Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090304

A biometric-based Password Authentication with key Exchange Scheme using Mobile Device for Multi-Server Environment

*Xuelei Li**, *Qiaoyan Wen, Wenmin Li, Hua Zhang and Zhengping Jin*

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Received: 5 Jul. 2014, Revised: 6 Oct. 2014, Accepted: 7 Oct. 2014
Published online: 1 May 2015

Abstract: Remote authentication for multi-server environment can help users register only once and access arbitrary services conveniently in the same registry realm. However, most of the solutions are plagued by security problems. In this paper, we point out that 'a novel smart card and dynamic ID based remote user authentication scheme for multi-server environment' is vulnerable to user impersonation attack, server masquerade attack and cannot achieve forward secrecy. Therefore, by introducing biometrics as the third authentication factor, we devise an enhanced three-factor based remote authentication with key agreement scheme for multi-server environment. In our designation, we combine the technologies of *Client Puzzle*, *Fuzzy Extractor*, message authentication code (*MAC*) and *Diffie-Hellman key exchange*. Moreover, our proposal not only maintains the advantages of the original, but also preserves user privacy with optional access mode. Meanwhile, it can be also reduced to two-factor based scheme with less security properties for specific applications. Finally, the proposed scheme is proved to work correctly through BAN-Logic, and the security analysis and performance cost are discussed to show that our proposal is more secure, robust and practical.

Keywords: authentication, biometrics, mobile device, password, smart card

1 Introduction

1.1 Background and Problems

With the fast development of network technologies and communication systems, more and more information services can be obtained anytime and anywhere using advanced mobile devices (e.g., mobile phones and handheld computers) through wireless networks. Various applications, such as E-government and E-commerce, are carried out and provided around the world as information services. Multi-server architecture [3,4,5,6,7,8,9,10] as a new communication model appears to solve the tedious and duplicate registrations in single-server environment for convenience and privacy. There are three types of participants, including registration center, service servers and users, involved in the multi-server communication systems. Registration center administrates all the service servers and users, who have registered in its trust domain. Users, who have registered in the registration center only once, can obtain their desired services from different service servers under the help of the same registration center.

However, security is the main concern for users and service providers, because it guarantees the confidentiality, integrity and authentication of private communications over public channels. Foremost, as the first line of defense, authentication plays an important role to ensure the security of the information and communication system. It can ensure legal users obtaining their authorized services and forbid the illegal users accessing the information systems. In addition, it also provides the approval for accounting, authorization and auditing as the fundamental mechanism in access control. As a consequence, ideal remote authentication schemes are urgent needed, nowadays, to secure information and communication systems.

The motivation of this paper is to investigate remote user authentication schemes for multi-server architecture based on three factors (password, mobile device and

* Corresponding author e-mail: will8898@163.com

biometrics). Traditional two-factor (i.e., password and smart card) authentication schemes cannot satisfy the security requirements under the assumption that the property of tamper-resistance for smart cards can be breached by some approaches. In other words, the credential used for authentication can be compromised within the smart card, and such situation threats the security of authentication scheme in its application. Therefore, we introduce the third authentication factor to improve the information assurance in distributed systems.

1.2 Related Works

Password-based remote user authentication schemes in single server environment have been investigated for decades, since Lamport [1] developed the password authentication with insecure communication in 1981. However, password is chosen from a small space and vulnerable to guessing attacks. In addition, password verification table stored in the database of remote server is also susceptible to compromise attack by the administrators or intruders. Accordingly, smart card as the second factor appeared in authentication schemes [20,19] to enhance the security of password-based authentication schemes. It is generally assumed that smart card is a tamper-resistant device, i.e., smart card can protect the sensitive data stored in its memory from being tampered or compromised. Unfortunately, Kocher et al. [15], Messerges et al. [16] and Leng [17] pointed out that the above assumption may be problematic. In fact, the adversary can extract the data from the smart card by monitoring the power consumption or analyzing the leaked information, and further launch smart card breach attacks in authentication schemes. Consequently, a lot of smart card based authentication schemes [25,21,22,24] were broken down without the assumption of tamper-resistance. Later on, the third authentication factor, biometrics, was introduced for constructing secure authentication schemes [23,18,27,26]. Specifically, in 2011, Huang et al. [18] presented a generic framework for three-factor authentication scheme, which preserves security and privacy in distributed systems.

Nevertheless, the above conventional schemes in single server environment cannot be applied to multi-server environment directly for the concerns of security and practicability, due to the particular features of multi-server architecture [3,4,5,6,7,8,9,10]. In 2009, Liao and Wang [11] proposed a secure dynamic ID based remote user authentication scheme for multi-server environment. It can achieve user's anonymity to avoid being traced and identified user's request by static ID. They claimed that their proposal satisfied all the requirements for multi-server environment, including single registration, low computation, no verification table, password update freely, mutual authentication with key agreement and security. Later on, Hsiang et al. [12] pointed out that Liao and Wang's scheme is defenseless against several attacks and not reparable. In addition, Hsiang et al. presented an efficient improvement over Liao and Wang's scheme with more security. In 2011, Lee at al. [13] observed that Hsiang et al.'s improved scheme is still open to masquerade attack and server spoofing attack. Meanwhile, Lee et al. also proposed an enhancement to conquer the weaknesses in Hsiang et al.'s scheme. Recently, Li et al. [14] analyzed Lee et al.'s improvement and pointed out that forgery attack and server spoofing attack are effective on their scheme, and their scheme could not provide proper authentication if mutual authentication message is partly modified. In order to remove these weaknesses, Li et al. proposed a novel smart card and dynamic ID based remote user authentication scheme for multi-server environment. They also demonstrated that their scheme could satisfy all the essential requirements for multi-server environment. After that, several other proposals are presented in the literature [38,33,34,32,37,35] for dynamic ID authentication. However, there still exist some minor weaknesses in terms of security and efficiency. Furthermore, the methods of dynamic ID-based authentication in [36,39] provide advanced approaches to achieve user-privacy-preserving.

1.3 Contributions

In this paper, we find out Li et al.'s scheme [14] is vulnerable to user impersonation attack, server masquerade attack, and cannot achieve forward security. In order to overcome the weaknesses, we present a three-factor authentication scheme with key agreement for multi-server environment. We also prove the validity of our scheme through BAN-Logic. In addition, our designation can be reduced to two-factor (password and mobile device) authentication scheme if it is necessary, and it supports both anonymous and real-identity access mode. Finally, the security and performance analysis are presented to show that our scheme is more secure, robust and practical through comparing with related proposals.

1.4 Organizations

This paper is organized as follows. In section 2, the scheme in [14] is reviewed in brief and the security analysis of their scheme is presented in section 3. Some preliminaries of related technologies are introduced in section 4. Then, we present our enhanced scheme in section 5 and prove its validity in section 6. In addition, we also analyze our proposal in section 7. Finally, the conclusion is given in section 8. The abbreviations and notations used in this paper are listed in Table 1.

Table 1: Notations Used in This Paper

symbols	notions
U_i	ith user
S_j	jth service server
RC	registration center
ID_i	identity of U_i
PW_i	password of U_i
w_i	biometrics information of U_i
$Nonce$	random number used only once
SID_j	identity of S_j
CID_i	dynamic identity of U_i
x	master secret key of the registration center
y	secret number of the registration center
SK	session key shared between U_i and S_j
$h(\cdot)$	collision-resistant one way hash function
$\oplus$	exclusive OR operation
$\|\|$	concatenation
$\longrightarrow$	public channels
$\Longrightarrow$	secret channels
Adv	adversary

2 Review of Li et al.'s scheme

In this section, we briefly review the scheme in [14], which consists of four phases: registration, login, verification and password change phase. The detailed steps are described as follows and illustrated in Fig 1.

