Journal of Applied Mathematics and Physics, 2018, 6, 18-28
http://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Improvement of an Anonymous and Lightweight Authentication Scheme for TMIS

Chien-Ming Chen[1], Bin Xiang[1], Eric Wang Ke[1], Tsu-Yang Wu[2,3], Jerry Chun-Wei Lin[1]

[1]Department of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
[2]Fujian University of Technology Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, China
[3]National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou, China
Email: chienming@hit.edu.cn

How to cite this paper: Chen, C.-M., Xiang, B., Ke, E.W., Wu, T.-Y. and Lin, J.C.-W. (2018) Improvement of an Anonymous and Lightweight Authentication Scheme for TMIS. *Journal of Applied Mathematics and Physics*, **6**, 18-28.
https://doi.org/10.4236/jamp.2018.61003

Received: October 23, 2017
Accepted: January 2, 2018
Published: January 5, 2018

Abstract

Telecare Medicine Information Systems (TMIS) provides flexible and convenient healthcare for patients. However, the medical data transmitted between patients and doctors are exposed to unsecure public networks. To protect the patient's personal information, many authentication schemes are designed. Recently, Kang *et al.* proposed a hash based authentication schemes for TMIS and claimed that it could resist various attacks. However, we find that their proposed scheme is unsecure to traceability attack and user impersonation attack. In order to enhance the security and preserve the efficiency of Kang *et al.*'s, we proposed a new anonymous and lightweight scheme. The analysis demonstrates that our proposed scheme is superior to Kang *et al.*'s and the related schemes in security.

Keywords

Authentication, TMIS, Smart Card

1. Introduction

Advances in computer networks and communications boost the development of telecare medicine information systems (TMIS), through which patients can use flexible and convenient healthcare. A typical medical application scenario of TMIS is shown as **Figure 1**. Patients submit their healthcare data to a telecare server via wired or wireless medical devices in their home. When the doctors receive the patient's medical reports, they perform the diagnosis at their clinical center, and then provide the clinical decisions and treatments to the patients via the internet.

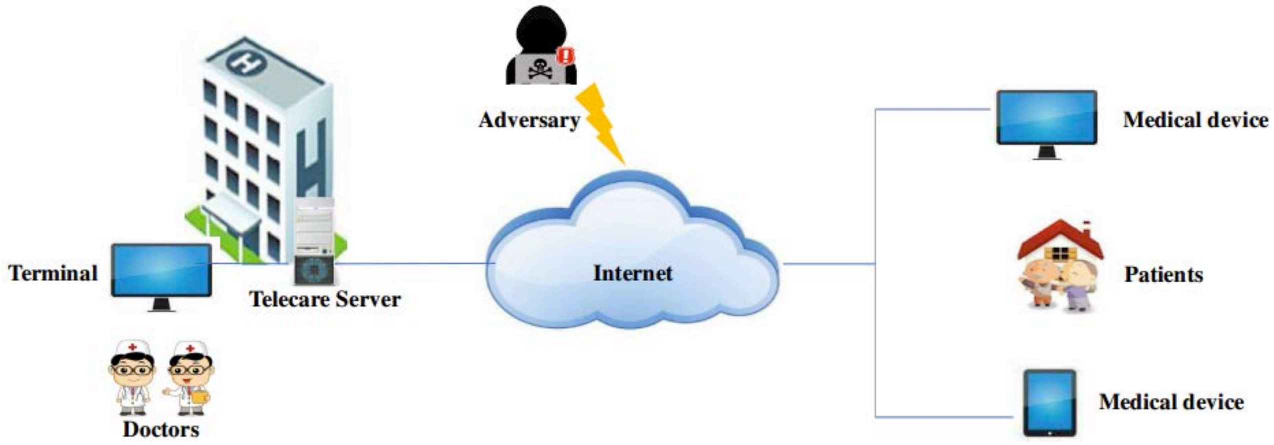

Figure 1. A typical medical application scenario of TMIS.

