On Security of Privacy-Preserving Remote User Authentication with k-Times Untraceability

Qijia Zhang[1], Jianhong Zhang[1,2], Linhan Liu[1], Jing Wang[3], and Pei Liu[3]
(Corresponding author: Jianhong Zhang)
School of Information Sciences and Technology, North China University of Technology[1]
Beijing 100144, China
GuiZhou University,Guizhou Provincial Key Laboratory of Public Big Data[2]
Guizhou Guiyang 550025, China
Beijing Jingdong Century Information Technology Company, Limited[3]
Beijing 100100, China
Email: zjhncut@163.com
(Received Dec. 31, 2019; Revised and Accepted July 23, 2020; First Online Apr. 17, 2021)

Abstract

As an important access control technique, k-times anonymous authentication (k-TAA) plays a vital role in e-coupon and e-bill. It allows a user to anonymously authenticate himself to a remote server a bounded number of times. However, most of the existing k-TAA schemes require heavy computation, which brings a challenge to resource-limited devices. In 2018, Tian *et al.* proposed a privacy-preserving remote user authentication with k-times untraceability. Unlike the traditional k-TAA schemes, Tian *et al.*'s is more suitable for mobile devices due to avoiding expensive pairing operations. And they claim that their scheme provides user authenticity and k-times untraceability. Unfortunately, in this paper, we find that their scheme is insecure by analyzing it. Their scheme can neither prevent a malicious user from passing the authentication nor trace the identity of a dishonest user authenticating for more than k times. Finally, the corresponding attacks are given.

Keywords: Anonymity; Attack; Authentication; User Privacy

1 Introduction

The development of internet makes it possible that people enjoy the service of remote providers. In the past few decades, various online applications and services emerged. In the most typical online service, users need to interact with a server. The first step of interaction is user authentication. Before opting for services, a user has to authenticate himself to an authentication server, and the server saves this user's identity into its database. However, in some cases, identity is a part of the users' privacy information that should be protected. In this situation, anonymity is one of the fundamental security properties that a secure system should provide. In order to preserve the privacy of users, anonymous remote user authentication is proposed and applied in a lot of fields [6, 7, 11, 14]. It allows users to anonymously authenticate themselves to a remote authentication server. To improve anonymous remote user authentication, researchers have exploited many cryptographic techniques in it, such as group signature [1], blind signature [8, 10], and ring signature [19].

However, sometimes the anonymity of users may bring harm to the system. For example, in the e-coupon system [5, 9, 20], for any customer who purchasing goods in a shop, the merchant can grant an electronic coupon as a bonus to the customer. To benefit from the service, a dishonest user may redeem this electronic coupon for more than the predetermined number of times. In this case, the service provider should be able to trace this dishonest user and obtain his identity. However, most of existing anonymous remote user authentication schemes neglect traceability against the dishonest users.

To address this issue, Teranishi, Furukawa and Sako [15] proposed *k-times anonymous remote user authentication* (k-TAA) scheme. It is a fine-grained method of privacy-preserving which enables users to anonymously authenticate themselves for a bounded number of times. If a dishonest user authenticate himself beyond the predetermined number of times, he will not remain anonymous, which means the service provider can immediately trace the identity of him. However, k-TAA can not support that a service provider control over giving users access permission to his service, so it is not flexible enough. Shortly after that, in order to make service providers have better control over their users, Nguyen and Safavi Naini [13] proposed dynamic k-TAA scheme, which allows service providers to grant and revoke the access of registered users. Dynamic k-TAA allows service providers to restrict

access to their services based on not only the number of times, but also other factors such as expiry date. So it can be used in a much wider range of realistic scenarios.

Later on, Nguyen [12] proposed an efficient dynamic k-TAA scheme, where computation and communication costs are constant and do not depend on the limited number k. After that, a lot of schemes [3, 4, 16, 18] that focus on reducing the computation and communication costs of k-TAA are proposed. In 2006 , Au *et al.* [2] optimized Nguyen's dynamic k-TAA scheme and reduced time complexity from $O(k)$ to $O(log(k))$.

Although k-TAA and dynamic k-TAA realize the anonymity in authentication and k-times untraceability against dishonest users, how to apply them to mobile devices is still a challenge. With the breakthrough of some key technologies in communication (*e.g.* 5G), people tend to rely on mobile devices, such as smartphones. These devices are low-power and limited resources, which means they can not handle complex cryptographic operations. However, most of existing k-TAA schemes require certain computation-intensive operations such as pairing and proof of knowledge. Therefore, it is infeasible to simply apply k-TAA schemes to these weak devices with low performance.

