscr{A}$ can get the server's secret key.

3. Cryptanalysis of Yeh's Scheme

3.1. Review of Yeh's Scheme. This section gives a brief review of Yeh's [43] scheme with user untraceability (shown in Figure 1).

3.1.1. Registration Phase

Step 1 ($U_i \Rightarrow S$). U_i chooses ID_i, PW_i, and two random numbers r_1, r_2 and then sends $\{\mathrm{ID}_i, \mathrm{PW}_i, r_1, r_2\}$ to S via a secure channel.

Step 2 ($S \Rightarrow U_i$). S computes $M_i = h(y \parallel r_2) \oplus h(\mathrm{PW}_i \parallel r_1)$ and $N_i = h(\mathrm{ID}_i \parallel x) \oplus h(\mathrm{PW}_i \parallel r_1)$ and then sends U_i a smart card with security parameters $\{M_i, N_i, r_1, h(\cdot)\}$ via a secure channel and stores $h(h(y \parallel r_2))$ in a table.

3.1.2. Login Phase and Authentication Phase

Step 1 ($U_i \rightarrow S$). U_i inputs ID_i', PW_i'. The smart card generates a random number r_3, computes $A = M_i \oplus h(\mathrm{PW}_i' \parallel r_1) = h(y \parallel r_2)$, $D = r_3 \oplus h(A)$, $B = N_i \oplus h(\mathrm{PW}_i' \parallel r_1) = h(\mathrm{ID}_i' \parallel x)$, $N_i' = N_i \oplus h(h(A \parallel r_3))$, $\mathrm{CID}_i = \mathrm{ID}_i' \oplus h(N_i \parallel h(A) \parallel r_3)$, and $C = h(N_i \parallel h(A) \parallel B \parallel r_3)$, and then sends $\{D, \mathrm{CID}_i, N_i', C\}$ to S.

Step 2 ($S \rightarrow U_i$). S traverses the $h(h(y \parallel r_2))^*$ in table T, computes $r_3^* = D \oplus h(h(y \parallel r_2))^*$, $N_i^* = N_i' \oplus h(h(h(y \parallel r_2))^* \parallel r_3^*)$, $\mathrm{ID}_i^* = \mathrm{CID}_i \oplus h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel r_3)$, $B^* = h(\mathrm{ID}_i^* \parallel x)$, and $C^* = h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel B^* \parallel r_3^*)$, and then checks $C^* =? C$. If none of the value $h(h(y \parallel r_2))^*$ satisfies the equation, end the session. Otherwise, S examines the freshness of r_3 and whether $\{D, \mathrm{CID}_i, N_i', C\}$ has been received before. If one of the conditions is invalid, terminate the session. Otherwise, S generates a random number r_4, computes $a = h(B^* \parallel h(h(y \parallel r_2))^* \parallel r_4)$, and sends $\{a, r_4\}$ to U_i.

Step 3. The smart card firstly checks r_4, then computes $a' = h(B \parallel h(A) \parallel r_4)$, and checks whether a' equals a. If true, U_i authenticates S.

3.1.3. Password Change Phase. If U_i wants to change the password, he/she inserts the smart card to the card reader and inputs ID_i, PW_i, and a new password $\mathrm{PW}_{i_{\mathrm{new}}}$. Then the smart card computes $M_{i_{\mathrm{new}}} = M_i \oplus h(\mathrm{PW}_i \parallel r_1) \oplus h(\mathrm{PW}_{i_{\mathrm{new}}} \parallel r_1)$ and $N_{i_{\mathrm{new}}} = N_i \oplus h(\mathrm{PW}_i \parallel r_1) \oplus h(\mathrm{PW}_{i_{\mathrm{new}}} \parallel r_1)$ and then replaces M_i and N_i with $M_{i_{\mathrm{new}}}$ and $N_{i_{\mathrm{new}}}$.

3.2. Cryptanalysis of Yeh's Schemes. In this section we show that Yeh's scheme cannot resist various attacks, such as password guessing attack, impersonation attack, and desynchronization attack.

3.2.1. Offline Dictionary Attack via Verification Value in Channel. Supposing the adversary $\mathscr{A}$ stole U_i's smart card and then got security parameters N_i, r_1, and M_i from the smart card, $\mathscr{A}$ also has $\{D_i, \mathrm{CID}_i, N_i', C\}$ through eavesdropping the open

channel between U_i and S; then $\mathscr{A}$ can perform the attack by the following steps:

(1) Guess the value of PW_i to be PW_i^* from the password dictionary space $\mathscr{D}_{\mathrm{pw}}$.

(2) Compute $A^* = M_i \oplus h(\mathrm{PW}_i^* \parallel r_1) = h(y \parallel r_2)^*$; M_i and r_1 are extracted from the smart card.

(3) Compute $r_3^* = D \oplus h(A^*)$; D is eavesdropped from the open channel.

(4) Compute $B^* = N_i \oplus h(\mathrm{PW}_i^* \parallel r_1)$; N_i and r_1 are extracted from the smart card.

(5) Compute $C^* = h(N_i \parallel h(A^*) \parallel B^* \parallel r_3^*)$; N_i is extracted from the smart card.

(6) Verify the correctness of PW_i^* by checking if $C^* =? C$, C is from the open channel.

(7) Repeat Steps (1), (2), (3), (4), (5), and (6) until the correct value of PW_i^* is found.

The time complexity of the above attack is $\mathcal{O}(|\mathscr{D}_{\mathrm{pw}}| * (3T_H + 3T_R))$. T_H is the running time for hash computation. T_R is the running time for exclusive-or operation. $|\mathscr{D}_{\mathrm{pw}}|$ denotes the number of passwords in $\mathscr{D}_{\mathrm{pw}}$, and $|\mathscr{D}_{\mathrm{pw}}|$ is very limited in practice [49, 50]; usually $|\mathscr{D}_{\mathrm{pw}}| \leq 10^6$; so the above attack is quite efficient.

Remark 1. The offline dictionary attack here uses the verification from the open channel. The inherent reason for this attack is that (1) the adversary can find a verification to check whether the guessing value is correct; (2) the password is the only unknown value to the adversary; that is, the adversary can get other parameters consisting of the verification, except the password or the identity. To such attack, the lightweight public key algorithm is the necessary condition, as explained in [36].

3.2.2. User Anonymity. Once the adversary $\mathscr{A}$ gets the password through "offline dictionary attack," he can get the user's ID by the following steps:

(1) Compute $A = M_i \oplus h(\mathrm{PW}_i^* \parallel r_1) = h(y \parallel r_2)$; M_i and r_1 are from the smart card.

(2) Compute $r_3 = D \oplus h(A)$; D is from open channel.

(3) Compute $\mathrm{ID}_i^* = \mathrm{CID}_i \oplus h(N_i \parallel h(A) \parallel r_3)$; N_i is from smart card; CID_i is from open channel.

In computing ID_i, what the server knows more than the adversary is $h(h(y \parallel r_2))$, while, after getting the PW_i, the adversary can get $h(h(y \parallel r_2))$ by $M_i \oplus h(\mathrm{PW}_i^* \parallel r_1)$, so in fact the adversary $\mathscr{A}$ has the same capacity as the server; thus $\mathscr{A}$ can get ID_i according to the way the server does.

3.2.3. User Impersonation Attack. With the PW_i^* and the ID_i^*, the adversary $\mathscr{A}$ can impersonate U_i as follows:

(1) Compute $A = M_i \oplus h(\mathrm{PW}_i^* \parallel r_1) = h(y \parallel r_2)$; M_i and r_1 are from the smart card.

(2) Generate a random number r_3.

(3) Compute $D_a = r_3 \oplus h(A)$.

(4) Compute $B = N_i \oplus h(\mathrm{PW}_i^* \parallel r_1) = h(\mathrm{ID}_i^* \parallel x)$.

(5) Compute $N_a' = N_i \oplus h(h(A \parallel r_3))$; N_i is extracted from the smart card.

(6) Compute $\mathrm{CID}_a = \mathrm{ID}_i^* \oplus h(N_i \parallel h(A) \parallel r_3)$.

(7) Compute $C_a = h(N_i \parallel h(A) \parallel B \parallel r_3)$.

(8) Interrupt $\{D, \mathrm{CID}_i, N_i', C\}$; send $\{D_a, \mathrm{CID}_a, N_a', C_a\}$ to S to impersonate U_i.