However, since the healthcare data transmitted through the public channel usually contains the secret information of the patients, it is essential to use authentication mechanism in the TMIS scenario. Recently, many authentication schemes for TMIS [1]-[6] have been designed to protect patient's personal information. In 2017, Kang *et al.* proposed a user authentication scheme for TMIS [1], which has simple computing cost due to the only usage of hash function. They claimed that the proposed scheme could resist various attacks. Unfortunately, we find that their scheme still suffers from traceability attack and user impersonation attack. To enhance the security and preserve the efficiency of Kang *et al.*'s scheme, we proposed a new anonymous and lightweight scheme.

2. Review of Kang *et al.*'s Scheme

This section presents Kang *et al.*'s scheme for TMIS [1]. Their scheme consists of four phases: 1) Registration Phase, 2) Login Phase, 3) Authentication Phase and 4) Password Change Phase. For convenience, some notations used in this paper are described as follows:

- U_i : Patient or user
- ID_i : The identity of the i^{th} user
- PW_i : The password of the i^{th} user
- X_s : The secret number of the TMIS server
- $E_K(\cdot)$: A symmetric encryption function with key K
- $D_K(\cdot)$: A symmetric decryption function with key K
- $h(\cdot)$: A one-way hash function
- $\oplus$: Exclusive-or operation
- $||$: Concatenation operation

2.1. Registration Phase

To access the facilities or services provided by the TMIS server, the user must register in the server first by the following steps.

Step 1. The user U_i chooses the identity and password ID_i, PW_i and generates a random integer *b*. Then, he computes $RPW_i = h(PW_i \| b)$ and sends the registration message $\{ID_i, RPW_i\}$ to the TMIS server via a secure channel.

Step 2. On receiving the registration message, the server generates a random integer N_s and then computes $\alpha_i = h(X_s \| N_s)$, $\beta_i = h(\alpha_i)$, $TPW_i = h(RPW_i \| ID_i) \oplus \alpha_i$, $A_i = \alpha_i \oplus \beta_i$. Next, the TMIS server issues a smart card and stores the data $\{TPW_i, h(\cdot), A_i, N_s\}$ and sends the card to the user U_i.

Step 3. When the user U_i receives the card from the server, he stores *b* into the card. Finally, the smart card contains $\{TPW_i, h(\cdot), A_i, N_s, b\}$.

2.2. Login Phase

When a registered user U_i desires to login in the TMIS, he must perform the following steps to construct a login request message.

Step 1. The user U_i inserts his smart card and enters the identity $ID_i^{'}$ and the password $PW_i^{'}$, which are chose by himself in the registration phase. Then, the smart card computes $\alpha_i^{'} = TPW_i \oplus h(h(PW_i^{'} \| b) \| ID_i^{'})$, $\beta_i^{'} = h(\alpha_i^{'})$, $A_i^{'} = \alpha_i^{'} \oplus \beta_i^{'}$ and checks whether $A_i^{'}$ equals to A_i or not. If true, the smart card believes the user is the owner and continues to execute *Step* 2. Otherwise, it terminates the login request.

Step 2. The smart card generates a random integer N_u and the current timestamp T_1, and then computes $L_1 = TPW_i \oplus h(\beta_i^{'} \| N_u)$, $L_2 = TPW_i \oplus \beta_i^{'}$, $L_3 = h(L_2 \| N_u \| \beta_i^{'} \| T_1)$. After that, it sends the login request message $\{TPW_i, L_1, L_3, N_u, N_s, T_1\}$ to the TMIS server via a public channel.

2.3. Authentication Phase

After receiving the login request message from the user, the TMIS server performs the following steps to achieve mutual authentication and establish a shared session key.

Step 1. The TMIS server retrieves the current timestamp $T_1^{'}$ and verifies the freshness of U_i's timestamp T_1.

Step 2. The server then continues to compute $\alpha_i = h(X_s \| N_s)$, $\beta_i = h(\alpha_i)$, $L_2^{'} = TPW_i \oplus \beta_i$, $L_3^{'} = h(L_2^{'} \| N_u \| \beta_i \| T_1)$ and checks whether $L_3^{'}$ equals to L_3. If the two values equal, then the user is authenticated and the authentication process continues.

Step 3. The server generates a random integer N_a and the current timestamp T_2, and computes $SK = h(TPW_i \| L_2^{'} \| N_u \| N_a \| \beta_i)$, $C_i = h(SK \| T_2)$. After that, it sends the authentication message $\{N_a, T_2, C_i\}$ to user U_i via a public channel.