Recently, tian *et al.* proposed a privacy-preserving remote user authentication with k-times untraceability scheme [17], which avoids expensive pairing operations and makes it possible to apply to mobile devices. They claimed that their scheme supports user authenticity and k-times untraceability. In other words, a malicious third party can not impersonate an authorized user. Besides, the authority can trace the real identities of dishonest users who have authenticated themselves for more than k times. However, in this work, we find that their scheme is not as secure as they claimed. Their scheme does not satisfy user authenticity or k-time untraceability. That is to say, by forging a credential, a dishonest user who do not enroll to the server can successfully pass the authentication. Moreover, the dishonest user can still keep anonymous after authenticating for more than k times.

The rest of this paper is organized as follows. In Section 2, we review the system model and security goals of Tian *et al.*'s scheme. Then, we briefly review Tian *et al.*'s scheme in Section 3. In Section 4, we give two concrete attacks and analyze the corresponding reasons. Finally, we draw our conclusion in Section 5.

2 Preliminary

2.1 System Model

There are two entities in Tian *et al.*'s privacy-preserving remote user authentication with k-times untraceability scheme: Enrolled users and an authentication server.

Users: The users should generate their key pairs, and enroll themselves. After enrolling themselves, they will generate a valid credential and submit it to the authentication server in the authentication phase. What's more, they need to generate k-size commitments and send them to the authentication server.

Authentication Server: The authentication server is an honest-but-curious entity. It is in charge of generating master key pair and authenticate users. If there is a dishonest user who authenticates himself for more than k times, the authentication server will detect his misbehavior and trace his real identity.

2.2 Security Goals

A k-times anonymous remote user authentication scheme should satisfy the following properties:

User Authenticity: An authorized user can generate a valid and legal credential to anonymously authenticate himself. During a session, a third party cannot impersonate an authorized user and forge a credential to pass the authentication.

k-time Untraceability: Users are only allowed to anonymously authenticate for k times. If there is a dishonest user who authenticate for more than k times, he will not remain anonymous, and the authentication server will be able to trace his real identity.

3 Reviews of Tian *et al.*'s Scheme

In this section, we briefly review Tian *et al.*'s scheme. Their scheme includes 5 phases: System initialization, key generation, enrollment, authentication, trace. These phases are given as follows.

3.1 System Initialization

Authentication server $\mathbb{S}$ takes a secure parameter as input, and output a multiplicative cyclic group $\mathbb{G}$ with order q. Let g denote a generator of $\mathbb{G}$. $\mathbb{S}$ chooses two hash function: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_2 : \mathbb{G} \rightarrow \mathbb{Z}_q$. $\mathbb{S}$ chooses $y, e, f \in \mathbb{Z}_q$, then computes $g^y, g_1 = g^e, g_2 = g^f, g_1^y, g_2^y$. Finally secretly keep master secret key: $msk = (y, e, f)$, and publish the public parameters:

$$PK = (\mathbb{G}, q, g, g^y, g_1, g_2, g_1^y, g_2^y).$$

3.2 Key Generation

The user i randomly chooses the secret key $x \in \mathbb{Z}_q$ as his secert key sk_i. Then uses system public key to computes g_1^x as his public key pk_i.

3.3 Enrollment

In this part, a user gets his credential by interacting with authentication server $\mathbb{S}$.

1) Firstly user i requests authentication server $\mathbb{S}$ for enrollment. Upon receiving i's request, $\mathbb{S}$ randomly chooses $\mathcal{K} \in \mathbb{Z}_q$, and computes:

$$\delta_1 = g^{\mathcal{K}}, \delta_2 = (g_1^x g_2)^{\mathcal{K}},$$

and sends them to user i.

2) User i randomly chooses $x_1, a, b \in \mathbb{Z}_q$, and computes:

$$\begin{aligned} \alpha &= (g_1^x g_2)^{y \cdot x_1}, \\ \beta &= (g_1^x g_2)^{x_1}, \\ m &= H_1(\alpha||\beta), \\ r &= m \cdot \beta^a \cdot \delta_2^{b \cdot x_1}, \\ m' &= H_2(m||r)/b, \end{aligned}$$

and sends m' to $\mathbb{S}$.