As for the server S, it computes $r_3^* = D_a \oplus h(h(y \parallel r_2))^*$, $N_i^* = N_a' \oplus h(h(h(y \parallel r_2))^* \parallel r_3^*)$, $\mathrm{ID}_i' = \mathrm{CID}_a \oplus h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel r_3^*)$, $B^* = h(\mathrm{ID}_i^* \parallel x)$, and $C^* = h(N_i^* \parallel h(h(y \parallel r_2))^* \parallel B^* \parallel r_3^*)$ and then checks (1) $C^* =? C_a$; (2) the freshness of r_3; and (3) whether $\{D_a, \mathrm{CID}_a, N_a', C_a\}$ has ever been received before. All of them are satisfied, so $\mathscr{A}$ is authenticated by S successfully.

With PW_i, ID_i, and smart card, the adversary $\mathscr{A}$ has the same capacity as the legitimate user; that is, $\mathscr{A}$ can impersonate the user to the server successfully. The original reason for this attack is the offline dictionary attack.

3.2.4. Server Impersonation Attack. With the PW_i^*, $\mathscr{A}$ can impersonate S as follows:

(1) Compute $h(y \parallel r_2) = M_i \oplus h(\mathrm{PW}_i^* \parallel r_1)$; M_i and r_1 are extracted the from smart card.

(2) Compute $B^* = N_i \oplus h(\mathrm{PW}_i^* \parallel r_1) = h(\mathrm{ID}_i^* \parallel x)$; N_i is extracted from the smart card.

(3) Generate a random number r_4; compute $a_a = h(B^* \parallel h(h(y \parallel r_2)) \parallel r_4^a)$.

(4) Interrupt $\{a, r_4\}$, and send $\{a_a, r_4^a\}$ to the user U_i to impersonate the server S.

On the user side, U_i computes $a' = h(B \parallel h(A) \parallel r_4^a)$, as $a' = a_a$; $\mathscr{A}$ is authenticated by the user U_i successfully.

In most cases, the capacity of legitimate user and remote server is the same; to be more precise, what the legitimate user knows can transform into what the remote server knows. So if the user impersonation attack can be performed, the server impersonation attack can be performed too.

3.2.5. Man-in-the-Middle Attack. With PW_i^* and ID_i^* that have been got from "password guessing attack" and "user anonymity," respectively, $\mathscr{A}$ can execute a man-in-the-middle attack as follows:

(1) Interrupt $\{D, \mathrm{CID}_i, N_i', C\}$ that U_i sends to S.

(2) Compute $\{D_a, \mathrm{CID}_a, N_a', C_a\}$ as in "user impersonation attack," and send them to S.

(3) Interrupt $\{a, r_4\}$ from S via an open channel.

(4) Compute $\{a_a, r_4^a\}$ as in "server impersonation attack," and send them to U_i.

Through above attack procedures, the adversary $\mathscr{A}$ can execute a man-in-the-middle attack without being noticed by U_i or S.

User/smart card	Channel	Server S
Choose $(\mathrm{ID}_i, \mathrm{PW}_i)$ Generate random number b Compute $\mathrm{RPW}_i = h(\mathrm{PW}_i \parallel b)$	$(\mathrm{ID}_i, \mathrm{RPW}_i)$ →	Generate $P_{i\times i}\ \forall\{i\} \in [1,n]$, $y_i \in [1,n]$ Compute π_{y_i} from $\pi = \pi P$ $\pi'_{y_i} = \mathrm{Shift}_d(\pi_{y_i})$ $\mathrm{TPW}_i = h(\mathrm{RPW}_i \parallel \mathrm{ID}_i)$ $P_i = h(x \parallel \pi'_{y_i})$ $Q_i = h(\pi'_{y_i} \parallel x) \oplus P_i$ $A_i = h(\mathrm{ID}_i \parallel \pi'_{y_i}) \oplus P_i$ $B_i = \mathrm{RPW}_i \oplus Q_i$
Input b into smart card Smart card contains $(\mathrm{TPW}_i, A_i, B_i, \pi'_{y_i}, b, h(\cdot))$	← *Smart card with* $(\mathrm{TPW}_i, A_i, B_i, \pi'_{y_i}, h(\cdot))$	
Input $(\mathrm{ID}'_i, \mathrm{PW}'_i)$ *Smart card* $\mathrm{RPW}'_i = h(\mathrm{PW}'_i \parallel b)$ $\mathrm{TPW}'_i = h(\mathrm{RPW}'_i \parallel \mathrm{ID}'_i)$ TPW'_i =? TPW_i $P'_i = A_i \oplus h(\mathrm{ID}'_i \parallel \pi'_{y_i})$ $Q'_i = B_i \oplus \mathrm{RPW}'_i$ $K'_i = P'_i \oplus Q'_i$ Generate a, choose time stamp T_1 Compute $TI_i = \mathrm{TPW}'_i \oplus K'_i$ $L_i = K'_i \oplus a$ $C_i = h(\mathrm{TPW}'_i \parallel K'_i \parallel T_1)$	$(TI_i, L_i, C_i, T_1, \pi'_{y_i})$ →	Check $(T' - T_1) \le \Delta T$ Compute $K_i = h(\pi'_{y_i} \parallel x)$ $N_i = TI_i \oplus K_i$ $C'_i = h(N_i \parallel K_i \parallel T_1)$ Check C'_i =? C_i Compute $a' = L_i \oplus K_i$ Generate z, time stamp T_1 Compute $\mathrm{SK} = h(a' \parallel z \parallel T_1 \parallel T_2)$ $M_i = h(\mathrm{SK} \parallel K_i)$ $J_i = z \oplus a'$
Check $(T'' - T_2) \le \Delta T$ Compute $z' = J_i \oplus a$ $\mathrm{SK}' = h(a \parallel z' \parallel T_1 \parallel T_2)$ $M'_i = h(\mathrm{SK}' \parallel K'_i)$ Check M'_i =? M_i If it holds, U_i authenticate S Accept SK as session key	← (J_i, M_i, T_2)	

Figure 2: The scheme of Kang et al.

In fact, man-in-the-middle attack usually is a result of "server impersonation attack" and "user impersonation attack," while offline dictionary attack is the original reason of these three attacks.

3.2.6. Desynchronization Attack. As there is no any verification in password change phase, an adversary $\mathcal{A}$ can execute desynchronization attack easily: stealing U_i's smart card and inputting a random ID_a, PW_a, and a new password $\mathrm{PW}_{a_{\mathrm{new}}}$. According to the scheme, M_i and N_i will be replaced by $M_{i_{\mathrm{new}}}$ and $N_{i_{\mathrm{new}}}$, respectively, where $M_{i_{\mathrm{new}}} = M_i \oplus h(\mathrm{PW}_a \parallel r_1) \oplus h(\mathrm{PW}_{a_{\mathrm{new}}} \parallel r_1)$ and $N_{i_{\mathrm{new}}} = N_i \oplus h(\mathrm{PW}_a \parallel r_1) \oplus h(\mathrm{PW}_{a_{\mathrm{new}}} \parallel r_1)$. As a result, even the legitimate user cannot login successfully.

Desynchronization attack often happens in password change phase where the user, without inputting the correct PW_i and ID_i, can change the password successfully. This results in the that legitimate user with correct old password cannot login successfully. So if a user wants to change the password, he should be authenticated firstly, and there are usually two ways: interacting with the remote server like the authentication phase and interacting with the smart card. The second way requires a verification value from the smart card; thus such scheme is vulnerable to offline dictionary attack, but it helps detect wrong password input, which saves user's time. The first one requires costing more time to make the user change the password and detect wrong password input.

3.2.7. Insider Attack. In this scheme, the user U_i submits a pair of PW_i and ID_i to the server S without any transformation or protection; thus the server S can get the PW_i and ID_i and carries out an insider attack to impersonate the user U_i.

Insider attack is quite easy to deal with: do some transformation to the PW_i and ID_i, such as $h(\mathrm{PW}_i \parallel b)$ and $h(\mathrm{ID}_i \parallel b)$ (b is a random number).

4. Cryptanalysis of Kang et al.'s Scheme

4.1. Review of Kang et al.'s Scheme. This section gives a brief review of Kang et al.'s scheme [44] (Figure 2). As little relevance as password change phase, we omit it.

4.2. Registration Phase

Step 1 ($U_i \Rightarrow S$). U_i chooses ID_i, PW_i, and a random number b and computes $\mathrm{RPW}_i = h(\mathrm{PW}_i \parallel b)$ and then sends $\{\mathrm{ID}_i, \mathrm{RPW}_i\}$ to the server S via a secure channel.