Step 4. On receiving the authentication message form the server, the user U_i retrieves the current timestamp $T_2^{'}$ and verifies the freshness of server's timestamp T_2.

Step 5. The user computes the key $SK^{'} = h(TPW_i \| L_2 \| N_u \| N_a \| \beta_i^{'})$, $C_i^{'} = h(SK^{'} \| T_2)$ and checks whether $C_i^{'}$ is equal to C_i. If true, the server is authenticated. Otherwise, the authentication process is terminated.

Step 6. The user then generates the current timestamp T_3 and continues to compute $D_i = h\left(SK' \| \beta_i' \| T_3 \| N_u \| N_a\right)$. After that, he sends the response message $\{D_i, T_3\}$ to the server via a public channel.

Step 7. When the server receives the response message, it retrieves the current timestamp T_3' and verifies the freshness of T_3. Then it computes $D_i' = h\left(SK \| \beta_i \| T_3 \| N_u \| N_a\right)$ and checks whether D_i' equals to D_i. If true, the server believes that they have established the session key *SK*.

Finally, the user and the TMIS server can use the shard session key to encrypt the information transmitted through the public channel without worrying about the privacy disclosure.

2.4. Password Change Phase

This phase is needed when a user desires to change his password. For this, the user has to perform the following steps.

Step 1. The user U_i inserts his smart card and then he enters the identity ID_i' and the password PW_i' to pass the smart card verification. The smart card will compute $\alpha_i' = TPW_i \oplus h\left(h\left(PW_i' \| b\right) \| ID_i'\right)$, $\beta_i' = h\left(\alpha_i'\right)$, $A_i' = \alpha_i' \oplus \beta_i'$ and checks whether A_i' equals to A_i or not. If true, the smart card believes the user is the owner and continues to execute *Step* 2. Otherwise, it terminates the password change request.

Step 2. The user inputs the new password PW_i^{new} and the smart card computes $TPW_i^{new} = TPW_i \oplus h\left(h\left(PW_i' \| b\right) \| ID_i'\right) \oplus h\left(h\left(PW_i^{new} \| b\right) \| ID_i'\right)$.

Step 3. The smart card replaces TPW_i with the new value TPW_i^{new} in its memory.

3. Cryptanalysis of Kang *et al.*'s Scheme

In this section, we describe our findings that the scheme of Kang *et al.* is vulnerable to traceability attack and user impersonation attack. Before that, an attacker model [7] [8] is defined as follows.

3.1. Attacker Model

1) The adversary has full control of the public channel, but not the secure channel. That means the adversary can obtain all the transmitted data in the login and authentication phase.

2) The adversary can alter, delete or replay the data that he captured form the public channel.

3) The adversary has the ability to read or extract the secret data from the smart card issued to the user.

4) The adversary can guess either the user's identity or the password, but not both at a time.

5) The adversary knows the authentication scheme since he can be an outsider user or a legal user.

3.2. Suffer from Traceability Attack

The main mechanism of the traceability attack is that the adversary can trace the user (patient) with the messages captured from the public channel. This happens when there exist invariant parameters in the login or response message. Unfortunately, in the of Kang *et al.*'s scheme, we find that the login messages contain TPW_i, which are equal in all sessions. Through it, the adversary can trace the user.

3.3. Suffer from User Impersonation Attack

The main mechanism of the user impersonation attack is that the adversary can impersonate the user (patient) to construct the login and response message sent to the TMIS server and establish a session key with it without being found malicious. We assume that the adversary obtains the user's login message $\{TPW_i, L_1, L_3, N_u, N_s, T_1\}$ transmitted in the public channel, the he can perform the user impersonation attack by the following steps.

Step 1. The adversary registers himself in the TMIS server with the registration message $\{ID_e, RPW_e = h(t)\}$, where t is a simple integer. And obtains a smart card contains the data $\{TPW_e, h(\cdot), A_e, N_e, b\}$. Then, he use the return parameters to compute the legal values $\alpha_e = h(X_s || N_e) = TPW_e \oplus h(RPW_e || ID_e)$ and $\beta_e = h(\alpha_e)$.