3) Upon receiving m', the authentication server $\mathbb{S}$ computes blinded signature:

$$s' = \mathcal{K} + y \cdot m',$$

and sends s' to user i.

4) Upon receiving s', user i checks

$$g^{s'} \stackrel{?}{=} g^{y \cdot m'}.$$

If it does not hold, abort. Otherwise, user computes

$$s = s' \cdot b + a,$$

and keeps (α, β, r, s) as a valid credential.

3.4 Authentication

In this part, $\mathbb{S}$ authenticates valid users.

1) Before making request for authentication, user i randomly chooses $s_1, s_2, ..., s_k \in \mathbb{Z}_q$ and computes two k-size sets:

$$\begin{aligned} (S_1, S_2, ..., S_k) &= (g_1^{x \cdot s_1}, g_1^{x \cdot s_2}, ..., g_1^{x \cdot s_k}) \\ (\bar{S}_1, \bar{S}_2, ..., \bar{S}_k,) &= (g_2^{s_1}, g_2^{s_2}, ..., g_2^{s_k}). \end{aligned}$$

Then encrypts them and generates ciphertext $C_i = Enc(S_i, \bar{S}_i)$ by using master public key, and sends it to the authentication server $\mathbb{S}$.

2) Upon receiving the request from the user i, $\mathbb{S}$ randomly chooses $c_i \in \mathbb{Z}_q$ and sends it to user i.

3) User i computes

$$\begin{aligned} R_1 &= x_1 + s_1 \cdot c_i + s_2 \cdot c_i^2 + ... + s_k \cdot c_i^k, \\ R_2 &= x \cdot R_1, \end{aligned}$$

and sends $(R_1, R_2, \alpha, \beta, r, s)$ to $\mathbb{S}$.

4) $\mathbb{S}$ checks whether user i's credential is valid:

$$H_1(\alpha||\beta) \stackrel{?}{=} \beta^{-s} \cdot \alpha^{H_2(H_1(\alpha||\beta)||r)} \cdot r. \quad (1)$$

If it does not hold, aborts. Otherwise, checks whether the following equation holds:

$$g_1^{R_2} \cdot g_2^{R_1} \stackrel{?}{=} \beta \cdot S_1^{c_i} \cdot S_2^{c_i^2} \cdots S_k^{c_i^k} \cdot \bar{S_1}^{c_i} \cdot \bar{S_2}^{c_i^2} \cdots \bar{S_k}^{c_i^k}.$$

If it does not hold, aborts. Otherwise, $\mathbb{S}$ authenticates user i.

3.5 Trace

For each authentication request, $\mathbb{S}$ randomly chooses a c_i and sends it to the user i, which is used to compute R_1 and R_2. Therefore, the user i generates different R_1 and R_2 every time. If there is a dishonest user i who maliciously uses a valid credential more than k times, the authentication server $\mathbb{S}$ will be able to get at least $k+1$ different R_1 and R_2. Then $\mathbb{S}$ has the following equations:

$$\begin{cases} R_{1_1} = x_1 + s_1 \cdot c_{i_1} + s_2 \cdot c_{i_1}^2 + ... + s_k \cdot c_{i'_1}^k \\ R_{1_2} = x_1 + s_1 \cdot c_{i_2} + s_2 \cdot c_{i_2}^2 + ... + s_k \cdot c_{i'_2}^k \\ \quad ... \\ R_{1_{k+1}} = x_1 + s_1 \cdot c_{i_{k+1}} + s_2 \cdot c_{i_{k+1}}^2 + ... + s_k \cdot c_{i_{k+1}}^k \end{cases}, \quad (2)$$

and

$$\begin{cases} R_{2_1} = x \cdot (x_1 + s_1 \cdot c_{i_1} + s_2 \cdot c_{i_1}^2 + ... + s_{k_1} \cdot c_{i_1}^k) \\ R_{2_2} = x \cdot (x_1 + s_1 \cdot c_{i_2} + s_2 \cdot c_{i_2}^2 + ... + s_k \cdot c_{i_2}^k) \\ \quad ... \\ R_{2_{k+1}} = x \cdot (x_1 + s_1 \cdot c_{i_{k+1}} + s_2 \cdot c_{i_{k+1}}^2 + ... + s_k \cdot c_{i_{k+1}}^k) \end{cases}. \quad (3)$$

By calculating (2) and (3), $\mathbb{S}$ can get x_1 and $x_1 \cdot x$ respectively. Obviously, the secret key sk_i of the user i can be successfully obtained by $\mathbb{S}$. After getting the secret key of the user i, the authentication server $\mathbb{S}$ can easily trace the real identity of this user.