Step 2 ($S \Rightarrow U_i$). S generates $P_{i\times i}$ and y_i, where $i \in [1, n]$, $y_i \in [1, n]$, and n is a small number, computes π_{y_i} from $\pi = \pi P$ and $\pi'_{y_i} = \mathrm{Shift}_d(\pi_{y_i})$, computes $\mathrm{TPW}_i = h(\mathrm{RPW}_i \parallel \mathrm{ID}_i)$, $P_i = h(x \parallel \pi'_{y_i})$, $Q_i = h(\pi'_{y_i} \parallel x) \oplus P_i$, $A_i = h(\mathrm{ID}_i \parallel \pi'_{y_i}) \oplus P_i$, and $B_i = \mathrm{RPW}_i \oplus Q_i$, and then issues the user U_i a smart card containing $\{\mathrm{TPW}_i, A_i, B_i, \pi'_{y_i}, h(\cdot)\}$ via a secure channel.

Step 3. U_i inputs b into the smart card.

4.2.1. Login Phase and Authentication Phase

Step 1 ($U_i \rightarrow S$). U_i inserts his smart card and inputs ID'_i, PW'_i. The smart card computes $\mathrm{RPW}'_i = h(\mathrm{PW}'_i \parallel b)$ and $\mathrm{TPW}'_i = h(\mathrm{RPW}'_i \parallel \mathrm{ID}'_i)$ and then checks TPW'_i =? TPW_i. If not satisfied, reject the request. Otherwise, the smart card computes $P'_i = A_i \oplus h(\mathrm{ID}'_i \parallel \pi'_{y_i})$, $Q'_i = B_i \oplus \mathrm{RPW}'_i$, and $K'_i = P'_i \oplus Q'_i$, generates a random number a and time stamp T_1, computes $\mathrm{TI}_i = \mathrm{TPW}'_i \oplus K'_i$, $L_i = K'_i \oplus a$, and $C_i = h(\mathrm{TPW}'_i \parallel K'_i \parallel T_1)$, and finally sends $\{\mathrm{TI}_i, L_i, C_i, T_1, \pi'_{y_i}\}$ to S.

Step 2 ($S \rightarrow U_i$). S firstly checks the freshness of T_1, then calculates $K_i = h(\pi'_{y_i} \parallel x)$, $N_i = \mathrm{TI}_i \oplus K_i$, and $C'_i = h(N_i \parallel K_i \parallel T_1)$, and compares C'_i with C_i. If their values are not the same, reject the request; else generate a random number z and then compute $a' = L_i \oplus K_i$, and the session key $\mathrm{SK} = h(a' \parallel z \parallel T_1 \parallel T_2)$, where T_2 is the time stamp, $M_i = h(\mathrm{SK} \parallel K_i)$, and $J_i = z \oplus a'$. After that S sends $\{J_i, M_i, T_2\}$ to U_i.

Step 3. U_i firstly checks the freshness of T_2 and then computes $z' = J_i \oplus a$, $\mathrm{SK}' = h(a \parallel z' \parallel T_1 \parallel T_2)$, and $M'_i = h(\mathrm{SK}' \parallel K'_i)$ and verifies S through comparing M'_i with M_i. If the values of them are the same, U_i authenticates S, and accepts SK as the session key. Otherwise, end the session.

4.3. Cryptanalysis of Kang et al.'s Schemes

4.3.1. Offline Dictionary Attack via Verification Value in Channel. Supposing the adversary $\mathscr{A}$ got U_i's smart card and then acquired security parameters TPW_i, A_i, b, and B_i from the smart card, $\mathscr{A}$ also has $\{\mathrm{TI}_i, L_i, C_i, T_1, \pi'_{y_i}\}$ through eavesdropping the open channel between U_i and S; then $\mathscr{A}$ can perform the attack by the following steps:

(1) Guess PW_i to be PW^*_i and ID_i to be ID^*_i.

(2) Compute $\mathrm{RPW}^*_i = h(\mathrm{PW}^*_i \parallel b)$, $\mathrm{TPW}^*_i = h(\mathrm{RPW}^*_i \parallel \mathrm{ID}^*_i)$, $P^*_i = A_i \oplus h(\mathrm{ID}^*_i \parallel \pi'_{y_i})$, $Q^*_i = B_i \oplus \mathrm{RPW}^*_i$, and $K^*_i = P^*_i \oplus Q^*_i$; TPW_i, A_i, b, B_i are extracted from the smart card.

(3) Compute $N_i = \mathrm{TI}_i \oplus K^*_i$ and $C^*_i = h(N_i \parallel K^*_i \parallel T_1)$, and $\{\mathrm{TI}_i, L_i, T_1, \pi'_{y_i}\}$ is from the channel.

(4) Verify the correctness of PW^*_i and ID^*_i by checking C^*_i =? C_i; C_i is from the channel.

(5) Repeat Steps (1), (2), (3), and (4) until the correct values of PW^*_i and ID^*_i are found.

The time complexity is $\mathcal{O}(|\mathscr{D}_{\mathrm{pw}}| * 4(T_H + T_R))$, so the above attack is quite efficient. Once $\mathscr{A}$ has the PW_i, he/she also can carry out user impersonation attack, server impersonation attack, and man-in-the-middle attack. And as the methods to those attacks are similar to the methods in Yeh's schemes, it is unnecessary to go into details here.

4.3.2. Offline Dictionary Attack via Verification Value in Smart Card. Supposing an adversary $\mathscr{A}$ got U_i's smart card and then acquired security parameters TPW_i, A_i, b, and B_i from the smart card, then $\mathscr{A}$ can perform the attack as follows:

(1) Guess the value of PW_i to be PW^*_i from the password dictionary space $\mathscr{D}_{\mathrm{pw}}$ and ID_i to be ID^*_i from the identity dictionary space $\mathscr{D}_{\mathrm{id}}$.

(2) Compute $\mathrm{RPW}^*_i = h(\mathrm{PW}^*_i \parallel b)$ and $\mathrm{TPW}^*_i = h(\mathrm{RPW}^*_i \parallel \mathrm{ID}^*_i)$; b is from the smart card.

(3) Verify the correctness of PW^*_i and ID^*_i by checking whether TPW^*_i =? TPW_i; TPW_i is extracted from the smart card.

(4) Repeat Steps (1), (2), and (3) until the PW^*_i and ID^*_i are found.

Remark 2. The time complexity is $\mathcal{O}(|\mathscr{D}_{\mathrm{pw}}| * 2T_H)$, so the above attack is quite efficient. This kind of offline dictionary attack above uses the verification from the smart card, while with the verification the user can change password locally. This is exactly what Wang et al. [29] demonstrated which is the trade-off between changing password locally and resisting offline-password attack. Luckily, in [11], D. Wang and P. Wang for the first time integrated "honeywords" and "fuzzy-verifiers" to settle such a long-standing security-usability conflict. So according to [11], we simply give an improved way to avoid such conflict. Let the verification $\mathrm{TPW}_i = h((h(\mathrm{PW}_i \parallel b) \parallel \mathrm{ID}_i) \bmod n_0)$, where $2^4 \le n_0 \le 2^8$ and n_0 determines the capacity of the pool of the (ID, PW). So now there are $|\mathscr{D}_{\mathrm{pw}}| * \mathscr{D}_{\mathrm{id}} \setminus n_0 \approx 2^{32}$ candidates of (ID, PW) pair for adversary to guess when $n_0 = 2^8$ and $|\mathscr{D}_{\mathrm{pw}}| = \mathscr{D}_{\mathrm{id}} = 2^6$. For these candidates, the adversary can only guess the right one from online guessing, while there is also a way called "honeywords" to avoid such online dictionary guessing; "honeywords" in fact is a word list to timely detect whether the smart card is extracted.

4.3.3. Forward Secrecy. Supposing $\mathscr{A}$ knew S's secret key x, then he can calculate the session key SK as follows:

(1) Interrupt $\{\mathrm{TI}_i, L_i, C_i, T_1, \pi'_{y_i}\}$ that U_i sends to S.

(2) Compute $K_i = h(\pi'_{y_i} \parallel x)$; π'_{y_i} is from smart card.

(3) Compute $a' = L_i \oplus K_i$.

(4) Interrupt $\{J_i, M_i, T_2\}$ that S sends to U_i.

(5) Compute $z' = J_i \oplus a'$.

(6) Compute $\mathrm{SK} = h(a' \parallel z' \parallel T_1 \parallel T_2)$; at this point the user $\mathscr{A}$ gets SK successfully.