Step 2. The adversary generates a random integer N_u and the current timestamp T_1, and then he constructs the required login parameters $L_1 = TPW_i \oplus h(\beta_e || N_u)$, $L_2 = TPW_i \oplus \beta_e$, $L_3 = h(L_2 || N_u || \beta_e || T_1)$, where TPW_i is obtained from the user's old login request message. After that, it sends the login request message $\{TPW_i, L_1, L_3, N_u, N_e, T_1\}$ to the TMIS server via a public channel, where N_e is obtained from the adversary's smart card.

Step 3. On receiving the login request message, the TMIS server first checks T_1 and $L_3^{'}$ by computing $\alpha_e = h(X_s || N_e)$, $\beta_e = h(\alpha_e)$, $L_2^{'} = TPW_i \oplus \beta_e$, $L_3^{'} = h(L_2^{'} || N_u || \beta_e || T_1)$. These will pass the verification and the server will take the adversary as the real user who has the parameter TPW_i in his smart card since there is no table to record the corresponding relations between TPW_i and the generated random integer N_s in the registration phase.

Step 4. The server then generates a random integer N_a and the current timestamp T_2, and computes $SK = h(TPW_i || L_2^{'} || N_e || N_a || \beta_e), C_i = h(SK || T_2)$. After that, it sends the authentication message $\{N_a, T_2, C_i\}$ to the adversary, who he thinks is the real user.

Step 5. When receives the authentication message from the server, the adversary computes $SK^{'} = h(TPW_i || L_2 || N_e || N_a || \beta_e)$. And then generates the current timestamp T_3 and computes $D_i = h(SK^{'} || \beta_e || T_3 || N_u || N_a)$, and sends the response message $\{D_i, T_3\}$ to the server via a public channel.

Step 6. When the server receives the response message, the server verifies the freshness of T_3 and checks the validity of $D_i^{'} = h(SK || \beta_i || T_3 || N_u || N_a)$ by comparing the values of $D_i^{'}$ and D_i. There is no doubt that it will pass the verification.

Finally, the adversary and the TMIS server establish a shared session key SK, with which the adversary can make requests for the private information such as medical records of the user (patient) without being detected.

4. The Proposed Scheme

In previous sections, we show that Kang *et al.*'s scheme fails to achieve the claimed goals since the user can be tracked through a constant quantity during the authentication process in the TMIS. To erase the mentioned security weaknesses, we present a new anonymous and lightweight authentication scheme for TMIS. The proposed scheme also consists of four phase: 1) Registration Phase, 2) Login Phase, 3) Authentication Phase and 4) Password Change Phase.

4.1. Registration Phase

When a user desires to use the facilities or services provided by the TMIS server, he must become the legal user first. For this, he needs to perform the following steps to register in the TMIS server.

Step 1. The user U_i chooses the identity and password ID_i, PW_i and generates a random integer *b*. Then, he computes $RPW_i = h(PW_i \| b)$ and sends the registration message $\{ID_i, RPW_i\}$ to the TMIS server via a secure channel.

Step 2. When the server receives the registration message, he generates a random integer N_i and computes $PID_i = E_{X_S}(ID_i \| N_i)$, $A_i = h(ID_i \| RPW_i)$, $B_i = h(ID_i \| X_s) \oplus h(ID_i) \oplus RPW_i$. Then, he issues a smart card and stores the data $\{PID_i, A_i, B_i, h(\cdot)\}$ into it. After that, he sends the card to the user U_i via a secure channel.

Step 3. On receiving the smart card from the server, the user U_i stores *b* into the card. Finally, the smart card contains $\{PID_i, A_i, B_i, b, h(\cdot)\}$.

4.2. Login Phase

A registered user U_i can construct a login request message to login in the TMIS by the following steps.

Step 1. The user U_i inserts his smart card and enters the identity $ID_i^{'}$ and the password $PW_i^{'}$. Then, the smart card computes $RPW_i^{'} = h(PW_i^{'} \| b)$, $A_i^{'} = h(ID_i^{'} \| RPW_i^{'})$ and checks whether $A_i^{'}$ equals to A_i or not. If true, the smart card believes the user is the owner and continues. Otherwise, it terminates the login request.