4 Attacks on Tian *et al.*'s Scheme

In this section, we show the detail attacks on Tian *et al.*'s scheme. By analyzing it, we show that their scheme satisfies neither user authenticity nor k-times untraceability. This means that a dishonest user can authenticate himself without enrollment by forging a credential, and the dishonest user can still keep anonymous after authenticating himself more than k times. The detail attacks are given as follows.

4.1 Attack on User Authenticity

We assume a misbehaving user i', who does not enroll himself to the authentication server $\mathbb{S}$ or generate a valid credential. However, i' can forge a credential and pass

the verifying equations. Since the server $\mathbb{S}$ cannot distinguish him from other honest users, it will allow him to be authenticated. The description of this attack is given as follows.

In order to forge a credential without enrollment, user i' randomly chooses $x', x'_1 \in \mathbb{Z}_q$ and computes:

$$\begin{aligned}\alpha' &= (g_1^{x'} g_2)^{x'_1},\\ \beta' &= \alpha',\\ r' &= H_1(\alpha'||\beta'),\\ s' &= H_2(H_1(\alpha'||\beta')||r'),\end{aligned}$$

and saves $(\alpha', \beta', r', s')$ as his forged credential.

To authenticate himself by using this forged credential, user i' needs to interact with $\mathbb{S}$ as below.

1) User i' randomly chooses $s_1, s_2, ..., s_k \in \mathbb{Z}_q$ and computes two k-size sets:

$$\begin{aligned}(S_1, S_2, ..., S_k) &= (g_1^{x'\cdot s_1}, g_1^{x'\cdot s_2}, ..., g_1^{x'\cdot s_k})\\ (\bar{S_1}, \bar{S_2}, ..., \bar{S_k},) &= (g_2^{s_1}, g_2^{s_2}, ..., g_2^{s_k}).\end{aligned}$$

Then i' encrypts them and generates ciphertext $C'_i = Enc(S'_i, \bar{S'_i})$ by using master public key, and sends it to the authentication server $\mathbb{S}$ as his authentication request.

2) Upon receiving the request from the user i', $\mathbb{S}$ randomly chooses $c_{i'} \in \mathbb{Z}_q$ and sends it to user i'.

3) Upon receiving c', user i' computes:

$$\begin{aligned}R'_1 &= x'_1 + s_1 \cdot c_{i'} + s_2 \cdot c_{i'}^2 + ... + s_k \cdot c_{i'}^k,\\ R'_2 &= x' \cdot R_1,\end{aligned}$$

and sends $(R'_1, R'_2, \alpha', \beta', r', s')$ to $\mathbb{S}$.

4) $\mathbb{S}$ checks whether user i's credential $(\alpha', \beta', r', s')$ is valid:

$$H_1(\alpha'||\beta') \stackrel{?}{=} \beta'^{-s'} \cdot \alpha'^{H_2(H_1(\alpha'||\beta')||r')} \cdot r'. \quad (4)$$

The Equation (4) can hold correctly. Because we have:

$$\begin{aligned}&\beta'^{-s'} \cdot \alpha'^{H_2(H_1(\alpha'||\beta')||r')} \cdot r'\\ =&\beta'^{-H_2(H_1(\alpha'||\beta')||r')} \cdot \beta'^{H_2(H_1(\alpha'||\beta')||r')} \cdot r'\\ =&r'\\ =&H_1(\alpha'||\beta').\end{aligned}$$

Then $\mathbb{S}$ checks whether the following equation holds:

$$g_1^{R'_2} \cdot g_2^{R'_1} \stackrel{?}{=} \beta' \cdot S_1^{c_{i'}} \cdot S_2^{c_{i'}^2} \cdots S_k^{c_{i'}^k} \cdot \bar{S_1}^{c_{i'}} \cdot \bar{S_2}^{c_{i'}^2} \cdots \bar{S_k}^{c_{i'}^k}. \quad (5)$$