User U/smart card	Channel	Server S
Choose ID_i, PW_i, and random number b $\mathrm{RPW}_i = h(b \parallel \mathrm{PW}_i)$	$(\mathrm{ID}_i, \mathrm{RPW}_i)$ →	Choose unique random number y_i Compute $\alpha_i = h((\mathrm{ID}_i \oplus x) \parallel y)$ $\beta_i = \alpha_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i)$ $\gamma_i = y_i \oplus h(\alpha_i \oplus \mathrm{RPW}_i)$ $\chi_i = h(\mathrm{ID}_i \parallel \mathrm{RPW}_i \parallel y_i \parallel \alpha_i)$
Compute $\eta_i = b \oplus h(\mathrm{ID}_i \oplus \mathrm{PW}_i)$ Input η_i into card reader	← *Smart card* $(\beta_i, \gamma_i, \chi_i, h(\cdot))$	
Input $(\mathrm{ID}_i^*, \mathrm{PW}_i^*)$ *Smart card* $b = \eta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{PW}_i^*)$ $\mathrm{RPW}_i^* = h(b \parallel \mathrm{PW}_i^*)$ $\alpha_i^* = \beta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$ $\gamma_i^* = \gamma_i \oplus h(\alpha_i^* \oplus \mathrm{RPW}_i^*)$ $\chi_i^* = h(\mathrm{ID}_i^* \parallel \mathrm{RPW}_i^* \parallel y_i^* \parallel \alpha_i^*)$ Check $\chi_i^* \stackrel{?}{=} \chi_i$; if not, exit Compute $\omega_i = y_i \oplus h(\mathrm{ID}_i \oplus \alpha_i) \oplus h(\mathrm{ID}_i \oplus \alpha_i \oplus T)$ $\vartheta_i = h(\mathrm{ID}_i \parallel \alpha_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T)$	$(\mathrm{ID}_i, \omega_i, \vartheta_i, T)$ →	Verify $T' - T \le \delta T$ Compute $\alpha_i^* = h((\mathrm{ID}_i^* \oplus x) \parallel y)$ $y_i^* = \omega_i^* \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^*) \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^* \oplus T)$ $\vartheta_i^* = h(\mathrm{ID}_i^* \parallel \alpha_i^* \parallel y_i^* \parallel (\alpha_i^* \oplus y_i^*) \parallel T)$ Check $\vartheta_i^* \stackrel{?}{=} \vartheta_i$; if not, exit Compute $\mu_i = h(\mathrm{ID}_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$
Verify $T_s' - T_s \le \delta T_s$ Compute $\mu_i^* = h(\mathrm{ID}_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$ Check $\mu_i^* \stackrel{?}{=} \mu_i$; if not, exit $\mathrm{SK} = h(\mathrm{ID}_i \oplus \alpha_i \oplus y_i \oplus T \oplus T_s)$	← (μ_i, T_s)	

Figure 3: The scheme of Kaul et al.

In this scheme, the session key consists of a random number a from U_i, a random number z from S, and two open time stamps T_1 and T_2. The key parameters are the two random number, while, compared to the adversary, what the server only knows more is the secret key x, so once the adversary knows the secret key x, he can compute the random number a' chosen by U_i as the way the server does. On the other hand, in computing SK, what the user only knows more than the adversary is the random number a. While the adversary has known a now, thus the adversary also can compute the random number z chosen by the server as the user does. With a and z, the adversary gets sk. Furthermore, it proves that "more than two exponentiation operations conducted on the server side are necessary to achieve forward secrecy" [36].

5. Cryptanalysis of Kaul et al.'s Schemes

5.1. Review of Kaul et al.'s Scheme. This section gives a brief review of Kaul et al.'s scheme [46] (Figure 3), and password change phase is also omitted.

5.1.1. Registration Phase

Step 1 ($U_i \Rightarrow S$). U_i chooses ID_i, PW_i, and a random number b, then computes $\mathrm{RPW}_i = h(b \parallel \mathrm{PW}_i)$, and submits $S\{\mathrm{ID}_i, \mathrm{RPW}_i\}$ via a secure channel.

Step 2 ($S \Rightarrow U_i$). S chooses a unique random number y_i for U_i and computes $\alpha_i = h((\mathrm{ID}_i \oplus x) \parallel y)$, $\beta_i = \alpha_i \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i)$, $\gamma_i = y_i \oplus h(\alpha_i \oplus \mathrm{RPW}_i)$, and $\chi_i = h(\mathrm{ID}_i \parallel \mathrm{RPW}_i \parallel y_i \parallel \alpha_i)$. Then S issues U_i a smart card with $\{\beta_i, \gamma, \chi_i, h(\cdot)\}$ via a secure channel.

Step 3. U_i enters η_i to the smart card.

5.1.2. Login Phase and Authentication Phase

Step 1 ($U_i \Rightarrow S$). User U_i inserts the smart card and inputs ID_i^* and PW_i^*. Smart card computes $b = \eta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{PW}_i^*)$, $\mathrm{RPW}_i^* = h(b \parallel \mathrm{PW}_i^*)$, $\alpha_i^* = \beta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$, $y_i^* = \gamma_i \oplus h(\alpha_i^* \oplus \mathrm{RPW}_i^*)$, and $\chi_i^* = h(\mathrm{ID}_i^* \parallel \mathrm{RPW}_i^* \parallel y_i^* \parallel \alpha_i^*)$. If $\chi_i^* \neq \chi_i$, the smart card declines the request, otherwise it computes $\omega_i = y_i \oplus h(\mathrm{ID}_i \oplus \alpha_i) \oplus h(\mathrm{ID}_i \oplus \alpha_i \oplus T)$ and $\vartheta_i = h(\mathrm{ID}_i \parallel \alpha_i \parallel y_i(\alpha_i \oplus y_i) \parallel T)$ and sends $\{\mathrm{ID}, \omega_i, \vartheta_i, T\}$ to S.

Step 2 ($S \Rightarrow U_i$). Server S first checks whether $(T' - T) \le \delta T$, then computes $\alpha_i^* = h((\mathrm{ID}_i^* \oplus x) \parallel y)$, $y_i^* = \omega_i^* \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^*) \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^* \oplus T)$, and $\vartheta_i^* = h(\mathrm{ID}_i^* \parallel \alpha_i^* \parallel y_i^* \parallel (\alpha_i^* \oplus y_i^*) \parallel T)$, and further checks computed $\vartheta_i^* \stackrel{?}{=} \vartheta_i$. If the verification passed, it computes $\mu_i = h(\mathrm{ID}_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$, where T_s is S's current time, and sends $\{\mu_i, T_s\}$ to U_i.

Step 3. Smart card first checks the freshness of T_s, then computes $\mu_i^* = h(\mathrm{ID}_i \parallel y_i \parallel (\alpha_i \oplus y_i) \parallel T_s)$, and then verifies $\mu_i^* \stackrel{?}{=} \mu_i$ to authenticate server.

Step 4. Both S and U_i accept the common session key $\mathrm{SK} = h(\mathrm{ID}_i \oplus \alpha_i \oplus y_i \oplus T \oplus T_s)$.

5.2. Cryptanalysis of Kaul et al.'s Schemes

5.2.1. User Anonymity. User anonymity preserves an adversary from acquiring user's privacy message including lifestyle, habit, and hobbies by analyzing the login history, communications, and services request. In an era of big data, user anonymity has a profound significance. A well-designed protocol needs to keep the identity notion not only unexposed, but also untraceable. The former requires that even if an adversary eavesdrops the message via the open channel, he still cannot know whose communication message it is; the latter requires that the adversary does not know whether the eavesdropped message is from the same user. In fact, the latter is more restrictive than the former. However, in this scheme the user identity ID_i was exposed in the open channel; the adversary just needs to eavesdrop the open channel to get the user ID_i. With the ID_i, every time the user logs in, the adversary can know. So the privacy of the user was revealed.

5.2.2. Offline Dictionary Attack via Verification Value in Channel. $\mathscr{A}$ who extracts η_i, β_i from smart card, and $\{\mathrm{ID}, \omega_i, \vartheta_i, T\}$ can perform an offline dictionary attack as follows:

(1) Guess the value of PW_i to be PW_i^* from the password dictionary space $\mathscr{D}_{\mathrm{pw}}$.

