Attacks and Solutions on the Quan et al.’s Smart Card Based Remote User Authentication with a Key Agreement Scheme

Chien-ming Chen¹, Yanyu Huang¹, Xiaoting Deng¹, Bin Xiang¹ and Tsu-yang Wu²,³

¹Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China
chienming@hit.edu.cn
²Fujian Provincial Key Laboratory of Big Data Mining and Applications Fujian University of Technology Fuzhou 350118, China
³National Demonstration Center for Experimental Electronic Information and Electrical Technology Education Fujian University of Technology Fuzhou 350118, China

Abstract. Recently, Quan et al. proposed a smart card based user authentication scheme to conquer the existing defects in An’s scheme. In their paper, they claimed that their proposed scheme could resist various attacks. However, by analysing their scheme, we demonstrate that it still suffers from a traceable attack and an off-line password guessing attack. To erase the weaknesses and enhance the security, we then suggest some modification in their scheme. Compared with Quan et al.’s and other related schemes, our modified one performs better in terms of security.

1. Introduction
Due to the development of computer networks and communications, various services based on the internet have sprung up, such as online shopping, remote monitoring, healthcare, etc. These services provide people a happy and high quality life. However, the data transmitted in these services usually contain people’s sensitive and private information, and thus it is essential to use some cryptographic mechanisms to guarantee the security. Remote user authentication with a key agreement is a cryptographic mechanism that not only realizes a mutual authentication between a user and server, but also establishes a session key for further communication. It ensures the authenticity and legitimacy of users and servers. Many remote user authentications with a key agreement schemes had been proposed in the past few years [1-10].

In 2015, a proposed enhanced smart card based remote user authentication scheme and claimed that it could resist various kinds of attacks [3]. However, Quan et al. later in their paper [4] pointed out that An’s scheme suffers from an off-line password guessing attack, a user impersonation attack. Besides, the scheme also fails to preserve user anonymity and lacks password change. To guarantee the reliability and security, they then proposed a new authentication scheme for remote user authentication environment [4]. Unfortunately, in this paper; we found that Quan et al.’s scheme is still vulnerable to a traceable attack and an off-line password guessing attack. In order to enhance the security and erase the weaknesses of Quan et al.’s scheme, we suggest a modification in their scheme.

The remainder of the paper is organized as follows. Section 2 briefly reviews the scheme of Quan et al.’s. We analyse their scheme and show the weaknesses in section 3. In section 4, we propose some modification in their scheme. In section 5, we discuss the performance among the modified one and some related schemes. Finally, we conclude our study in section 6.
2. Review of Quan et al.’s scheme
This section presents the scheme of Quan et al.’s, which contains five phases: (1) Initialization phase; (2) Registration phase; (3) Login phase; (4) Authentication phase; and (5) Password change phase.

2.1. Initialization Phase
In this phase, $S$ initializes some parameters. The details are as follow. $S$ selects a large prime $q$ and computes $p = 2q + 1$; $S$ selects a secret key $x \in \mathbb{Z}_q^*$; $S$ selects a secure one-way hash function $h(\cdot)$.

2.2. Registration Phase
Before enjoying the services provided by the remote server $S$, a user $U_i$ must register in $S$ first. The steps are as follows.

Step 1. $U_i$ selects his identity $ID$ and password $PW_i$, and inputs his biometric $BIO_i$. Then, he computes $Gen(BIO_i) = (R, P)$ and sends the registration message $\{ID, h(PW_i \oplus K), h(R \oplus K)\}$ to $S$.

Step 2. When receiving registration message, $S$ computes $A_i = h(PW_i \oplus K) \oplus h(R \oplus K) \mod p$. $B_i = h(ID_i \parallel x)^{h(PW_i \oplus K)} \mod p$. $N_i = h(ID_i \parallel x) \oplus h(x)$, $H_i = h(ID_i \parallel x) \oplus h(ID_i \parallel h(PW_i \oplus K))$ and stores the data $\{A_i, B_i, N_i, H_i, h(\cdot), p, q\}$ into a smart card $SC_i$. Finally, he sends the card to $U_i$.

Step 3. After receiving the smart card, $U_i$ stores values $\{K, P\}$ into it.

2.3. Login Phase
When a registered user $U_i$ desires to access the services, he needs to construct the login message with the following steps.

Step 1. $U_i$ inserts his smart card and inputs his $ID_i, PW_i, BIO_i$. Then, the smart card recovers $R'$ from $R' = Rep(BIO_i, P)$.