The Equation (5) can hold correctly. Because we have:

$$\begin{aligned}&\beta' \cdot S_1^{c_{i'}} \cdot S_2^{c_{i'}^2} \cdots S_k^{c_{i'}^k} \cdot \bar{S_1}^{c_{i'}} \cdot \bar{S_2}^{c_{i'}^2} \cdots \bar{S_k}^{c_{i'}^k}\\ =&(g_1^x g_2)^{x'_1} \cdot g_1^{x' s_1 c_{i'}} \cdot g_1^{x' s_2 c_{i'}^2} \cdots g_1^{x' s_k c_{i'}^k} \cdot g_2^{s_1 c_{i'}}\\ &\cdot g_2^{s_2 c_{i'}^2} \cdots g_2^{s_k c_{i'}^k}\\ =&g_1^{x'(x'_1 + s_1 c_{i'} + s_2 c_i^2 + \cdots + s_k c_{i'}^k)} \cdot g_2^{(x'_1 + s_1 c_{i'} + s_2 c_{i'}^2 + \cdots + s_k c_{i'}^k)}\\ =&g_1^{R'_2} \cdot g_2^{R'_1}.\end{aligned}$$

Both of the above two equations hold, so $\mathbb{S}$ authenticates the misbehaving user i'.

This attack indicates that Tian *et al.*'s scheme can not satisfy unforgeability. We show that a misbehaving user i' can successfully authenticate himself to the authentication server $\mathbb{S}$ without enrollment by forging a credential. What's more, the misbehaving user i' can use the same $(C_{i'}, R'_1, R'_2, \alpha', \beta', r', s')$ up to k times to authenticate himself to $\mathbb{S}$ without being detected.

4.2 Attack on k-Time Untraceability

Tian *et al.* claims their scheme can achieve k-time untraceability, which allows an honest user be authenticated anonymously only up to k times. If there is a dishonest user who authenticated himself more than k times, he will not remain anonymous and the authentication server will trace his real identity. We give an attack on it and show that in Tian *et al*'s scheme, a dishonest user can keep anonymous after authenticating for $k+1$ times. The detail of attack is described as follows:

User i' randomly chooses $x', x'_1 \in \mathbb{Z}_q$ and forges a credential:

$$\begin{aligned}\alpha' &= (g_1^{x'} g_2)^{x'_1},\\ \beta' &= \alpha',\\ r' &= H_1(\alpha'||\beta'),\\ s' &= H_2(H_1(\alpha'||\beta')||r').\end{aligned}$$

User i' uses this forged credential to authenticate himself as below.

1) User i' randomly chooses $s_1, s_2, ..., s_k \in \mathbb{Z}_q$ and computes two k-size sets:

$$\begin{aligned}(S_1, S_2, ..., S_k) &= (g_1^{x'\cdot s_1}, g_1^{x'\cdot s_2}, ..., g_1^{x'\cdot s_k}),\\ (\bar{S_1}, \bar{S_2}, ..., \bar{S_k},) &= (g_2^{s_1}, g_2^{s_2}, ..., g_2^{s_k}).\end{aligned}$$

Then i' encrypts them and generates ciphertext $C'_i = Enc(S'_i, \bar{S'_i})$ by using master public key, and sends it to the authentication server $\mathbb{S}$ as his authentication request.

2) Upon receiving the request from the user i', $\mathbb{S}$ randomly chooses $c_{i'} \in \mathbb{Z}_q$ and sends it to user i'.

3) Upon receiving c', user i' computes:

$$R'_1 = x'_1 + s_1 \cdot c_{i'} + s_2 \cdot c_{i'}^2 + ... + s_k \cdot c_{i'}^k,$$
$$R'_2 = x' \cdot R_1,$$

and sends $(R'_1, R'_2, \alpha', \beta', r', s')$ to $\mathbb{S}$.

4) $\mathbb{S}$ checks whether the following two equations are hold:

$$H_1(\alpha'||\beta') \stackrel{?}{=} \beta'^{-s'} \cdot \alpha'^{H_2(H_1(\alpha'||\beta')||r')} \cdot r', \quad (6)$$

$$g_1^{R'_2} \cdot g_2^{R'_1} \stackrel{?}{=} \beta' \cdot S_1^{c_{i'}} \cdot S_2^{c_{i'}^2} \cdots S_k^{c_{i'}^k} \cdot \bar{S_1}^{c_{i'}} \cdot \bar{S_2}^{c_{i'}^2} \cdots \bar{S_k}^{c_{i'}^k}. \quad (7)$$

According to the attack on authentication, we can know that Equation (6) and Equation (7) are both hold, and $\mathbb{S}$ authenticates the misbehaving user i'.