2.1 Registration phase

RC chooses the master secret key x and a secret number y to compute $h(x||y), h(SID_j||h(y))$, and shares them with S_j via a secure channel. U_i and RC perform the following steps to finish registration phase.

Step R1. U_i freely chooses ID_i and PW_i, and computes $A_i = h(b \oplus PW_i)$, where b is a random number generated by U_i. Then U_i sends ID_i and A_i to RC for registration through a secure channel.

Step R2. RC computes $B_i = h(ID_i||x), C_i = h(ID_i||h(y)||A_i), D_i = h(B_i||h(x||y)), E_i = B_i \oplus h(x||y)$.

Step R3. RC sends the smart card containing $\{C_i, D_i, E_i, h(\cdot), h(y)\}$ to U_i via a secure channel.

Step R4. U_i stores b into the smart card, and the smart card contains $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$.

2.2 Login phase

U_i performs the following steps to generate the login request.

Step L1. U_i inputs ID_i and PW_i after inserting the smart card into the card reader. Then the smart card computes $A_i = h(b \oplus PW_i), C_i^* = h(ID_i||h(y)||A_i)$, and checks whether the computed C_i^* is equal to the stored C_i. If $C_i^* = C_i$, the smart card continues the next step. Otherwise, the smart card terminates the procedure.

Step L2. The smart card generates a nonce N_i and computes $P_{ij} = E_i \oplus h(h(SID_j||h(y))||N_i), CID_i = A_i \oplus h(D_i||SID_j||N_i), M_1 = h(P_{ij}||CID_i||D_i||N_i)$, $M_2 = h(SID_j||h(y)) \oplus N_i$.

Step L3. U_i submits the login request $\{P_{ij}, CID_i, M_1, M_2\}$ to S_j.

2.3 Verification Phase

S_j and U_i execute the following steps for mutual authentication and key agreement after S_j received U_i's login request.

Step V1. S_j computes $N_i = h(SID_j||h(y)) \oplus M_2, E_i = P_{ij} \oplus h(h(SID_j||h(y))||N_i)$, $B_i = E_i \oplus h(x||y), D_i = h(B_i||h(x||y)), A_i = CID_i \oplus h(D_i||SID_j||N_i)$ with received login request $\{P_{ij}, CID_i, M_1, M_2\}$ and its secret keys $h(SID_j||h(y)), h(x||y)$.

Step V2. S_j computes and checks the equation $h(P_{ij}||CID_i||D_i||N_i) \overset{?}{=} M_1$. If they are not equal, S_j rejects the login request. Otherwise, S_j accepts and generates a nonce N_j to compute $M_3 = h(D_i||A_i||N_j||SID_j), M_4 = A_i \oplus N_i \oplus N_j$. After that, S_j sends the reply message $\{M_3, M_4\}$ to U_i

Step V3. U_i computes $N_j = A_i \oplus N_i \oplus M_4, h(D_i||A_i||N_j||SID_j)$ after receiving $\{M_3, M_4\}$. If the equation $h(D_i||A_i||N_j||SID_j) = M_3$ holds, U_i successfully authenticates S_j and continues computing the mutual authentication message $M_5 = h(D_i||A_i||N_i||SID_j)$ and sending $\{M_5\}$ to S_j. Otherwise, U_i rejects the message and terminates the current session.

Step V4. S_j computes and checks $h(D_i||A_i||N_i||SID_j) \overset{?}{=} M_5$ after receiving $\{M_5\}$. If the equation holds, S_j successfully authenticates U_i and mutual authentication is completed. Otherwise, the session will be terminated.

Finally, U_i and S_j compute and share the session key $SK = h(D_i||A_i||N_i||N_j||SID_j)$.

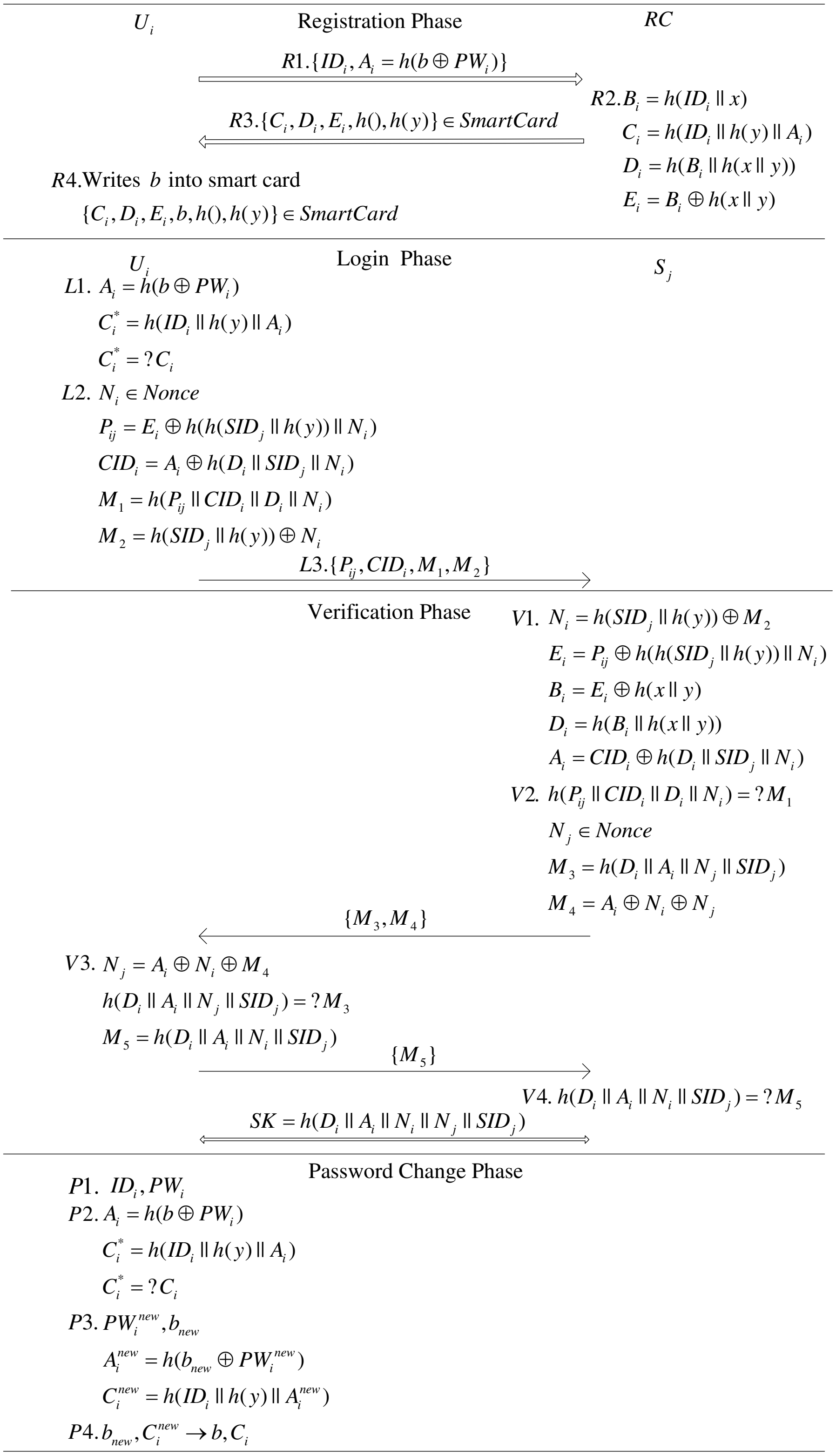
2.4 Password Change Phase

U_i can update PW_i freely anytime and anywhere by executing the following steps off-line.