Step 2. $SC_i$ computes $A'_i = h(PW_i \oplus K) \oplus h(R' \oplus K) \mod p$ and then checks whether $A'_i = A_i$. If it holds, the smart card then generates a random integer $x \in \mathbb{Z}_q^*$, and obtains the value $h(ID_i \parallel x)$ by computing $H_i \oplus h(ID_i || h(PW_i \oplus K))$. After that, $SC_i$ computes the parameters $C_i = B_i / h(ID_i \parallel x)^{h(PW_i \oplus K)} \mod p$, $D_i = h(ID_i \parallel x)^{x^p} \mod p$ and $M_i = h(C_i \parallel D_i \parallel T_i)$. $SC_i$ computes $C_i$ as follows:

\[
C_i = B_i / h(ID_i \parallel x)^{h(PW_i \oplus K)} \mod p
\]

Step 3. Finally, it sends the login request message $\{D_i, M_i, N_i, T_i\}$ to $S$.

2.4. Authentication Phase
The user $U_i$ and the server $S$ authenticate each other and establish a session key by the following steps.

Step 1. On receiving login request message from $U_i$, $S$ checks whether $(T'_i - T_i) \leq AT_i$, where $T_i$ is the current timestamp of $S$. If it is valid, $S$ continues to execute Step 2; otherwise, he terminates the authentication process at once.

Step 2. $S$ obtains $h(ID_i \parallel x)$ by computing $N_i \oplus h(x)$, then he computes $C_i' = h(ID_i \parallel x^p) \mod p$ and $M_i' = h(C_i' \parallel D_i \parallel T_i)$. Next, $S$ checks whether $M_i' = M_i$ or not. If this holds, the user is authenticated and the authentication process continues; otherwise, $S$ terminates the process.

Step 3. $S$ subsequently generates a random number $\beta \in \mathbb{Z}_q^*$ and the current timestamp $T_i$. Then he computes $V_i = h(ID_i \parallel x)^{\beta} \mod p$, the session key $sk = D_i^\beta$ and $M_i = (C_i' \parallel V_i \parallel sk \parallel T_i)$. Finally, he sends the authentication message $\{V_i, M_i, T_i\}$ to $U_i$.

Step 4. When $U_i$ receives the authentication message $\{V_i, M_i, T_i\}$ from $S$, it first verifies the freshness of $T_i$. If this holds, it computes $sk'^* = V_i^\alpha \mod p$, $M_i'^* = (C_i' \parallel V_i \parallel sk'^* \parallel T_i)$ and checks whether $M_i'^* = M_i'$. If this holds, the server is authenticated and $U_i$ accepts $sk'^*$ and uses it for further communication.
2.5. Password Change Phase

This phase is performed when a registered user \( U \), desires to change his password. The whole process is performed by the following steps with no help from the remote server.

**Step 1.** \( U \) inserts his smart card and inputs \( ID_i^r, PW_i^r \) and \( BIO^r \). The smart card recovers \( R^r \) by computing \( \text{Rep}(BIO^r, P) \).

**Step 2.** \( SC_i \) computes \( A_i^r = h(PW_i^r \oplus K) \oplus (R^r \oplus K) \mod p \) and validates whether \( A_i^r = A_i \). If this holds, the smart card believes \( U \) is the owner. After that, \( U \) can input a new password \( PW_i^{NEW} \).

**Step 3.** \( SC_i \) obtains \( h(ID_i \parallel x) \) from computing \( H_i \oplus h(ID_i \parallel h(PW_i \oplus K)) \). And then, it computes \( A_i^{NEW} = h(PW_i^{NEW} \oplus K) \oplus (R \oplus K) \mod p, \quad B_i^{NEW} = B \cdot h(ID_i \parallel x)^{1/hh(PW_i^{NEW} \oplus K)} / h(ID_i \parallel x)^{1/hh(PW_i \oplus K)} \mod p \), and \( H_i^{NEW} = H_i \oplus h(ID_i \parallel h(PW_i \oplus K)) \oplus h(ID_i \parallel h(PW_i^{NEW} \oplus K)) \).

**Step 4.** \( SC_i \) uses \( \{ A_i^{NEW}, B_i^{NEW}, H_i^{NEW} \} \) to replace \( \{ A_i, B_i, H_i \} \) in its memory.

3. Weaknesses of Quan et al.’s scheme

In this section, we will describe our findings that Quan et al.’s scheme suffers from a traceable attack and an off-line password guessing attack.

3.1. Suffers from a Traceable Attack

The traceable attack means that with the messages transmitted over communication channel, an attacker \( Eve \) can trace the user. In Quan et al.’s scheme, the login message \( \{ D_i, M_i, N_i, T_i \} \) contains the constant parameter \( N_i \). With it, \( Eve \) can easily trace the user. Therefore, Quan et al.’s scheme suffers from a traceable attack.