(2) Compute $b = \eta_i \oplus h(\mathrm{ID}_i \oplus \mathrm{PW}_i^*)$, $\mathrm{RPW}_i^* = h(b \parallel \mathrm{PW}_i^*)$, $\alpha_i^* = \beta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$, $y_i^* = \omega_i^* \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^*) \oplus h(\mathrm{ID}_i^* \oplus \alpha_i^* \oplus T)$, and $\vartheta_i^* = h(\mathrm{ID}_i^* \parallel \alpha_i^* \parallel y_i^* \parallel (\alpha_i^* \oplus y_i^*) \parallel T)$, and η_i, β_i are from smart card and $\{\mathrm{ID}, \omega_i, T\}$ is from the open channel.

(3) Verify the correctness of PW_i^* by checking whether $\vartheta_i^* \stackrel{?}{=} \vartheta_i$, and ϑ_i is from smart card.

(4) Repeat Steps (1), (2), and (3) until the correct PW_i^* is found.

The time complexity is $\mathcal{O}(|\mathscr{D}_{\mathrm{pw}}| * (6T_H + 9T_R))$, so the above attack is quite efficient. With ID_i and PW_i, $\mathscr{A}$ can conduct further attack such as impersonation attack, man-in-the-middle attack, and getting session key by the ways described in Sections 3.2.3, 3.2.4, 3.2.5, and 4.3.3. Thus, the whole security of the system is compromised.

5.2.3. Offline Dictionary Attack via Verification Value in Smart Card. An adversary $\mathscr{A}$ who gets the smart card from U_i extracts security parameters $\beta_i, \gamma, \chi_i, \eta_i$. Further as shown in the previous paragraph, $\mathscr{A}$ also can easily get ID_i. So now the adversary $\mathscr{A}$ can perform an offline dictionary attack by the following steps:

(1) Guess the value of PW_i to be PW_i^* from the password dictionary space $\mathscr{D}_{\mathrm{pw}}$.

(2) Compute $b = \eta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{PW}_i^*)$, $\mathrm{RPW}_i^* = h(b \parallel \mathrm{PW}_i^*)$, $\alpha_i^* = \beta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$, $y_i^* = \gamma_i \oplus h(\alpha_i^* \oplus \mathrm{RPW}_i^*)$, and $\chi_i^* = h(\mathrm{ID}_i^* \parallel \mathrm{RPW}_i^* \parallel y_i^* \parallel \alpha_i^*)$; β_i, γ, χ_i, and η_i are extracted from U_i's smart card.

(3) Verify the correctness of PW_i^* by checking whether $\chi_i^* \stackrel{?}{=} \chi_i$; χ_i is extracted from the smart card.

(4) Repeat Steps (1), (2), and (3) until the PW_i^* is found.

The time complexity is $\mathcal{O}(|\mathscr{D}_{\mathrm{pw}}| * (5T_H + 5T_R))$, so the above attack is quite efficient.

6. A Deep Exploration to Offline Dictionary Attack

The scheme of Yeh, Kang et al., and Kaul et al. cannot resist offline-password guessing attacks, while this is exactly what most two-factor remote authentication schemes actually suffer from. As we mentioned before, such attack is also one of the original reasons for other attacks. In this section, we try to explain why it is so hard to avoid offline dictionary attack. Furthermore, we for the first time recommend distinguishing *offline dictionary attack via verification value in smart card* (hereafter called Attack I) from *offline dictionary attack via verification value in channel* (hereafter called Attack II). When talking about offline dictionary attack, most papers [36, 51, 52] ignore the difference between them and collectively call them as offline dictionary attack (offline-password guessing attack). Although the basic principles of these two attacks are the same, the key parameters transmitted in the insecure channel or in smart card, having no "camouflage" by random numbers or other special parameters only owned by the user or the server, the adversary can get a verification (usually it is the key parameter for the server or the user to verify the validity of the other one) to perform dictionary attack. Where the verifications come from is different, Attack I uses the verification from the smart card and Attack II from the channel. Do not overlook this little difference; this results in the quite slight difference in the corresponding solutions. Distinguishing them contributes to in-depth analysis of design principles. In this section, we analyze these two attacks thoroughly.

6.1. Solutions to Offline Dictionary Attack via Verification Value in Smart Card. In the schemes of Kang et al. and Kaul et al., to achieve better user-friendliness, that is, changing password locally and detecting the wrong password-inputting timely, a verification for a smart card to authenticate the user is stored in smart card. This results in $\mathscr{A}$ getting the key parameters ϑ_i and TPW_i in smart card, which leads to Attack I. What if there was no such verification parameters? Then the password change phase may be influenced, such as Yeh's scheme which changes password remotely and fails to detect wrong password input timely. In fact, [29] points out that "there is an unavoidable trade-off when achieving the password change locally and resisting offline dictionary attack." More specifically and accurately, the offline dictionary attack here should be specific to Attack I; it usually can be avoided by two ways:

(i) A new approach called "a fuzzy verifier" and "honeywords" [11], which is a new solution to such problem. This approach can greatly increase the cost of guessing password in respect of $\mathscr{A}$. And we have given a simple application case in Section 4.3.2.

(ii) Sacrificing certain performance (e.g., not providing the attribute of changing password locally). In other

words, it is a problem of the trade-off between security and effectiveness. According to this principle, some schemes [43, 53, 54] just simply remove the authentication between the user and the smart card. Therefore, this scheme is secure to Attack II, while the cost is failing to detect the wrong password input timely (costing more time).

Obviously, Attack II is not included in the above situation. So just collectively calling the two attacks as offline dictionary attack will result in confusion and making the problem more complicated.

6.2. Solutions to Offline Dictionary Attack via Verification Value in Channel. In Yeh's scheme, if we regard the parameters in smart card as opened, then the key parameters (i.e., the verification refers to the parameters used to verify the validity of the participants, and we use *Veri_Vau* to represent it) $C = f(\mathrm{PW}_i, \mathrm{ID}_i, r_3)$, where $f()$ refers to a series of cryptographic operations in the protocol, and $f(\mathrm{PW}_i, \mathrm{ID}_i, r_3)$ means that those cryptographic operations are actually only related to $\mathrm{PW}_i, \mathrm{ID}_i, r_3$. On the face of it, the *Veri_Vau* C is protected by the random number r_3. While with in-depth examination, it is clear that r_3 can be computed by PW_i. So in fact, the *Veri_Vau* $= C = f(\mathrm{PW}_i, \mathrm{ID}_i)$. Then if an adversary guesses the PW_i and ID_i to be PW_i^* and ID_i^*, he/she can use *Veri_Vau* to check the correct guessed value and thus carry out an offline dictionary attack, that is, Attack II.

Typically, only when the *Veri_Vau* was "camouflaged" by random numbers or other special parameters which only the user and the server can get can the scheme resist Attack II, such as [11, 14, 55]. So how can we conceal the *Veri_Vau* and those sensitive parameters?

Naturally, someone may think of a symmetric cryptography way: if the message transmitted in the open channel is encrypted, then only the one owning the key can read the message. It seems a good solution. However, it is far from practical: how can the key be distributed and stored? Especially to the users, where can they store that private key securely? Furthermore, with the number of servers accessed increasing, the number of the keys which the user needs to store increases too; for the servers, the storage of those keys is also a big problem; it will consume a lot of storage space, and once the storage space is leaked, the security of the system will run down. Anyway, symmetric cryptography is beyond our consideration. In this work, we focus on what Ma et al. [36] advised, that is, the public key algorithm to deal with Attack II. And here is our brief explanation about the necessary public-key algorithm (for more detail please refer to [36]).

Getting a verification (*Veri_Vau*) by the adversary in open channel is the main reason for Attack II. So a well-designed scheme has to protect the *Veri_Vau* in the open channel. Then how can we protect them? *On the one hand*, the parameters in the smart card can be captured by the adversary. So if we do not consider a symmetric cryptography, the only way is to use random numbers as a camouflage to protect the *Veri_Vau*. What is more, the random number cannot be exposed to the open channel and cannot be computed only with PW_i and ID_i. *On the other hand*, the server needs to know the random number to compute the *Veri_Vau*. So based on the two points, the public key algorithm is a necessary approach. Moreover, smart cards always have limited memory and computing power; thus the lightweight public key operations are good choices.

However, many schemes though equipped with a public key algorithm still fail to resist Attack II, such as [52, 56, 57]. The inherent reason is the incorrectness of deploying the public key algorithm. So how can we correctly apply the public key algorithm to a authentication protocol? We will give one of the solutions, although it is not the only one, but it actually is one of the many effective solutions. Furthermore, we will give an reference model of such solution (see Figure 4) and use Yeh's scheme as an example to verify its feasibility.