Step P1. U_i inputs ID_i and PW_i after inserting the smart card into the card reader.

Step P2. The smart card computes $A_i = h(b \oplus PW_i), C_i^* = h(ID_i||h(y)||A_i)$, and checks whether the computed C_i^* is equal to the stored C_i. If $C_i^* = C_i$, U_i inputs a new password PW_i^{new} and the smart card generates a new random number b_{new}. Otherwise, the smart card rejects the password change request and terminates the procedure.

Step P3. The smart card computes $A_i^{new} = h(b_{new} \oplus PW_i^{new}), C_i^{new} = h(ID_i||h(y)||A_i^{new})$.



U_i Registration Phase RC

$R1.\{ID_i, A_i = h(b \oplus PW_i)\}$ $\Longrightarrow$

$R2. B_i = h(ID_i \parallel x)$

$C_i = h(ID_i \parallel h(y) \parallel A_i)$

$D_i = h(B_i \parallel h(x \parallel y))$

$E_i = B_i \oplus h(x \parallel y)$

$R3.\{C_i, D_i, E_i, h(), h(y)\} \in SmartCard$ $\Longleftarrow$

$R4.$Writes b into smart card

$\{C_i, D_i, E_i, b, h(), h(y)\} \in SmartCard$

U_i Login Phase S_j

$L1.\ A_i = h(b \oplus PW_i)$

$C_i^* = h(ID_i \parallel h(y) \parallel A_i)$

$C_i^* = ?C_i$

$L2.\ N_i \in Nonce$

$P_{ij} = E_i \oplus h(h(SID_j \parallel h(y)) \parallel N_i)$

$CID_i = A_i \oplus h(D_i \parallel SID_j \parallel N_i)$

$M_1 = h(P_{ij} \parallel CID_i \parallel D_i \parallel N_i)$

$M_2 = h(SID_j \parallel h(y)) \oplus N_i$

$L3.\{P_{ij}, CID_i, M_1, M_2\}$ $\longrightarrow$

Verification Phase

$V1.\ N_i = h(SID_j \parallel h(y)) \oplus M_2$

$E_i = P_{ij} \oplus h(h(SID_j \parallel h(y)) \parallel N_i)$

$B_i = E_i \oplus h(x \parallel y)$

$D_i = h(B_i \parallel h(x \parallel y))$

$A_i = CID_i \oplus h(D_i \parallel SID_j \parallel N_i)$

$V2.\ h(P_{ij} \parallel CID_i \parallel D_i \parallel N_i) = ?M_1$

$N_j \in Nonce$

$M_3 = h(D_i \parallel A_i \parallel N_j \parallel SID_j)$

$M_4 = A_i \oplus N_i \oplus N_j$

$\{M_3, M_4\}$ $\longleftarrow$

$V3.\ N_j = A_i \oplus N_i \oplus M_4$

$h(D_i \parallel A_i \parallel N_j \parallel SID_j) = ?M_3$

$M_5 = h(D_i \parallel A_i \parallel N_i \parallel SID_j)$

$\{M_5\}$ $\longrightarrow$

$V4.\ h(D_i \parallel A_i \parallel N_i \parallel SID_j) = ?M_5$

$SK = h(D_i \parallel A_i \parallel N_i \parallel N_j \parallel SID_j)$ $\Longleftrightarrow$

Password Change Phase

$P1.\ ID_i, PW_i$

$P2.\ A_i = h(b \oplus PW_i)$

$C_i^* = h(ID_i \parallel h(y) \parallel A_i)$

$C_i^* = ?C_i$

$P3.\ PW_i^{new}, b_{new}$

$A_i^{new} = h(b_{new} \oplus PW_i^{new})$

$C_i^{new} = h(ID_i \parallel h(y) \parallel A_i^{new})$

$P4.\ b_{new}, C_i^{new} \rightarrow b, C_i$

Fig. 1: Li et al.’s scheme

Step P4. The smart card replaces b, C_i with b_{new}, C_i^{new}, and password change phase is successfully finished.

3 Analysis of Li et al.'s scheme

In this section, the analysis of the scheme in [14] is presented to show that their scheme is exposed to lost smart card (smart card breach) attack under the following assumptions:

(i)*Adv* has the full control of public communication channels, i.e., *Adv* can eavesdrop, insert, intercept, modify and delete the messages transmitted through insecure networks arbitrarily.

(ii)Moreover, *Adv* can get the legal user's smart card and extract the data $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$ stored in its memory. Although smart card is usually used in authentication protocols as the tamper-resistant device, which can protect the data stored in its memory, several investigations [15,16,17] show that such an assumption may be problematic. The adversary could extract the data stored in the smart card by monitoring the power assumption or analyzing the leaked information. In addition, *Adv* could get U_i's smart card by stealing, copying, losing, corrupting and so on.

(iii)Note that, although SID_j is not transmitted in the login request, it must be known to every participant involved in the scheme including *Adv*, because SID_j represents the destination of the login request. Generally speaking, service servers always public their addresses or identities in the bulletin board.

Under the above reasonable assumptions, we demonstrate smart card breach attacks are effective on the scheme in [14].

3.1 User Impersonation Attack

If the adversary obtains the information $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$ from U_i's smart card, then *Adv* can impersonate as legal user U_i to login S_j without knowing ID_i or PW_i by performing the following steps.
Step UL1: *Adv* ignores the step L1 in Li et al.'s scheme and executes the next step.
Step UL2: *Adv* forges the login request as follows:

$$\overline{P_{ij}} = E_i \oplus h(h(SID_j||h(y))||\overline{N_i}),$$

$$\overline{CID_i} = \overline{A_i} \oplus h(D_i||SID_j||\overline{N_i}),$$

$$\overline{M_1} = h(\overline{P_{ij}}||\overline{CID_i}||D_i||\overline{N_i}),$$

$$\overline{M_2} = h(SID_j||h(y)) \oplus \overline{N_i},$$

where $\overline{N_i}$ and $\overline{A_i}$ are forged by *Adv*.
Step UL3: *Adv* submits $\{\overline{P_{ij}}, \overline{CID_i}, \overline{M_1}, \overline{M_2}\}$ to S_j as the login request message.

After S_j receiving the login message $\{\overline{P_{ij}}, \overline{CID_i}, \overline{M_1}, \overline{M_2}\}$, S_j and *Adv* perform the following steps.
Step UV1: S_j computes:

$$\overline{N_i} = h(SID_j||h(y)) \oplus \overline{M_2},$$

$$E_i = \overline{P_{ij}} \oplus h(h(SID_j||h(y))||\overline{N_i}),$$

$$B_i = E_i \oplus h(x||y),$$

$$D_i = h(B_i||h(x||y)),$$

$$\overline{A_i} = \overline{CID_i} \oplus h(D_i||SID_j||\overline{N_i}).$$

Step UV2. S_j computes and checks the equation

$$h(\overline{P_{ij}}||\overline{CID_i}||D_i||\overline{N_i}) \overset{?}{=} \overline{M_1}.$$

The equation must hold, because *Adv* forges $\overline{M_1} = h(\overline{P_{ij}}||\overline{CID_i}||D_i||\overline{N_i})$. Then S_j generates a nonce N_j and computes

$$M_3 = h(D_i||\overline{A_i}||N_j||SID_j),$$

$$M_4 = \overline{A_i} \oplus \overline{N_i} \oplus N_j.$$

Finally, S_j sends to U_i the message

$$\{M_3, M_4\}.$$

Step UV3. *Adv* intercepts the message $\{M_3, M_4\}$ from S_j, and computes

$$N_j = \overline{A_i} \oplus \overline{N_i} \oplus M_4,$$

and then sends $\{\overline{M_5}\}$ to S_j.
Step UV4. After receiving the message $\overline{M_5}$ from *Adv*, S_j computes and checks the equation

$$h(D_i||\overline{A_i}||\overline{N_i}||SID_j) \overset{?}{=} \overline{M_5}.$$

The equation must be hold, because *Adv* forges it with correct D_i. Then, S_j successfully authenticates *Adv* as the legal user and the mutual authentication is completed.