3.2. Suffers from an Off-Line Password Guessing Attack

In Quan et al.’s paper, they claimed that even an attacker \( Eve \) has extracted the data stored in the smart card and obtained the transmitted messages between the user and the server, he still cannot launch an off-line password guessing attack. However, we find that it is not the case. \( Eve \) can carry out an off-line password guessing attack successfully with the following steps.

**Step 1.** \( Eve \) extracts \( \{ A_i, B_i, N_i, H_i, K \} \) in \( U_i \)'s smart card and obtains the login message \( \{ D_i, M_i, N_i, T_i \} \) in the communication channel.

**Step 2.** \( Eve \) registers in the remoter server and obtains a smart card contains \( \{ A_i, B_i, N_i, H_i, h() \parallel p, q \} \). He then computes \( h(x) = N_i \oplus h(ID_i \parallel h(PW_i \oplus K)) \).

**Step 3.** \( Eve \) guesses a possible password \( PW_i^r \) and computes \( h(ID_i \parallel x) = N_i \oplus h(x) \), where \( N_i \) is extracted from the card and \( h(x) \) is obtained in step 2.

**Step 4.** \( Eve \) computes \( C_i^r = B_i / h(ID_i \parallel x)^{h(PW_i^r \oplus K)} \) and \( M_i^r = h(C_i^r \parallel D_i \parallel T_i) \). Then, he checks whether \( M_i^r = M_i \) or not. If the two values equal, return \( PW_i^r \); otherwise, repeats step 3.

With the above steps, \( Eve \) can successfully guessing the password of \( U_i \). Quan et al.’s scheme suffers from an offline password guessing attack.

4. Possible Improvement

To erase the weaknesses of Quan et al.’s scheme, in this section, we suggest some simple but effective modification in login phase and authentication phase.

4.1. The Improved Login Phase

**Step 1.** \( U \) inserts his smart card and inputs his \( ID_i^r, PW_i^r, BIO^r \). Then, the smart card recovers \( R^r \) from \( R^r = \text{Rep}(BIO^r, P) \).

**Step 2.** \( SC_i \) computes \( A_i^r = h(PW_i^r \oplus K) \oplus h(R^r \oplus K) \mod p \) and then checks whether \( A_i^r = A_i \). If it holds, the smart card then generates a random integer \( \alpha \in Z_q^* \), and obtains the value \( h(ID_i \parallel x) \) by
computing $H_i \oplus h(ID_i \| h(PW_i \oplus K))$. After that, $SC_i$ computes the parameters $C_i = B_i \cdot h(ID_i \| x) \mod p$, $D_i = h(ID_i \| x)^{\alpha} \mod p$, $G_i = N_i \oplus C_i^\alpha$, and $M_i = h(G_i \| D_i \| T_i)$.

**Step 3.** Finally, it sends the login request message $\{D_i, M_i, G_i, T_i\}$ to $S$.

**Remark:** Note that besides the parameters $D_i, M_i, N_i$, $U_i$ computes one more parameter $G_i$ and the parameter $M_i$ is computed in a different way. Finally, the transmitted login messages are replaced by the new one $\{D_i, M_i, G_i, T_i\}$.

### 4.2. The Improved Authentication Phase

**Step 1.** On receiving login request message from $U_i$, $S$ checks whether $(T_i - T'_i) \leq \Delta T$, where $T'_i$ is the current timestamp of $S$. If it is valid, $S$ continues to execute **Step 2**; otherwise, he terminates the authentication process at once.

**Step 2.** $S$ obtains $h(ID_i \| x)$ by computing $G_i \oplus D'_i \oplus h(x)$, then he computes $C_i = h(ID_i \| x)^{\alpha} \mod p$ and $G'_i = D'_i \oplus M'_i = h(G'_i \| D_i \| T_i)$. Next, $S$ checks whether $M'_i = M_i$ or not. If this holds, the user is authenticated and the authentication process continues; otherwise, $S$ terminates the process.

**Step 3.** $S$ subsequently generates a random number $\beta \in Z_n^*$ and the current timestamp $T'_i$. Then he computes $V_i = h(ID_i \| x)^{\beta} \mod p$, the session key $sk = D'_i$ and $M'_i = (G'_i \| V_i \| sk \| T'_i)$. Finally, he sends the authentication message $\{V_i, M'_i, T'_i\}$ to $U_i$.

**Step 4.** When $U_i$ receives the authentication message $\{V_i, M'_i, T'_i\}$ from $S$, it first verifies the freshness of $T'_i$. If this holds, then he computes $sk' = V_i^{\alpha} \mod p$, $M'_i = (C_i \| V_i \| sk' \| T'_i)$ and checks whether $M'_i = M_i$. If this holds, the server is authenticated and $U_i$ accepts $sk'$ as the session key and uses it for further communication.