As we all know, the nature of a authentication protocol is to provide a secure mutual authentication. And the basis for authentication in a password-based scheme with smart card is "what you know" and "what you have." To a user, the PW_i and ID_i are what they know; the smart card is what they have. While as the parameters in the card can be extracted by the adversary and the card itself can easily be stolen, so it seems that the smart card acts as an assistant. To a server, the long term key x is what they know. The verifier table is what they have. Similarly, the verifier table actually also acts as an assistant.

So when the server authenticates the user, the user has to prove that he really knows PW_i and ID_i and has the corresponding smart card, while, in order to prevent inside attack and eavesdropping attack, the user cannot tell the server the value of PW_i directly. Then a good way to solve such conflict is to negotiate an intermediate parameter *Mid_Vau* as a key evidence of authentication, where *Mid_Vau* $= f(\mathrm{PW}_i, \mathrm{ID}_i, x)$. This step is finished in registration phase and the related auxiliary parameters are stored in the smart card. So if a user has the corresponding smart card, then he/she can computes the *Mid_Vau* as $f(\mathrm{PW}_i, \mathrm{ID}_i)$. Therefore, with the *Mid_Vau*, we can verify not only whether the user knows about the PW_i and ID_i, but also whether the user has the corresponding smart card. However, as *Mid_Vau* $= f(\mathrm{PW}_i, \mathrm{ID}_i)$ in helping with the smart card, if *Veri_Vau* $= f(Mid\_Vau)$, then once the adversary gets the smart card, he/she can guess the PW_i and ID_i to be PW_i^* and ID_i^*, respectively, then intercept the *Veri_Vau* to check the correctness of the guessed value, and thus carry out Attack II. Take Yeh's scheme as an example, in this scheme, the *Mid_Vau* $= h(\mathrm{ID}_i \parallel x)$. To U_i, *Mid_Vau* $= N_i \oplus h(\mathrm{PW}_i \parallel r_1)$, where N_i and r_1 are from smart card, *Veri_Vau* $= f(Mid\_Vau) = f(\mathrm{PW}_i, \mathrm{ID}_i)$, and thus an adversary with the card conducts Attack II successfully.

Therefore, besides the *Mid_Vau*, there should be other key parameters which only S and U_i can compute. As PW_i, ID_i, and smart card have been used for *Mid_Vau*, it makes no sense to use them again. Thus now, to a user, he/she has no more things which can be used as evidence. The one good choice seems to initiate a challenge for the server to respond. Actually, the challenge is a random number, and the transmission and response to the challenge require the help of a public key algorithm. All in all, the other key parameters which consist of *Veri_Vau* should be equipped with a public key algorithm; then the *Veri_Vau* should be $f(Mid\_Vau, Pub\_Vau)$,

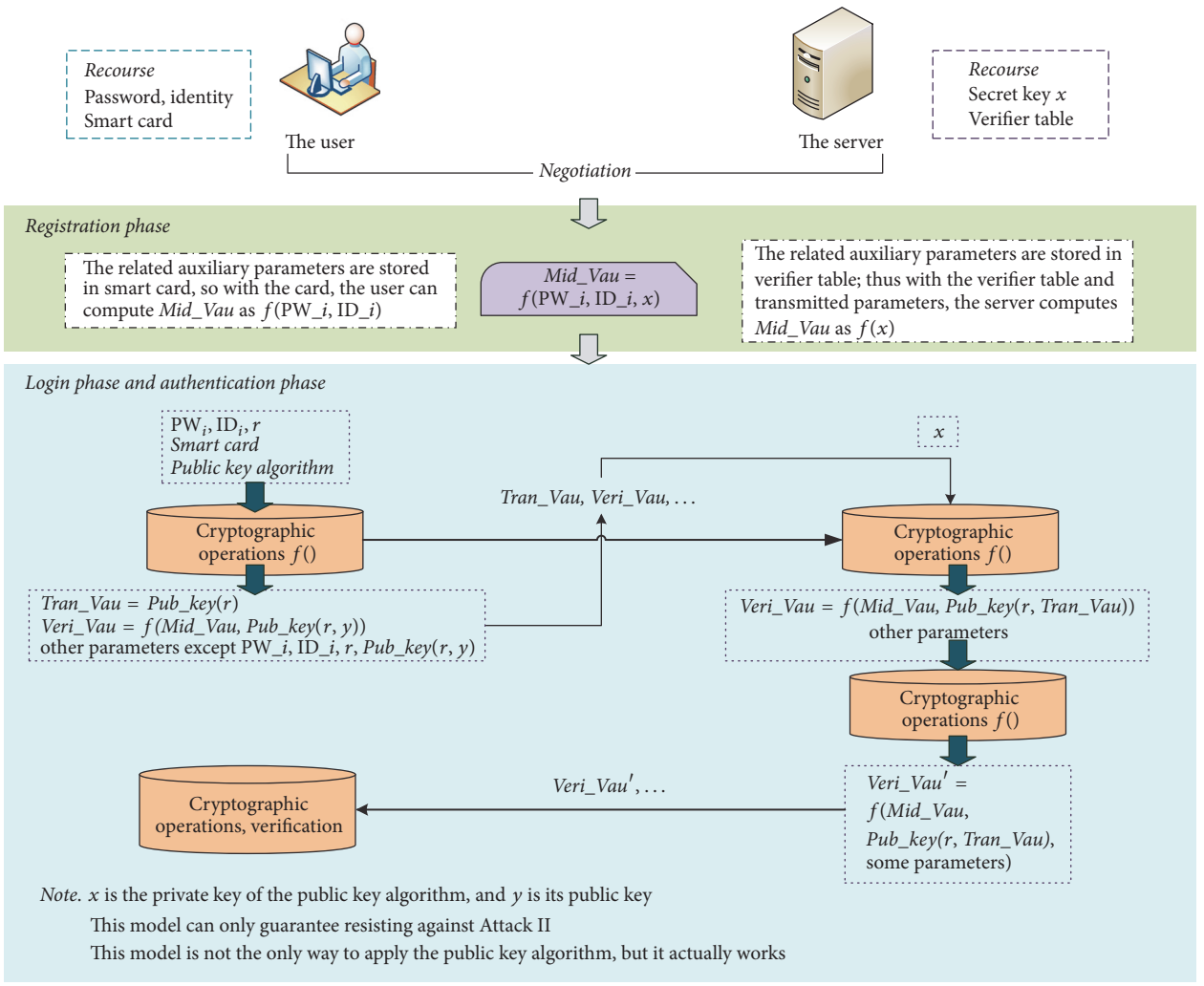


Figure 4: The reference model to employ a public key algorithm securely.

where the *Pub_Vau* refers to the parameters deploying a public key algorithm. To U_i, with the knowledge of the random number r, $Pub\_Vau = Pub\_key(r, y)$, where y is the public key and $Pub\_key(r, y)$ refer to a series of public key operations; to S, with the knowledge of private key x, $Pub\_Vau = Pub\_key(x, Tran\_Vau)$, where $Tran\_Vau = Pub\_key(r)$. All in all, to resist Attack II, the *Veri_Vau* should satisfy $Veri\_Vau = f(Mid\_Vau, Pub\_Vau)$; the *Mid_Vau*, PW_i, ID_i, r, and *Pub_Vau* cannot be exposed to the open channel; furthermore, *Tran_Vau* should be transmitted to the server. When U_i authenticates S, if only considering resisting Attack II, S only requires proving that it knows about the *Mid_Vau* and *Pub_Vau*. Furthermore, the parameters transmitted in the open channel follow the same principles as above.

Now, we take Yeh's scheme as an example to check the effectiveness of our reference model. And we select the Computational Diffie-Hellman problem to construct the public key algorithm. In Yeh et al.'s scheme, the *Mid_Vau* has been designed well, so we only need to apply the Computational Diffie-Hellman problem to this scheme. The definition of the Computational Diffie-Hellman problem is as follows.

g is the generator of a cyclic group Z_p^*; then, given g^β and g^α where $\beta, \alpha \in Z_p^*$, it is hard to compute α, β within a polynomial time.