Step 2. The smart card generates a random integer r_u and the current timestamp T_1, and then computes $C_i = B_i \oplus h(ID_i^{'}) \oplus RPW_i^{'}$, $D_i = C_i \oplus r_u$, $E_i = h(ID_i^{'} \| D_i \| C_i \| T_1)$. After that, he sends the login request message $\{PID_i, D_i, E_i, T_1\}$ to the TMIS server via a public channel.

4.3. Authentication Phase

After receiving the login request message from the user, the TMIS server performs the following steps to build up a shared session key with the user.

Step 1. The TMIS server retrieves the current timestamp T_1' and verifies the freshness of U_i's timestamp T_1.

Step 2. The TMIS server then obtains $(ID_i \| N_i) = D_{X_s}(PID_i)$ and computes $C_i' = h(ID_i \| X_s)$, $E_i' = h(ID_i \| D_i \| C_i' \| T_1)$, and checks whether E_i' is equal to E_i or not. If true, the user is authenticated and the authentication process continued. Otherwise, the server aborts the authentication process.

Step 3. The server generates two random integers N_i^{new}, r_s and the current timestamp T_2, and then continues to compute $PID_i^{new} = E_{X_s}(ID_i \| N_i^{new})$, $r_u' = D_i \oplus C_i'$, $F_i = C_i' \oplus r_s$, $SK = h(ID_i \| r_u' \| r_s \| C_i')$, $H_i = h(ID_i \| F_i \| C_i' \| SK \| T_2)$. After that, he sends the authentication message $\{PID_i^{new}, F_i, H_i, T_2\}$ to the user U_i via a public channel.

Step 4. On receiving the authentication message form the server, the user U_i retrieves the current timestamp T_2' and verifies the freshness of server's timestamp T_2.

Step 5. The user then continues to compute $r_s' = F_i \oplus C_i$, $SK' = h(ID_i' \| r_u \| r_s' \| C_i)$, $H_i' = h(ID_i' \| F_i \| C_i \| SK' \| T_2)$ and checks whether H_i', H_i are equal or not. If true, the server is authenticated and PID_i^{new} is used to replace PID_i in the smart card's memory. Otherwise, the authentication process is terminated.

Step 6. The user generates the current timestamp T_3 and computes $M_i = h(SK' \| C_i \| T_3)$, and sends the response message $\{M_i, T_3\}$ to the server via a public channel.

Step 7. When the server receives the response message, it retrieves the current timestamp T_3' and verifies the freshness of T_3. Then it computes $M_i' = h(SK \| C_i' \| T_3)$ and checks whether M_i' equals to M_i. If the two values equal, the server believes that they have established the session key *SK*.

4.4. Password Change Phase

When a user desires to change his password, he can perform the following steps without any assistance from the TMIS server.

Step 1. The user U_i inserts his smart card and enters the identity ID_i' and the password PW_i'. Then, the smart card computes $RPW_i' = h(PW_i' \| b)$, $A_i' = h(ID_i' \| RPW_i')$ and checks whether A_i' equals to A_i or not. If true, the smart card believes the user is the owner and continues. Otherwise, it terminates the password change request.

Step 2. The user then inputs the new password PW_i^{new} and the smart card computes $A_i^{new} = h(ID_i' \| h(PW_i^{new} \| b))$, $B_i^{new} = B_i \oplus RPW_i' \oplus h(PW_i^{new} \| b)$.

Step 3. The smart card replaces A_i, B_i with A_i^{new}, B_i^{new} in its memory.

5. Security Analysis

Various authentication schemes have been demonstrated insecure [9] [10] [11] [12]. Thus, in this section we discuss the security features of the proposed scheme under the adversary model mentioned in the Section 3.

5.1. User Anonymity

Anonymity is a mechanism that there is no adversary having the capacity to compromise the user's (patient's) real identity. In the proposed scheme, the user's identity is masked in parameters $PID_i = E_{X_s}(ID_i \| N_i)$, $E_i = h\left(ID_i^{'} \| D_i \| C_i \| T_1\right)$, $PID_i^{new} = E_{X_s}\left(ID_i \| N_i^{new}\right)$ and $H_i = h\left(ID_i \| F_i \| C_i^{'} \| SK \| T_2\right)$. With the protection of the one-way hash function, the adversary has no way to retrieve the user's identity.