After mutual authentication phase, *Adv* and S_j compute and share the session key

$$\overline{SK} = h(D_i||\overline{A_i}||\overline{N_i}||N_j||SID_j).$$

Finally, under the assumption of smart card breach, *Adv* successfully impersonates U_i to cheat S_j without knowing ID_i or PW_i, because S_j cannot distinguish $\overline{A_i}, \overline{N_i}$ from A_i, N_i, which are generated by U_i or *Adv*.

3.2 Server Masquerade Attack

Adv can masquerade as the remote server to cheat the legal user U_i with the extracted information $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$ from U_i's smart card. The details of such attack are described as follows.

When U_i wants to login S_j and executes the step L1-L4 in Li et al.'s scheme as usual, and then U_i submits $\{P_{ij}, CID_i, M_1, M_2\}$ to S_j as the login request message.

After that, Adv intercepts the login request $\{P_{ij}, CID_i, M_1, M_2\}$ transmitted from U_i to S_j over the insecure channels and performs the following steps to masquerade as S_j to cheat U_i without knowing $h(x||y)$.

Step SV1: Adv computes:

$$N_i = h(SID_j||h(y)) \oplus M_2,$$

$$A_i = CID_i \oplus h(D_i||SID_j||N_i).$$

Step SV2: Adv generates a nonce $\overline{N_j}$ and computes

$$\overline{M_3} = h(D_i||A_i||\overline{N_j}||SID_j),$$

$$\overline{M_4} = A_i \oplus N_i \oplus \overline{N_j}.$$

After that, Adv sends to U_i the message

$$\{\overline{M_3}, \overline{M_4}\}.$$

Step SV3. Upon receiving the message $\{\overline{M_3}, \overline{M_4}\}$ from Adv, U_i computes and checks

$$\overline{N_j} = A_i \oplus N_i \oplus \overline{M_4},$$

$$h(D_i||A_i||\overline{N_j}||SID_j) \overset{?}{=} \overline{M_3}.$$

The equation must hold, because Adv forges $\overline{M_3}$ with correct D_i, A_i, SID_j. Then, U_i computes and sends the mutual authentication message

$$M_5 = h(D_i||A_i||N_i||SID_j)$$

to the server S_j.

Step SV4. Adv intercepts the message $M_5 = h(D_i||A_i||N_i||SID_j)$, computes and checks the equation

$$h(D_i||A_i||N_i||SID_j) \overset{?}{=} M_5.$$

If the equation holds, Adv successfully authenticates U_i and the mutual authentication is completed. This step could be ignored, because Adv's target is to masquerade as the remote server to cheat the legal user.

After mutual authentication phase, U_i and Adv compute and share the session key

$$\overline{SK} = h(D_i||A_i||N_i||\overline{N_j}||SID_j).$$

Finally, under the assumption of smart card breach, Adv successfully masquerades as the remote server S_j to cheat U_i without knowing $h(x||y)$.

3.3 Forward Security

Forward security of the session key means that there is nobody can recover the session key even if all the participants' credentials are compromised to the adversary, who has recorded all the communication transcripts. Nevertheless, Li et al. do not consider such an attack in their proposal, because the adversary can reconstruct the previous session keys as well as the legal user whose smart card is corrupted and compromised. In details, if Adv gets U_i's smart card and extracts the data $\{C_i, D_i, E_i, b, h(\cdot), h(y)\}$ stored in its memory, while recording all the transcripts between U_i and S_j. Then Adv can recover the key materials $N_i = h(SID_j||h(y)) \oplus M_2, A_i = CID_i \oplus h(D_i \oplus SID_j \oplus N_i), N_j = M_4 \oplus A_i \oplus N_i$ and reveal the session key $SK = h(D_i||A_i||N_i||N_j||SID_j)$.

4 Preliminaries

In this section, we introduce some concepts and technologies used in our proposal.

4.1 Client Puzzle

Definition 1. *Client Puzzle* [28,29,25] consists of five algorithms, named: $CP\_Setup, CP\_GenPuz, CP\_FindSoln, CP\_VerAuth, CP\_VerSoln$.

(i) CP_Setup : This probabilistic polynomial-time algorithm takes as input a security parameter λ, generates and returns a set of public parameters $PubPara$ and a secret key s, the former of which includes a puzzle difficulty parameter space $QSpace$.

(ii) CP_GenPuz : This probabilistic polynomial-time algorithm takes as inputs $s, QSpace$ and a session string str, and returns a client puzzle $ClientPuzzle$.

(iii) $CP\_FindSoln$: This is a probabilistic puzzle solving algorithm takes as inputs $ClientPuzzle$ and t, and returns potential solution $soln$ after running time at most t.

(iv) $CP\_VerAuth$: This deterministic polynomial time puzzle authenticity verification algorithm takes as inputs $s, ClientPuzzle$ and returns true or false

(v) $CP\_VerSoln$: This deterministic polynomial time puzzle solution verification algorithm takes as inputs $s, str, ClientPuzzle, soln$ and returns true or false.

In our construction, we take the mechanism *Client Puzzle* as a component. It generates a client puzzle $ClientPuzzle$ by service server in terms of its current system overhead and verifies user's $soln$ as the reply of the $ClientPuzzle$. Figure 2 illustrates the main process of our construction. In addition, we design *Client Puzzle* with the technology *CAPTCHA* (Completely Automated Public Turing Test to Tell Computers and Humans Apart)

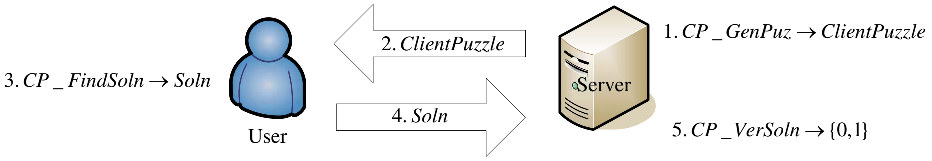


Fig. 2: The Component of Client Puzzle

[30] to resist DoS attack. In details, the service server generates a client puzzle and sends it to the user as the reply of access request in the form of picture. The solution of the client puzzle can be found easily for a real person (user), but hard for a machine. Generally speaking, it can prevent massive login request sent by the machine, then resist denial of service attacks caused by exhausting computation/communication resource.

4.2 Fuzzy Extractor

Definition 2. *Fuzzy Extractor* [31,18] generates a nearly random string R from its biometrics input w in an error tolerant way. If the biometrics input changes but remains close, the extracted R can be recovered as the same as before under the help of the auxiliary string P. *Fuzzy Extractor* consists of two procedures called FE_Gen, FE_Rep described formally by the parameters (M,m,l,t,ε) as follows.