**Remark:** Note that $S$ obtains $h(ID_i \| x)$ by computing $G_i \oplus D'_i \oplus h(x)$ rather than $N_i \oplus h(x)$, and the computation of parameters $M_i, M'_i$ are changed.

### 5. Discussion

In this section, we demonstrate that our improvement in Quan et al.’s scheme enhance the security since the improved scheme can resist a traceable attack and an off-line password guessing attack. We further compare the improved scheme with Quan et al.’s and some related schemes in Table 1. The features we used to compare are **A1:** user anonymity; **A2:** resist offline password guessing attack; **A3:** resist user impersonation attack; **A4:** resist server masquerading attack; **A5:** resist insider attack; **A6:** resist replay attack; **A7:** resist traceable attack. When the scheme has the security feature, we use **Y** to mark; otherwise, we use **N**.

| SF | Sood et al. [1] | Chen et al. [2] | An [3] | Quan et al. [4] | Our improved |
|----|-----------------|-----------------|--------|-----------------|--------------|
| A1 | N               | N               | N      | Y               | Y            |
| A2 | N               | N               | N      | N               | Y            |
| A3 | N               | N               | N      | Y               | Y            |
| A4 | N               | N               | Y      | Y               | Y            |
| A5 | N               | N               | Y      | Y               | Y            |
| A6 | Y               | Y               | Y      | Y               | Y            |
| A7 | N               | N               | N      | N               | Y            |

**The improved scheme resists a traceable attack.** From the analysis in section 3, we realize that when the parameters in login messages are different in each conversation, an attacker cannot trace the user. In the improved scheme, the parameters in login messages are $D_i, M_i, G_i$ and $T_i$. The parameters $D_i, M_i, G_i$ are all protected by the once random integer $\alpha$ and thus are different every conversation.
The parameter $T$ is the current timestamp and thus is different every time as well. Therefore, the improved scheme resists a traceable attack.

**The improved scheme resists an offline password guessing attack.** From the analysis above, we know that Quan et al.’s scheme suffers from an off-line password guessing attack because with the extracted data in the smart and the login message, an attacker Eve can guess the possible password and use another parameter to validate. In the improved scheme, Eve can guess a possible password but when he intends to validate the correctness he face the hardness of computing $G_i$ and $M_i$, both of which are protected by the once random integer $a$ and Eve cannot obtain the value. Therefore, the improved scheme resists an offline password guessing attack.

6. Conclusion

In this paper, we analyse a smart card based remote user authentication with a key agreement scheme. Although the authors claimed that their scheme is secure against various attack, we still find it suffers from a traceable attack and an offline password guessing attack. To erase the weaknesses and enhance the security, we later suggest some simple but effective modification in their scheme. The discussion shows that the improved scheme performs better in terms of security.

7. Acknowledgement

The work was supported in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755.

8. References

[1] Sood SK, Sandeep K, Sarje A K, and Singh K 2010 An improvement of Wang et al.’s authentication scheme using smart cards *Proc. National Conf. on Communications (NCC)*. IEEE (Chennai: IEEE) pp 1–5

[2] Chen B L, Wen C K and Wuu L C 2014 Robust smart card based remote user password authentication scheme *Int. J. Commun. Syst.* 27 377–89

[3] An Y H 2015 Security enhancements of smart card-based remote user password authentication scheme with session key agreement *Proc. Int. Conf. on Advanced Communication Technology (ICACT)*. IEEE (Seoul: IEEE) pp 669–674

[4] Quan C Y, Jung J W, Sun Q M, Lee D H and Won D H 2017 Cryptanalysis and improvement of a biometric and smart card based remote user authentication scheme *Proc. Int. Conf. on Ubiquitous Information Management and Communication*. ACM (New York: ACM) p 50

[5] Kim J Y, Lee D H, Jeon W R, Lee Y S and Won D H 2014 Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks *Sensors* 14 6443–62

[6] Wang K H, Chen C M, Fang W C and Wu T Y 2018 on the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags *J. Supercomput.* 74 65 – 70

[7] Wang K H, Chen C M, Fang W C and Wu T Y 2017 A Secure Authentication Scheme for Internet of Things *Pervasive. Mob. Comput.* 42 15 – 26

[8] Chen C M, Xu L L, Wu T Y and Li C R 2016 On the Security of a Chaotic Maps-based Three-Party Authenticated Key Agreement Protocol *J. Net. Intel.* 1 61 – 65

[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 – 22

[10] Sun H M, He B Z, Chen C M, Wu T Y, Lin C H and Wang H X 2015 A Provable Authenticated Group Key Agreement Protocol for Mobile Environment *Inform. Sciences* 321 224–37