5.2. Mutual Authentication

In the proposed scheme, the user (patient) and the TMIS server achieve mutual authentication with the assistance of $E_i = h\left(ID_i^{'} \| D_i \| C_i \| T_1\right)$ and $H_i = h\left(ID_i \| F_i \| C_i^{'} \| SK \| T_2\right)$. That means, the user is authenticated by the server according to the value E_i since no one knows all the values needed to construct E_i besides the real user. Also, only the TMIS server can construct H_i for verification.

5.3. Session Key Security

In the proposed scheme, only the user (patient) and TMIS server can compute the shared session key $SK = h\left(ID_i \| r_u^{'} \| r_s \| C_i^{'}\right) = h\left(ID_i^{'} \| r_u \| r_s^{'} \| C_i\right)$ since all the values required to calculated the key is only known to the user and TMIS. With knowing the parameters transmitted in the authentication process, the adversary cannot construct the key.

5.4. Traceability Attack

In different sessions of the proposed scheme, the parameters in the user's (patient's) login request message $\{PID_i, D_i, E_i, T_1\}$ and response message $\{M_i, T_3\}$ are changeable. Thus, with the transmitted messages captured from different, the adversary cannot trace the user. Our proposed scheme resists traceability attack.

5.5. Replay Attack

When the adversary eavesdrops the whole transmitted message between the user (patient) and the TMIS server and replay it later, it will be immediately detected as the timestamp is outdate in the parameters $E_i = h\left(ID_i^{'} \| D_i \| C_i \| T_1\right)$ and $H_i = h\left(ID_i \| F_i \| C_i^{'} \| SK \| T_2\right)$. Then the adversary may try to reconstruct the two parameters. However, with the protection of the parameters ID_i ($ID_i^{'}$) and C_i ($C_i^{'}$), the adversary cannot realize it. Thus, our proposed scheme resists replay attack.

5.6. Offline Password Guessing Attack

In the proposed scheme, the adversary can only guess the user's (patient's) password through $A_i = h(ID_i \| RPW_i)$ or $B_i = h(ID_i \| X_s) \oplus h(ID_i) \oplus RPW_i$

in the smart card. However, without the knowledge of the user's real identity ID_i, the adversary cannot compute the values required.

5.7. Impersonation Attack

When the adversary desires to impersonate the user (patient), he needs to construct the login request message first. However, the adversary is no way to know the user's identity ID_i and C_i, which are required to construct the parameter $E_i = h\left(ID_i^{'} \| D_i \| C_i \| T_1\right)$ in the login request message. Thus, he cannot impersonate the user. On the other hand, without knowing the server's secret key X_s, the adversary cannot decrypt PID_i and computes the parameter $C_i^{'} = h\left(ID_i \| X_s\right)$ and $H_i = h\left(ID_i \| F_i \| C_i^{'} \| SK \| T_2\right)$. Thus, he cannot impersonate the server. Our proposed scheme resists impersonation attack.

5.8. Stolen Verifier Attack

The stolen verifier attack means that the adversary gets some precious information that is stored in the server's end. This happens especially when the server maintains the database of the user's information like password. In the proposed scheme, the TMIS server does not keep any storage database, which is an essential requirement to launch this attack. Thus, our proposed scheme resists stolen verifier attack.

6. Performance Analysis

In this section, we compare the communication cost and compare the security features with the related schemes [1]-[6]. We only compare the time cost in the login and authentication phase of the proposed scheme and Kang *et al.*'s since the two phases are performed frequently. During the login and authentication phase of our proposed scheme, hash function is used 12 times. However, Kang *et al.*'s uses 14 times. Next, we compare the security features with Kang *et al.*'s and the other related schemes. From the **Table 1**, we can see that our proposed

Table 1. Comparison of security features. (O: Satisfy X: Not Satisfy).