(i)FE_Gen : This is a probabilistic generation algorithm, which on biometrics input $w \in M$ outputs an ”extracted” string $R \in \{0,1\}^l$ and an auxiliary string $P \in 0,1^*$. A set M is a metric space with a distance function dis:$M \times M \rightarrow R^+ = [0,\infty)$, which obeys various natural properties. For any distribution W on M of min-entropy m, if $< R,P >\leftarrow Gen(w)$, then we have $SD(< R,P >,< U_l,P >) \leq \varepsilon$, where $SD(A,B) = \frac{1}{2}\sum_v |$ Pr$(A = v)-$ Pr$(B = v)|$ denotes the statistical distance between two probability distribution A and B, and U_l denotes the uniform distribute on l-bit binary strings.

(ii)FE_Rep : This is a deterministic reproduction procedure allowing to recover R from the corresponding auxiliary string P and any vector w' close to w: for all $w,w' \in M$ satisfying dis$(w,w') \leq t$ if $< R,P >\leftarrow$ Gen(w), then we have $Rep(w',P) = R$.

In our construction, we take advantage of algorithm *Fuzzy Extractor* to generate a nearly random number R and the auxiliary random number P. The main process of *Fuzzy Extractor* is shown in Figure 3. Here, we utilize R and PW_i to protect authentication credentials from being compromised, meanwhile using R to protect PW_i from being guessed. Thus, *Fuzzy Extractor* is a critical mechanism, because the secret number R can be recovered only by the unique legal user owning corresponding biometrics w'_i and having the auxiliary random number P stored in his/her mobile device.

4.3 MAC

Definition 3. *MAC* is a tuple of probabilistic polynomial algorithms $MAC\_Gen, MAC\_Mac, MAC\_Vrfy$ fulfilling the follows:

(i)MAC_Gen : This algorithm takes as input 1^n and outputs a uniformly distributed key k of length n denoted by $k \leftarrow MAC\_Gen(1^n)$.

(ii)MAC_Mac : This algorithm receives for input some $k \in \{0,1\}^n$ and $m \in \{0,1\}^*$, and outputs some $t \in \{0,1\}^*$, which we call it MAC_{tag}.

(iii)MAC_Vrfy : This algorithm receives for input some $k \in \{0,1\}^n, m \in \{0,1\}^*$ and $t \in \{0,1\}^*$, and outputs a bit $b \in \{0,1\}$.

(iv)For every n, every $k \in \{0,1\}^n$ and every $m \in \{0,1\}^*$, it holds that $MAC\_Vrfy_k(m, MAC\_Mac(m)) = 1$.

In our construction, we take advantage of *MAC* to verify the transmitted messages online. It identifies the source of received information indeed from the intended participant. It also plays the role of identity authentication.

4.4 Diffie-Hellman key exchange

Definition 4. *Diffie-Hellman key exchange* is a primitive cryptographic protocol described as follows:

(i)DH_Setup : The system public parameters consist of a large prime number q, a multiplicative group $(G,\cdot)$ and an element $g \in G$ as the generator of group G, where the order of g is a large prime number p.

(ii)$DH\_Exchange$: The participants A and B randomly choose ephemeral secret numbers a and b, respectively. Then A computes $\widehat{A} = g^a$ and B computes $\widehat{B} = g^b$. After that, A and B exchange their public key material $\widehat{A}$ and $\widehat{B}$ to each other, and keep their secret number a and b private.

(iii)$DH\_Generation$: A computes the session key $SK = \widehat{B}^a$ and B computes the session key $SK = \widehat{A}^b$.

As well as known, the primitive DH key exchange protocol is vulnerable to man-in-the-middle attack. Therefore, in our construction, we embedded an improved authenticated DH key exchange protocol to overcome the aforementioned security flaw.

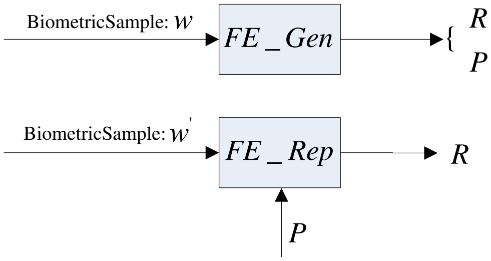


Fig. 3: The Component of Fuzzy Extractor

5 Proposal

In this section, we devise a three-factor authentication scheme with session key agreement for multi-server environment, including registration phase, login and authentication with key exchange phase, and password change phase.

5.1 Registration Phase

There are two operations in registration phase, including service server registration and user registration.

5.1.1 Service Server Registration

When the service server S_j wants to provide services for the users managed by the registration center RC, it should register in RC as a legal service server by performing the following steps:

(i)$S_j \rightarrow RC : SID_j$
S_j chooses its identifier SID_j and sends it to RC over public channels.

(ii)$RC \Rightarrow S_j : MK_j$
RC generates the MAC key MK_j by calling MAC_Gen and sends it to S_j through secure channels.

S_j registers as a legal service server and obtains its MAC key MK_j, which should be kept secure by itself. Then RC writes SID_j with corresponding MK_j to its server registration list. It means that RC and S_j pre-share the MAC key MK_j, and they will use it to confirm the message resource and ensure the legality of each server without compromising MK_j.

5.1.2 User Registration

When a user wants to obtain the service, he/she should register in RC by performing the following steps:

(i)$U_i \Rightarrow RC : \{ID_i, RPW_i\}$
U_i inputs his/her chosen identity ID_i and password PW_i, and the extracted biometrics w_i into the mobile device. The mobile device generates the secret random number R_i and the auxiliary string P_i by calling algorithm $(R_i, P_i) \leftarrow FE\_Gen$, then computes $RPW_i = h(PW_i||R_i)$ and sends $\{ID_i, RPW_i\}$ to RC over a secure channel for registration.

(ii)$RC \Rightarrow U_i : \{A_i, g^x\}$
RC writes ID_i into its user registration list if he/she satisfies the registration policy and computes $C_i = h(ID_i||x), A_i = h(ID_i \oplus RPW_i) \oplus C_i$. Then, RC sends $\{A_i, g^x\}$ to U_i, where x and g^x are master secret key and public key of registration center.

(iii)U_i stores $\{P_i, A_i, g^x, V_i\}$ in its mobile device, where $V_i = h(ID_i||RPW_i)$ is computed by mobile device.

After that, U_i registers in RC successfully and obtains his/her credential C_i, which is protected by ID_i, PW_i, R_i and used for remote authentication.

5.2 Login and Authentication with Key Exchange Phase

Figure 4 shows the architecture overview of login and authentication with key exchange phase. The interactions among U_i, S_j and RC consists of **Access_Request, Access_Reply, Login_Request, Auth_Request, Auth_Reply, Login_Reply and Key_Confirm**.

When U_i wants to obtain services from S_j, U_i and S_j should authenticate each other and exchange a secure session key for private communication. RC plays the role of the third trusted party between U_i and S_j to help them establish trust relationships. The details about this phase are described as follows.

5.2.1 Access_Request

$U_i \rightarrow S_j : \{SID_j, g^s\}$

U_i generates the access request $\{SID_j, g^s\}$ and sends it to S_j as the initialization of this session, where g is the generator of multiplicative group $(G, \cdot)$ with prime order p, $s \in Z_q^*$ is a nonce chosen by U_i and q is a large prime number. Note that, we denote the random ephemeral number as nonce in this paper. The mobile device stores and denotes $\{SID_j, g^s\}$ as the session identifier.