According to this, we can design a lightweight public key algorithm for the user and the server: S selects a larger prime p, a generator g of cyclic group Z_p^*, and a secret key x and computes the public key $Y = g^x \bmod p$. Then if U_i selects a random number r_1, it computes $M_1 = g^{r_1} \bmod p$ and $M_2 = Y^{r_1} \bmod p$ and sends M_1 to S. S can compute M_2 as $M_1^x \bmod p$. Here, $M_1 = Tran\_Vau$ and $M_2 = Pub\_Vau$; even the adversary intercepts the M_1; he/she still cannot compute the M_2. So then S only needs to prove that it knows about the M_2. Furthermore, to U_i, $M_1 = Pub\_key(r)$ and $M_2 = Pub\_key(r, Y)$; to S, $M_2 = Pub\_key(x, Tran\_Vau)$; it also follows the principles mentioned above. In short, we improve Yeh's scheme as shown in Figure 5.

In the improved scheme, $Veri\_Vau = C = f(Mid\_Vau, Pub\_Vau)$; even the adversary guesses the PW_i and ID_i to be PW_i^* and ID_i^*, while without *Pub_Vau*, he/she cannot find a verification to check the correctness of the guessed value

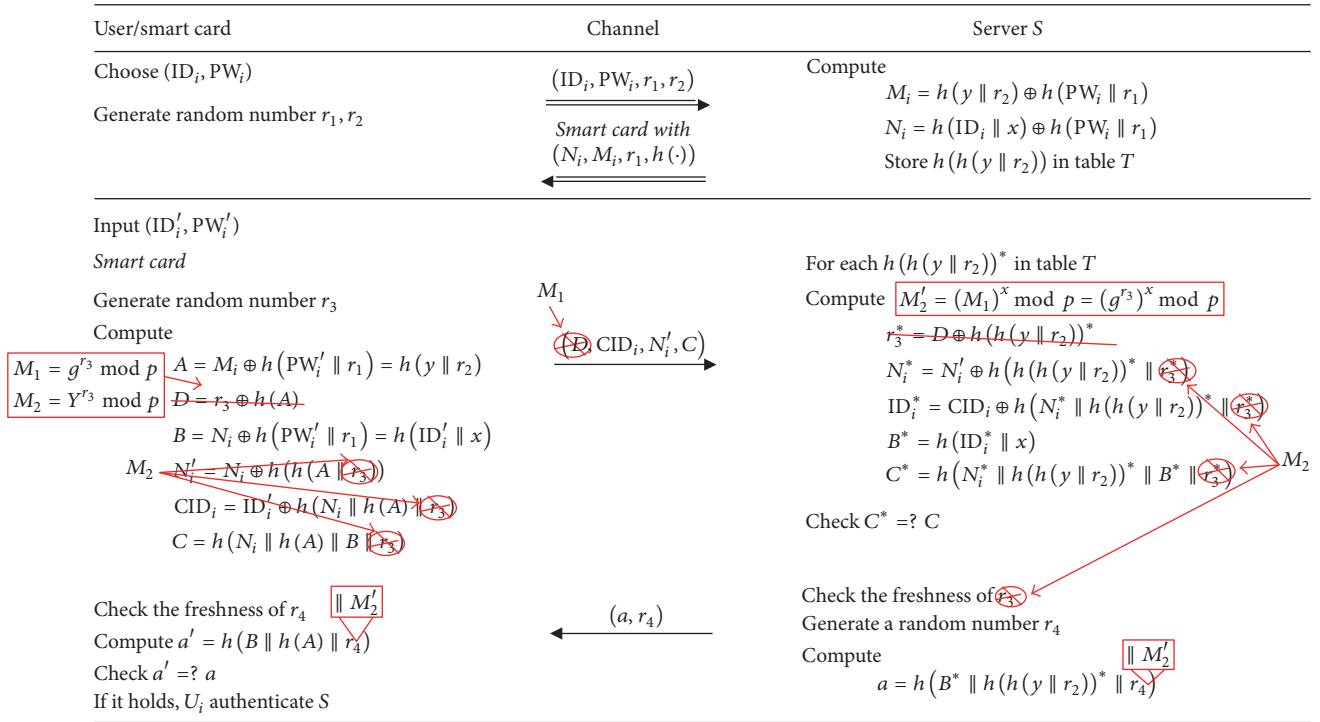


Figure 5: Improving Yeh et al.'s scheme to resist Attack II.

and thus fails to perform Attack II. It should be noted that we only improve Yeh's scheme to be secure to Attack II, and the reference model can only be applied to resisting Attack II.

It is generally accepted that public key algorithm is necessary for resisting offline dictionary attack, while, according to our analysis, the offline dictionary attack here should be specific to Attack II, and Attack I is not included in it: the public key algorithm consists of a private key and a public key. In Attack II, the vulnerability takes place in the authentication between the server and the smart card. So the public key algorithm acts on the server and the smart card. Usually the server takes the responsibility to keep private key x, while, in Attack I, the vulnerability takes place in the authentication between the user and the smart card. But it makes no sense to both of them to own such private key for two reasons at least:

(i) *To the users*: he always uses the password as the unique parameter to get authenticated, and the private key plays the same role as the password. Moreover, the private key is too long for the user to remember or preserve.

(ii) *Smart card*: as we have stated before, the parameters in the smart card can be easily obtained by an adversary.

7. Conclusion

In this paper, we demonstrated that the schemes of Yeh, Kang et al., and Kaul et al. all suffer from various attacks, such as offline dictionary attack and impersonation attack. Furthermore, we showed that offline dictionary attack is the original reason of many other attacks. Remarkably, we divide offline dictionary attacks into two categories: (1) the ones using the verification from smart cards and (2) the ones using the verification from the open channel. The solution to the first type involves the trade-off between the security and effectiveness, or "a fuzzy verifier" + "honeywords" as suggested in [11]. While the solution to the second is using a public key algorithm as advised by Ma et al. [36], this solution is not applicable to the first type. Furthermore, even many schemes using a public key algorithm still suffer from Attack II. The original reason is incorrectly deploying the public key algorithms. Thus, we proposed a reference model to guide the protocol designers to deploy the public key algorithms correctly. Our reference model is not the only way to deal with such problem, but it really is one of them. We hope that this work provides new insights into future research.

Notations and Abbreviations

U_i: ith user
S: Remote server
$\mathcal{A}$: Malicious adversary
ID_i: Identity of U_i
PW_i: Password of U_i
x: The secret key of S
y: The secret number of S
$h(\cdot)$: Collision-free one-way hash function
$\oplus$: The bitwise XOR operation
$\|$: The string concatenation operation
$\rightarrow$: An insecure channel
$\Rightarrow$: A secure channel.

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Dr. Wang Ding at Peking University for invaluable help. This research is supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61401038 and 2016 Guangdong Provincial Science and Technology Department Frontier and Key Technology Innovation Project under Grant no. 2016B010110002.

References

[1] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards Privacy-preserving Content-based Image Retrieval in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[2] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

[3] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM '16)*, pp. 1–9, San Francisco, Calif, USA, April 2016.

[4] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ecc for wireless sensor networks," *Journal of Network & Computer Applications*, vol. 76, pp. 37–48, 2016.

[5] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[6] A. Shimizu, "Dynamic password authentication method using a one-way function," *Systems and Computers in Japan*, vol. 22, no. 7, pp. 32–40, 1991.

[7] S.-P. Shieh, W.-H. Yang, and H.-M. Sun, "An authentication protocol without trusted third party," *IEEE Communications Letters*, vol. 1, no. 3, pp. 87–89, 1997.

[8] T. Hwang, "Password authentication using public-key encryption," in *Proceedings of IEEE Int. Carnahan Conf. Security Technol.*, pp. 141–144, 1983.

[9] T. Hwang, Y. Chen, and C. J. Laih, "Non-interactive password authentications without password tables," in *Proceedings of IEEE Region 10 Conference on Computer and Communication Systems Conference*, pp. 429–431, Hong Kong.

[10] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proceedings E: Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[11] D. Wang and P. Wang, "Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-1.

[12] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2011.

[13] D. He, S. Zeadally, B. Xu, and X. Huang, "An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[14] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, Article ID 11496, pp. 162–178, 2015.

[15] V. Odelu, A. K. Das, and A. Goswami, "A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[16] C. Yuan, X. Sun, and R. Lv, "Fingerprint liveness detection based on multi-scale LPQ and PCA," *China Communications*, vol. 13, no. 7, pp. 60–65, 2016.

[17] D. He and D. Wang, "Robust Biometrics-Based Authentication Scheme for Multiserver Environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[18] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[19] M. Khan and S. Kim, "Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[20] Y.-F. Chang, W.-L. Tai, and H.-C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[21] H.-Y. Chien and C.-H. Chen, "A remote authentication scheme preserving user anonymity," in *19th International Conference on Advanced Information Networking and Applications, AINA 2005*, pp. 245–248, twn, March 2005.