5) If this misbehaving user i' uses his forged credential to authenticate himself for more than k times, the authentication server $\mathbb{S}$ will detect his misbehavior and try to trace him. $\mathbb{S}$ gets at least $k+1$ different R'_1 and R'_2 with different $c_{i'}$. Then $\mathbb{S}$ computes the following equations:

$$\begin{cases} R'_{1_1} = x'_1 + s_1 \cdot c_{i'_1} + s_2 \cdot c_{i_1^*}^2 + ... + s_k \cdot c_{i'_1}^k \\ R'_{1_2} = x'_1 + s_1 \cdot c_{i'_2} + s_2 \cdot c_{i'_2}^2 + ... + s_k \cdot c_{i'_2}^k \\ \quad ... \\ R'_{1_{k+1}} = x'_1 + s_1 \cdot c_{i'_{k+1}} + s_2 \cdot c_{i'_{k+1}}^2 + ... \\ \qquad + s_k \cdot c_{i'_{k+1}}^k \end{cases}, \quad (8)$$

and

$$\begin{cases} R'_{2_1} = x' \cdot (x'_1 + s_1 \cdot c_{i'_1} + s_2 \cdot c_{i'_1}^2 + ... + s'_k \cdot c_{i'_1}^k) \\ R'_{2_2} = x' \cdot (x'_1 + s_1 \cdot c_{i'_2} + s_2 \cdot c_{i'_2}^2 + ... + s_k \cdot c_{i'_2}^k) \\ \quad ... \\ R'_{2_{k+1}} = x' \cdot (x'_1 + s_1 \cdot c_{i'_{k+1}} + s_2 \cdot c_{i'_{k+1}}^2 + ... \\ \qquad + s_k \cdot c_{i'_{k+1}}^k) \end{cases}, \quad (9)$$

to get x'_1 and $x'_1 \cdot x'$ respectively. Then $\mathbb{S}$ can easily obtain the secret key of i', which is x'.

After getting x', $\mathbb{S}$ computes $g_1^{x'}$ and try to find out the identity whose public key matches this value. However, $\mathbb{S}$ could not determine the identity of dishonest user i' in its database since there is no public key of i'.

This attack indicates that Tian *et al.*'s k-RUA scheme does not satisfy k-time untraceability. We show that by forging a credential, a misbehaving user can not only successfully authenticate himself to the authentication server, but also keep anonymous after authenticating for more than k times.

4.3 Discussion

In the following, we analyze the reason to lead to the above attacks and provide our suggestions. In authentication phase of Tian *et al.*'s scheme, we can find that any one can forge a credential to impersonate an authorized user and pass authentication. We can see that in the Equation (1) hash value $H_2(H_1(\alpha||\beta)||r)$ is irrelevant to s. This makes the verifying Equation (1) vulnerable. In this situation, attackers can randomly choose s. When setting $s' = H_2(H_1(\alpha||\beta)||r)$, attackers can forge a credential $(R'_1, R'_2, \alpha', \beta', r', s')$ to pass authentication. To improve Tian *et al.*'s scheme, we suggest applying group signature technique to associate hash value $H_2(H_1(\alpha||\beta)||r)$ with s. With anonymity, group signature can protect users' privacy and help to get rid of the threats that attackers set $s = H_2(H_1(\alpha||\beta)||r)$. Then attackers are not able to forge a credential to pass authentication in this way.

5 Conclusion

In this paper, we analyze the security of Tian *et al.*'s privacy-preserving remote user authentication with k-times untraceability scheme. We found that their scheme is insecure although it is proven to be secure. It can not satisfy user authenticity and k-time untraceability. This is to say, in their k-RUA scheme, any third party can forge a credential and impersonate an authorized user to pass authentication. Furthermore, the authentication server can not trace the real identity of a dishonest user, even though he authenticated for more than k times. Our analysis is confirmed thought two concrete attacks. In the last, we show the reason to produce such attacks and give the corresponding suggestions. Our future work is improving privacy-preserving remote user authentication with k-times untraceability scheme.