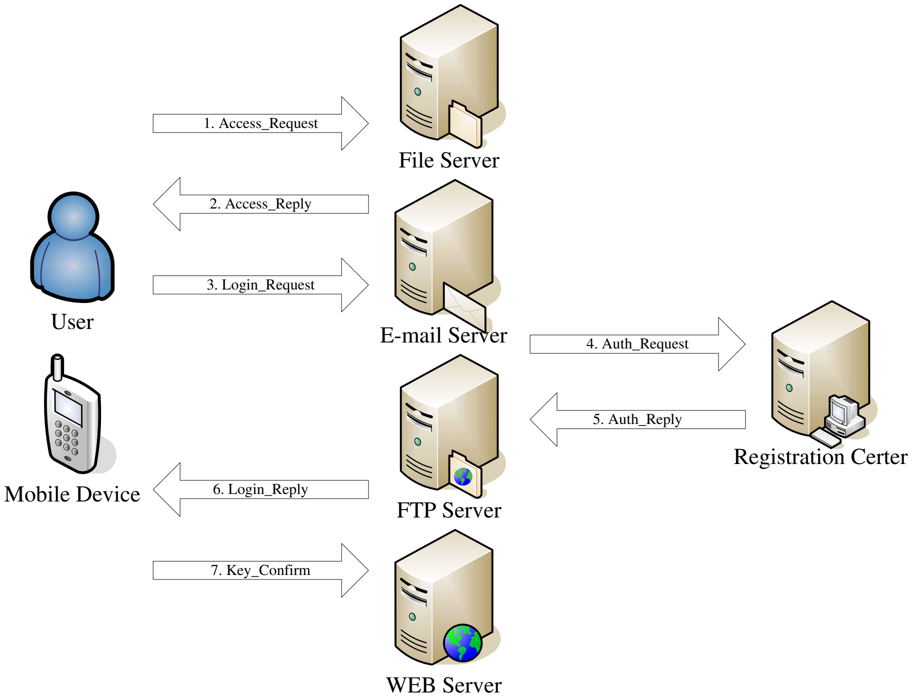


Fig. 4: Architecture Overview of Our Devise

5.2.2 Access_Reply

$S_j \rightarrow U_i : \{ClientPuzzle\}$

Upon receiving the access request, S_j replies the $\{ClientPuzzle\}$ to U_i, where client puzzle is pre-generated by calling the algorithm CP_GenPuz. Note that, the solution of the client puzzle consists of the contributor g^t of the session key, where $t \in Z_q^*$ is a nonce. In addition, $\{ClientPuzzle\}$ is sent by the technology *CAPTCHA* to resist DoS attack. S_j stores and denotes $\{SID_j, g^s, g^t\}$ as the session identifier.

5.2.3 Login_Request

$U_i \rightarrow S_j : \{SID_j, g^s, g^t, CID_i, MAC_1\}$

Upon receiving **Access_Reply**, U_i first solves $ClientPuzzle$ to get g^t by human cognition, then inputs ID_i', PW_i' and biometric data w_i' into the mobile device. The mobile device computes $V_i' = h(ID_i'||RPW_i') = h(ID_i'||h(PW_i'||R_i'))$, where $R_i' = FE\_Rep(w_i', P_i)$, and then verifies $V_i' \stackrel{?}{=} V_i$. If $V_i' = V_i$, mobile device confirms its legal holder and continues the next process. Otherwise, mobile device terminates the current process. (We note that, the equation $V_i' = V_i$ means that $ID_i' = ID_i, PW_i' = PW_i$ and $R_i' = R_i, RPW_i' = RPW_i$ where w_i' should be close to w_i enough and satisfy the requirement of *Fuzzy Extractor*).

After confirming its legal holder, mobile device computes $C_i = A_i \oplus h(ID_i \oplus RPW_i), CID_i = (g^x)^s \oplus ID_i, MAC_1 = MAC\_Mac_{C_i}(SID_j||g^s||g^t||CID_i)$ and sends the **Login_Request** $\{SID_j, g^s, g^t, CID_i, MAC_1\}$ to S_j. Meanwhile, mobile device should update its session identifier as $\{SID_j, g^s, g^t, CID_i\}$.

5.2.4 Auth_Request

$S_j \rightarrow RC : \{SID_j, g^s, g^t, CID_i, MAC_1, MAC_2\}$

After confirming validity of $ClientPuzzle$ by the algorithm CP_Vrfy, S_j forwards the **Login_Request** with $MAC_2 = MAC\_Mac_{MK_j}(SID_j||g^s||g^t||CID_i||MAC_1)$ to RC as **Auth_Request**.

5.2.5 Auth_Reply

$RC \rightarrow S_j : \{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_3, MAC_4\}$

Upon receiving S_j's **Auth_Request**, RC first verifies

$$MAC\_Vrfy_{MK_j}(SID_j||g^s||g^t||CID_i||MAC_1, MAC_2) \stackrel{?}{=} 1.$$

If MAC_2 is true, RC computes $ID_i = CID_i \oplus (g^s)^x, C_i = h(ID_i||x)$ and verifies $MAC\_Vrfy_{C_i}(SID_j||g^s||g^t||CID_i, MAC_1) \stackrel{?}{=} 1$. If MAC_1 is true, RC confirms the legality of the remote user and generates **Auth_Reply** $\{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_3, MAC_4\}$ to tell S_j the authentication result, where z is a nonce generated by RC,

$$MAC_3 = MAC\_Mac_{C_i}(SID_j||g^s||g^t||ID_i||h(C_i||z))$$

and $MAC_4 = MAC\_Mac_{MK_j}(SID_j||g^s||g^t||CID_i||h(C_i||z))$.

5.2.6 Login_Reply

$S_j \rightarrow U_i : \{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_3, MAC_5\}$

After receiving **Auth_Reply**, S_j confirms the legality of U_i through verifying $MAC\_Vrfy_{MK_j}(SID_j||g^s||g^t||CID_i||h(C_i||z), MAC_4) = 1$, then computes $SK = h((g^s)^t||h(C_i||z)), MAC_5 = MAC\_Mac_{SK}(SID_j||g^s||g^t||CID_i||h(C_i||z))$ and sends **Login_Reply** $\{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_3, MAC_5\}$ to U_i. S_j updates the session identifier as $\{SID_j, g^s, g^t, CID_i, h(C_i||z)\}$.

5.2.7 Key_Confirm

$U_i \rightarrow S_j : \{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_6\}$

Upon receiving **Login_Reply**, U_i verifies

$$MAC\_Vrfy_{C_i}(SID_j||g^s||g^t||ID_i||h(C_i||z), MAC_3) \overset{?}{=} 1.$$

If it is true, U_i computes $SK = h((g^t)^s||h(C_i||z))$ and verifies

$$MAC\_Vrfy_{SK}(SID_j||g^s||g^t||CID_i||h(C_i||z), MAC_5) \overset{?}{=} 1.$$

If it is true, U_i authenticates S_j and replies the message **Key_Confirm** $\{SID_j, g^s, g^t, CID_i, h(C_i||z), MAC_6\}$, where

$$MAC_6 = MAC\_Mac_{SK}(SID_j||g^t||g^s||CID_i||h(C_i||z)).$$

Then, U_i updates its session identifier by $\{SID_j, g^s, g^t, CID_i, h(C_i||z)\}$.

S_j successfully authenticates U_i and confirms the session key by verifying $MAC\_Vrfy_{SK}(SID_j||g^t||g^s||CID_i||h(C_i||z), MAC_6) = 1$.