[22] Y.-Y. Wang, J.-Y. Liu, F.-X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[23] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using AM demodulation on commercial smart cards with SEED," *Journal of Systems and Software*, vol. 85, no. 12, pp. 2899–2908, 2012.

[24] K. Nohl, D. Evans, S. Starbug, and H. Plötz, "Reverse-engineering a cryptographic rfid tag," in *Proceedings of USENIX Security*, pp. 185–193, San Jose, CA, USA.

[25] T. S. Messerges, E. A. Dabbish, and R. . Sloan, "Examining smart-card security under the threat of power analysis attacks," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[26] A. Moradi, A. Barenghi, T. Kasper, and C. Paar, "On the vulnerability of FPGA bitstream encryption against power analysis attacks: Extracting keys from Xilinx Virtex-II FPGAs," in *18th ACM Conference on Computer and Communications Security, CCS'11*, pp. 111–123, usa, October 2011.

[27] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

[28] J. L. Tsai, N. W. Lo, and T. C. Wu, "Novel anonymous authentication scheme using smart cards," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2004–2013, 2013.

[29] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[30] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards and Interfaces*, vol. 32, no. 5-6, pp. 321–325, 2010.

[31] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards and Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[32] K. S. Sandeep, K. S. Anil, and S. Kuldip, "An improvement of xu et al.'s authentication scheme using smart cards," in *Proceedings of the ACM 3th Bangalore Conference*, vol. 1, pp. 240–245, Bangalore, India, 2010.

[33] B.-L. Chen, W.-C. Kuo, and L.-C. Wuu, "Robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 27, no. 2, pp. 377–389, 2014.

[34] S. Kumari and M. K. Khan, "Cryptanalysis and improvement of a robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 27, no. 12, pp. 3939–3955, 2014.

[35] X. Li, J. Niu, M. Khurram Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365–1371, 2013.

[36] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2014.

[37] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: A comparative evaluation of two-factor authentication schemes," in *Proceedings of 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS*, pp. 475–486, China, June 2016.

[38] Y. Wang, "Password protected smart card and memory stick authentication against off-line dictionary attacks," *IFIP Advances in Information and Communication Technology*, vol. 376, pp. 489–500, 2012.

[39] R. Madhusudhan and R. C. Mittal, "Dynamic ID-based remote user password authentication schemes using smart cards: a review," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1235–1248, 2012.

[40] X. Li, Y. Xiong, J. Ma, and W. Wang, "An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network & Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[41] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

[42] S. Kumari, M. K. Khan, X. Li, and F. Wu, "Design of a user anonymous password authentication scheme without smart card," *International Journal of Communication Systems*, vol. 29, no. 3, pp. 441–458, 2016.

[43] K.-H. Yeh, "A lightweight authentication scheme with user untraceability," *Frontiers of Information Technology and Electronic Engineering*, vol. 16, no. 4, pp. 259–271, 2015.

[44] D. Kang, J. Jung, J. Mun, D. Lee, Y. Choi, and D. Won, "Efficient and robust user authentication scheme that achieve user anonymity with a markov chain," *Security & Communication Networks*, 2016.

[45] B. Djellali, K. Belarbi, A. Chouarfia, and P. Lorenz, "User authentication scheme preserving anonymity for ubiquitous devices," *Security and Communication Networks*, vol. 8, no. 17, pp. 3131–3141, 2015.

[46] S. D. Kaul and A. K. Awasthi, "Security Enhancement of an Improved Remote User Authentication Scheme with Key Agreement," *Wireless Personal Communications*, pp. 1–17, 2016.

[47] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, 2016.

[48] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[49] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted Online Password Guessing," in *Proceedings of ACM CCS 16*, pp. 1242–1254, Vienna, Austria, October 2016.

[50] D. Wang and P. Wang, "On the implications of zipfs law in passwords," *Proceedings of ESORICS*, pp. 111–131, 2016.

[51] S. H. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *International Journal of Communication Systems*, vol. 29, no. 11, pp. 1708–1719, 2016.

[52] D. Wang, C. Ma1, and P. Wu, "Secure password-based remote user authentication scheme with non-tamper resistant smart cards," in *Proceedings of DBSec*, vol. 7371, pp. 114–121, France, Paris, 2012.

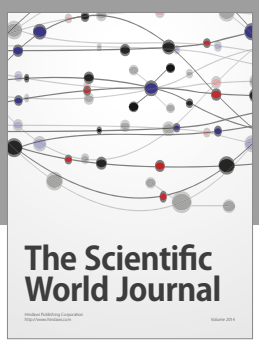
[53] H. Zhu, "Cryptanalysis and Improvement of a Mobile Dynamic ID Authenticated Key Agreement Scheme Based on Chaotic Maps," *Wireless Personal Communications*, vol. 85, no. 4, pp. 2141–2156, 2015.

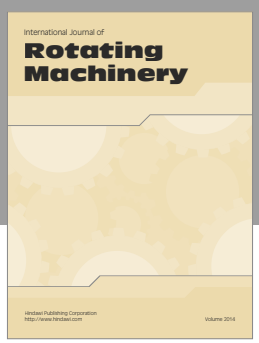
[54] J. Wei, W. Liu, and X. Hu, "Secure and efficient smart card based remote user password authentication scheme," *International Journal of Network Security*, vol. 18, no. 4, pp. 782–792, 2016.

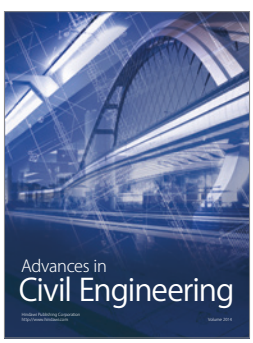
[55] Q. Jiang, J. Ma, and Y. Tian, "Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of Zhang et al.," *International Journal of Communication Systems*, vol. 28, no. 7, pp. 1340–1351, 2015.

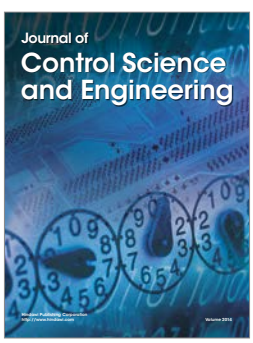
[56] K. Marimuthu and R. Saravanan, "A secure remote user mutual authentication scheme using smart cards," *Journal of Information Security & Applications*, vol. 19, no. (4-5), pp. 282–294, 2014.

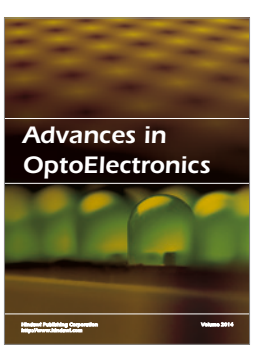
[57] T. Maitra, M. Obaidat, R. Amin, S. Islam, S. Chaudhry, and D. Giri, "A robust elgamal-based password-authentication protocol using smart card for client-server communication," *International Journal of Communication Systems*, 2016.

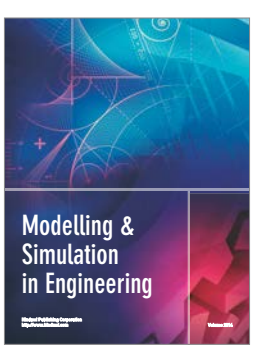


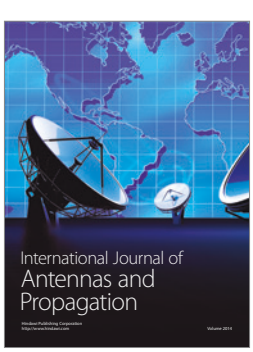


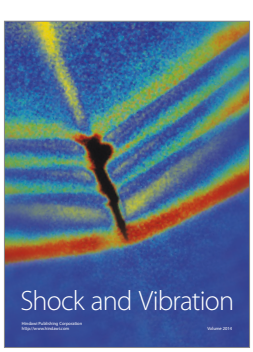


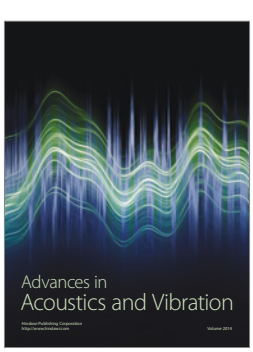












Hindawi

Submit your manuscripts at
https://www.hindawi.com