Acknowledgments

This research was was supported by Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201808, GCIS201710), the Engineering Program Project of CUC (3132015XNG1541), National Key *R&D* Program of China(2018YFB0803900) and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BDKFJJ012). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Annual International Cryptology Conference*, pp. 255–270, 2000.

[2] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *International Conference on Security and Cryptography for Networks*, pp. 111–125, 2006.

[3] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to win the clonewars: Efficient periodic n-Times anonymous authentication," in *Proceedings of the 13th ACM conference on Computer and Communications Security*, pp. 201–210, 2006.

[4] J. Camenisch, M. Kohlweiss, and C. Soriente, "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *International Workshop on Public Key Cryptography*, pp. 481–500, 2009.

[5] L. Chen, M. Enzmann, A. R. Sadeghi, M. Schneider, and M. Steiner, "A privacy-protecting coupon system," in *International Conference on Financial Cryptography and Data Security*, pp. 93–108, 2005.

[6] D. He, N. Kumar, J. Chen, C. C. Lee, N. Chilamkurti, and S. S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

[7] D. He, S. Zeadally, N. Kumar, and J. H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2590–2601, 2016.

[8] M. S. Hwang, C. C. Lee, Y. C. Lai, "An untraceable blind signature scheme", *IEICE Transactions on Foundations*, vol. E86-A, no. 7, pp. 1902–1906, July 2003.

[9] J. V. Jokinen, L. Blants, R. Pitkänen, S. Pienimäki, J. Mattila, and R. Suomela, *Real-Time Wireless E-Coupon (promotion) Definition based on Available Segment*, US20080120186A1, 2008.

[10] C. C. Lee, M. S. Hwang, and W. P. Yang, "A new blind signature based on the discrete logarithm problem for untraceability", *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 837–841, May 2005.

[11] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1-2, pp. 214–222, 2012.

[12] L. Nguyen, "Efficient dynamic k-times anonymous authentication," in *International Conference on Cryptology in Vietnam*, pp. 81–98, 2006.

[13] L. Nguyen and R. S. Naini, "Dynamic k-times anonymous authentication," in *International Conference on Applied Cryptography and Network Security*, pp. 318–333, 2005.

[14] J. Shao, X. Lin, R. Lu, and C. Zuo, "A threshold anonymous authentication protocol for vanets," *IEEE Transactions on vehicular technology*, vol. 65, no. 3, pp. 1711–1720, 2015.

[15] I. Teranishi, J. Furukawa, and K. Sako, "K-times anonymous authentication," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 308–322, 2004.

[16] I. Teranishi and K. Sako, "K-times anonymous authentication with a constant proving cost," in *International Workshop on Public Key Cryptography*, pp. 525–542, 2006.

[17] Y. Tian, Y. Li, B. Sengupta, R. H. Deng, A. Ching, and W. Liu, "Privacy-preserving remote user authentication with k-times untraceability," in *International Conference on Information Security and Cryptology*, pp. 647–657, 2018.

[18] Y. Yang, H. Cai, Z. Wei, H. Lu, and K. K. R. Choo, "Towards lightweight anonymous entity authentication for iot applications," in *Australasian Conference on Information Security and Privacy*, pp. 265–280, 2016.

[19] F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 533–547, 2002.

[20] H. Zhu, Y. Zhang, and X. Wang, "A novel one-time identity-password authenticated scheme based on biometrics for e-coupon system," *International Journal of Network Security*, vol. 18, no. 3, pp. 401–409, 2016.

Biography

Qijia Zhang received the B.E. degree from Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2018. He is currently working towards a M.E. degree, North China University of Technology, Beijing. His research interests include applied cryptography and information security.

Jianhong Zhang received his Ph.D. degrees in Cryptography from Xidian University, Xian, Shanxi, in 2004 and his M.S. degree in Computer Software from Guizhou University, Guiyang, Guizhou, in 2001. He was engaging in postdoctoral research at Peking University from October 2005 to December 2007. He now is a Professor of School of Electronic and Information Engineering, North China University of Technology, Beijing China. His research interests include computer networks, cryptography, electronic commerce security, computer software.

Linhan Liu is currently working towards a B.E. degree in North China University of Technology, Beijing. Her research interests are information security and computer networks.

Jing Wang is an employee of Beijing Jingdong Century Information Technology Company, Limited. His research interests include applied cryptography and information security.

Pei Liu is an employee of Beijing Jingdong Century Information Technology Company, Limited. His research interests include applied cryptography and information security.