Finally, S_j and U_i authenticate each other and establish a secure channel denoted by the session identifier $\{SID_j, g^s, g^t, CID_i, h(C_i||z)\}$. The session key $SK = h(g^{st}||h(C_i||z))$ is used for encrypting and decrypting the subsequent communications. In addition, this process will be terminated by any mistake or error caused by the verification procedures, and the participant should send ”Error Warning!” as the reply to the intended participants.

5.3 Password Change Phase

When U_i wants to change his/her password for security concern, U_i should pass the verification process of the mobile device first (see 5.2.3), and then inputs a new password PW_i^{new}. Mobile device computes $V_i^{new} = h(ID_i||h(PW_i^{new}||R_i))$, $A_i^{new} = A_i \oplus h(ID_i \oplus h(PW_i||R_i)) \oplus h(ID_i \oplus h(PW_i^{new}||R_i))$. At last, mobile device replaces V_i, A_i with V_i^{new}, A_i^{new}

6 Proof of Our Scheme

In this section, we demonstrate the validity of our proposed scheme by BAN-logic [2]. The notations used in BAN-logic analysis are defined as follows:

- $\mathscr{P} \mid\equiv X$: The principal $\mathscr{P}$ believes a statement X or $\mathscr{P}$ would be entitled to believe X.
- $\sharp(X)$: The formula X is fresh.
- $\mathscr{P} \Rightarrow X$: The principal $\mathscr{P}$ has jurisdiction over the statement X.
- $\mathscr{P} \lhd X$: The principal $\mathscr{P}$ sees the statement X.
- $\mathscr{P} \mid\sim X$: The principal $\mathscr{P}$ once said the statement X.
- (X, Y): The formula X or Y is one part of the formula (X, Y).
- $\langle X \rangle_Y$: The formula X is combined with the formula Y.
- $\{X\}_Y$: The formula X is encrypted under the key Y.
- $\mathscr{P} \overset{K}{\longleftrightarrow} \mathscr{Q}$: The principals $\mathscr{P}$ and $\mathscr{Q}$ use the shared key K to communicate. Here, K will never be discovered by any principal except for $\mathscr{P}$ and $\mathscr{Q}$.
- $\mathscr{P} \overset{K}{\rightleftharpoons} \mathscr{Q}$: K is shared secret known to $\mathscr{P}$, $\mathscr{Q}$, and possibly to one trusted by them.
- SK: The session key used in the current session.

Some main logical postulates of BAN-logic are described as follows:

- The message-meaning rule: $\frac{\mathscr{P}|\equiv\mathscr{Q}\overset{K}{\rightleftharpoons}\mathscr{P}, \mathscr{P}\lhd\langle X\rangle_K}{\mathscr{P}|\equiv\mathscr{Q}|\sim X}$.
- The freshness-conjuncatenation rule: $\frac{\mathscr{P}|\equiv\sharp(X)}{\mathscr{P}|\equiv\sharp(X,Y)}$.
- The nonce-verification rule: $\frac{\mathscr{P}|\equiv\sharp(X), \mathscr{P}|\equiv\mathscr{Q}|\sim X}{\mathscr{P}|\equiv\mathscr{Q}|\equiv X}$.
- The jurisdiction rule: $\frac{\mathscr{P}|\equiv\mathscr{Q}\Rightarrow X, \mathscr{P}|\equiv\mathscr{Q}|\equiv X}{\mathscr{P}|\equiv X}$, $\frac{\mathscr{P}|\equiv(X,Y)}{\mathscr{P}|\equiv X}$, $\frac{\mathscr{P}\lhd(X,Y)}{\mathscr{P}\lhd X}$, $\frac{\mathscr{P}|\equiv\mathscr{Q}|\sim(X,Y)}{\mathscr{P}|\equiv\mathscr{Q}|\sim X}$.

According to the analytic procedures of BAN-logic, we list the verification goals of the proposed scheme below:

Goal.1: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal.2: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Next, the proposed scheme is arranged from the generic type to the idealized form in the following:

Login_Request: $U_i \rightarrow S_j$: $(SID_j, g^s, g^t, CID_i, \langle SID_j, g^s, g^t, CID_i\rangle_{C_i})$

Auth_Request: $S_j \rightarrow RC$: $(SID_j, g^s, g^t, CID_i, \langle SID_j, g^s, g^t, CID_i\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, MAC_1\rangle_{MK_j})$

Auth_Reply: $RC \rightarrow S_j$: $(SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, \langle SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t)\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s)\rangle_{MK_j})$

Login_Reply: $S_j \rightarrow U_i$: $(SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, \langle SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t)\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}\rangle_{SK})$

Key_Confirm: $U_i \rightarrow S_j$: $(SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, \langle g^s, g^t, g^{st}, CID_i, \langle z\rangle_{C_i}\rangle_{SK})$

We make the following assumptions about the initial state of the scheme to further analyze the proposed scheme:

A.1: $U_i |\equiv (U_i \overset{C_i}{\rightleftharpoons} RC)$

A.2: $S_j |\equiv (S_j \overset{MK_j}{\rightleftharpoons} RC)$

A.3: $RC |\equiv (U_i \overset{C_i}{\rightleftharpoons} RC)$

A.4: $RC |\equiv (S_j \overset{MK_j}{\rightleftharpoons} RC)$

A.5: $U_i |\equiv \sharp(g^s)$

A.6: $S_j |\equiv \sharp(g^t)$

A.7: $S_j |\equiv RC \Rightarrow (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$

A.8: $S_j |\equiv t$

A.9: $U_i |\equiv RC \Rightarrow (SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$

A.10: $U_i |\equiv s$

Based on the above-mentioned assumptions and rules of BAN-logic, we analyze the idealized form of the proposed scheme and the main procedures of proof as follows:

According to the *Auth_Request*, we obtain:
$RC \triangleleft (SID_j, g^s, g^t, CID_i, \langle SID_j, g^s, g^t, CID_i\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, MAC_1\rangle_{MK_j})$.

According to the jurisdiction rule, we obtain:
$RC \triangleleft \langle SID_j, g^s, g^t, CID_i\rangle_{C_i}$,
$RC \triangleleft \langle SID_j, g^s, g^t, CID_i, MAC_1\rangle_{MK_j}$.

According to the assumption A.3, A.4 and the message-meaning rule, we obtain:
$RC |\equiv U_i |\sim (SID_j, g^s, g^t, CID_i)$,
$RC |\equiv S_j |\sim (SID_j, g^s, g^t, CID_i, MAC_1)$.

According to the jurisdiction rule, we obtain:
$RC |\equiv U_i |\sim g^s$,
$RC |\equiv S_j |\sim g^t$.

According to the *Auth_Reply*, we obtain:
$S_j \triangleleft (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, \langle SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t)\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s)\rangle_{MK_j})$.

According to the jurisdiction rule, we obtain:
$S_j \triangleleft \langle SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s)\rangle_{MK_j}$

According to the assumption A.2 and the message-meaning rule, we obtain:
$S_j |\equiv RC |\sim (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$.

According to the assumption A.6 and the freshness-conjuncatenation rule, we obtain:
$S_j |\equiv \sharp(SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$.

According to $S_j |\equiv RC |\sim (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$ and the nonce-verification rule, we obtain:
$S_j |\equiv RC |\equiv (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$.

According to the assumption A.7 and the jurisdiction rule, we obtain:
$S_j |\equiv (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, (U_i |\sim g^s))$.