Schemes	Security features									
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
[2]	O	X	X	O	O	O	O	X	X	O
[3]	O	X	X	X	O	O	O	O	X	O
[4]	O	X	X	X	O	O	O	O	X	O
[5]	X	X	X	X	X	O	O	O	X	O
[6]	X	X	X	X	X	O	O	O	O	O
[1]	O	O	X	O	X	O	O	O	O	O
Our's	O	O	O	O	O	O	O	O	O	O

F1. Insider Attack; F2. User Anonymity; F3. Traceability Attack; F4. Offline Password Guessing Attack; F5. User Impersonation Attack; F6. Replay Attack; F7. No Verification Table; F8. Session Key Agreement; F9. Detect Wrong Password Quickly; F10. Mutual Authentication.

scheme performs better in terms of providing security features.

7. Conclusion

In this paper, we analyze Kang *et al.*'s scheme which was designed for TMIS using hash function and claimed to resist various attacks. However, we still find that the scheme is susceptible to traceability attack and user impersonation attack. In order to erase the secure drawbacks we found, we present a new anonymous and lightweight scheme and prove that our proposed scheme has better performance in terms of communication cost and security.

Acknowledgements

The work of Chien-Ming Chen was supported in part by the Project NSFC (National Natural Science Foundation of China) under Grant number 61402135 and in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755. The work of Eric Ke Wang was supported in part by National Natural Science Foundation of China (No. 61572157), grant No. 2016A030313660 from Guangdong Province Natural Science Foundation, JCYJ20160608161351559 from Shenzhen Municipal Science and Technology Innovation Project.

References

[1] Kang, D., Lee, D., Cho, S., Jung, J. and Won, D. (2017) Cryptanalysis and Improvement of Robust Authentication Scheme for Telecare Medicine Information Systems. *In Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, ACM, 18.

[2] Zhu, Z. (2012) An Efficient Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, **36**, 3833-3838. https://doi.org/10.1007/s10916-012-9856-9

[3] Wei, J., Hu, X. and Liu, W. (2012) An Improved Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, **36**, 3597-3604. https://doi.org/10.1007/s10916-012-9835-1

[4] Debiao, H., Jianhua, C. and Rui, Z. (2012) A More Secure Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, **36**, 1989-1995. https://doi.org/10.1007/s10916-011-9658-5

[5] Wu, Z.Y., Lee, Y.C., Lai, F., Lee, H.C. and Chung, Y. (2012) A Secure Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, **36**, 1529-1535. https://doi.org/10.1007/s10916-010-9614-9

[6] Arya, K. and Vidwansh, A. (2015) A Robust Authentication Scheme for Telecare Medicine Information Systems. *International Journal of Computer Applications*, **123**, No. 6. https://doi.org/10.5120/ijca2015905341

[7] Kocher, P., Jaffe, J. and Jun, B. (1999) Differential Power Analysis. In: *Advances in cryptology—CRYPTO'*99, Springer, Berlin, 789-789. https://doi.org/10.1007/3-540-48405-1_25

[8] Messerges, T.S., Dabbish, E.A. and Sloan, R.H. (2002) Examining Smart-Card Secu-

rity under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, **51**, 541-552. https://doi.org/10.1109/TC.2002.1004593

[9] Chen, C.M., Li, C.T., Liu, S., Wu, T.Y. and Pan, J.S. (2017) A Provable Secure Private Data Delegation Scheme for Mountaineering Events in Emergency System. *IEEE Access*, **5**, 3410-3422. https://doi.org/10.1109/ACCESS.2017.2675163

[10] Sun, H.M., He, B.Z., Chen, C.M., Wu, T.Y., Lin, C.H. and Wang, H. (2015) A Provable Authenticated Group Key Agreement Protocol for Mobile Environment. *Information Sciences*, **321**, 224-237. https://doi.org/10.1016/j.ins.2015.01.037

[11] Chen, C.M., Fang, W., Wang, K.H. and Wu, T.Y. (2017) Comments on "An Improved Secure and Efficient Password and Chaos-Based Two-Party Key Agreement Protocol". *Nonlinear Dynamics*, **87**, 2073-2075.
https://doi.org/10.1007/s11071-016-3171-9

[12] Chen, C.M., Xu, L., Wu, T.Y. and Li, C.R. (2016) On the Security of a Chaotic Maps-Based Three-Party Authenticated Key Agreement Protocol. *Journal of Network Intelligence*, No. 2, 61-65.