According to the jurisdiction rule, we obtain:
$S_j |\equiv (U_i |\sim g^s)$,
$S_j |\equiv g^s$,
$S_j |\equiv \langle z\rangle_{C_i}$.

According to $SK = h(g^{st} \| h(C_i \| z))$ and the assumption A.8, we obtain:

$S_j |\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ **(Goal 2)**

According to the *Login_Reply*, we obtain:
$U_i \triangleleft (SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}, \langle SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t)\rangle_{C_i}, \langle SID_j, g^s, g^t, CID_i, \langle z\rangle_{C_i}\rangle_{SK})$.

According to the jurisdiction rule, we obtain:
$U_i \triangleleft \langle SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t)\rangle_{C_i}$.

According to the assumption A.1 and the message-meaning rule, we obtain:
$U_i |\equiv RC |\sim (SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$.

According to the assumption A.5 and the freshness-conjuncatenation rule, we obtain:
$U_i |\equiv \sharp(SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$.

According to $U_i |\equiv RC |\sim (SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$ and the nonce-verification rule, we obtain:
$U_i |\equiv RC |\equiv (SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$.

According to the assumption A.9 and the jurisdiction rule, we obtain:
$U_i |\equiv (SID_j, g^s, g^t, ID_i, \langle z\rangle_{C_i}, (S_j |\sim g^t))$.

According to the jurisdiction rule, we obtain:
$U_i |\equiv (S_j |\sim g^t)$,
$U_i |\equiv g^t$,
$U_i |\equiv \langle z\rangle_{C_i}$.

According to $SK = h(g^{st} \| h(C_i \| z))$ and the assumption A.10, we obtain:

$U_i |\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$ **(Goal 1)**.

7 Analysis and Discussion

In this section, we analyze our enhanced scheme in the view of security, efficiency, robustness and practicability.

7.1 Security Analysis

Before we analyze the security of our enhanced scheme, we present some assumptions:

(i) The assumption of adversary's capabilities have been shown in section 3. Specifically, mobile device is non-tamper resistant device, i.e., the information stored in its memory can be compromised to the adversaries.

(ii) The problems of decision Diffie-Hellman (DDH) problem, computational Diffie-Hellman (CDH) problem and discrete logarithm problem are hard to be solved by probability polynomial time algorithm.

(iii)The components of cryptographic primitives are secure under the security definitions, such as hash function, *MAC*, *Client Puzzle* and *Fuzzy Extractor*.

Generally speaking, there are three targets for the adversary to break down authentication with key agreement schemes, e.g., authentication, session key and user's credential. In details, no one can impersonate the other participants, the session key must be known only to intended participants, user's credential as his/her privacy cannot be compromised to any others. The following analysis is discussed in the view of authentication, session key and user's credential.

7.1.1 Authentication

Firstly, the adversary cannot impersonate as the legal user to access service servers. In our construction, the adversary needs to obtain U_i's credential C_i to generate legal **Login_Request**, which includes $MAC_1 = MAC\_Mac_{C_i}(SID_j||g^s||g^t||CID_i)$. C_i consists of RC's master secret key x and is protected by U_i's PW_i, w_i. In addition, C_i as the input key of MAC_1 is secure under the assumption of one-way *MAC* algorithm. Without knowing C_i, *Adv* cannot pass *RC*'s authentication. Therefore, user impersonation attack cannot be effective.

Secondly, the adversary cannot masquerade as the service server to cheat U_i or *RC*. Without knowing MK_j, which is pre-shared by S_j and *RC*, *Adv* cannot generate the legal **Auth_Request**, which consists of $MAC_2 = MAC\_Mac_{MK_j}(SID_j||g^s||g^t||CID_i||MAC_1)$ to masquerade as S_j to cheat *RC*. In addition, *Adv* cannot masquerade as S_j to cheat U_i without $MAC_5 = MAC\_Mac_{SK}(SID_j||g^s||g^t||CID_i||h(C_i||z))$, where *SK* is the key of MAC_5, because *Adv* cannot obtain the session key. Therefore, service server masquerade attack cannot be effective.

At last, the adversary cannot spoof U_i or S_j as *RC*. Without knowing MK_j or C_i, *Adv* cannot generate the valid

$$MAC_3 = MAC\_Mac_{C_i}(SID_j||g^s||g^t||ID_i||h(C_i||z))$$

or $MAC_4 = MAC\_Mac_{MK_j}(SID_j||g^s||g^t||CID_i||h(C_i||z))$. Therefore, *RC* spoofing attack cannot be effective.

7.1.2 Session Key

Our proposal provides authenticated key exchange to establish a secure session key $SK = h(g^{st}||h(C_i||z))$. Firstly, the session key is authenticated by U_i and S_j, because g^s and g^t are contributors for the session key and message authentication code. In other words, man-in-the-middle attack can be prevented. Secondly, the session key is fresh, because the materials g^s and g^t are generated in each session by authenticated participants. In addition, the mechanism of key exchange can achieve forward and backward security, because any compromised session key cannot affect the security of the other session keys, even if the long term secret confidential C_i is compromised. Moreover, MAC_6 supports key confirmation for U_i and S_j.

7.1.3 Credential

Three-factor authentication scheme should consider the privacy of user's credential, i.e., user's password PW_i, and biometrics w_i, secret random number R_i. Our enhanced scheme focuses on protecting these credentials. Most of all, users no longer need deliver their passwords or biometrics to the honest but curious remote servers as the verification tables, because password and biometrics as user's privacy will be reused in different areas for authentication. Secondly, mobile device must confirm its legal holder by the verification parameter $V_i = h(ID_i||h(PW_i||R_i))$ to validate ID_i, PW_i and w_i, P_i. Then, C_i can be correctly recovered from $A_i = h(ID_i \oplus h(PW_i||R_i)) \oplus C_i$. Without knowing the correct and matching authentication factors, *Adv* cannot acquire C_i. Moreover, R_i and PW_i are protected each other from being compromised, even if the mobile device is non-tamper-resistant.

7.2 Performance Analysis

Table 2 shows the detailed comparisons of computation cost among the schemes in [14,32,35] and ours. Specifically, the Diffie-Hellman problem in finite field F_p and on the elliptic curve $E_p(a,b)$ plays the same role, but the security level and computation costs are different. Here, we regard the modular exponentiation computation in F_p as the point multiplicative computation on $E_p(a,b)$ for convenient to compare the computation cost. In addition, the symmetric operations T_{Sym} cost the same level of complexity as well as T_h and T_{Mac} are symmetric operations. Therefore, we compare the computation cost based on the time T_h and T_{PM} in total.

The comparison in table 2 shows that our scheme is efficient than the scheme in [32], while costing more than the schemes in [14]. The total cost is similar with Guo and Wen's scheme in [35].

7.3 Discussion

Table 3 shows the comparisons among the schemes in [14,32,35] and our enhancement in the view of security, robustness, practicability and efficiency. For security, our enhancement can resist smart card lost or stolen attack if the adversary can compromise the data stored in mobile device. In addition, forward secrecy for session key can be provided in case of the leakage of credentials. For flexibility, in order to satisfy the specific applications, our

Table 2: Performance Cost

	T_h	T_{MAC}	T_{Sym}	T_{Exp}	T_{PM}	Total
[14]	15	0	0	0	0	$15T_h$
[32]	21	0	0	0	8	$21T_h+8T_{PM}$
[35]	10	0	8	6	0	$18T_h+6T_{PM}$
ours	7	12	0	6	0	$19T_h+6T_{PM}$

T_h denotes the time consumption of hash operation.
T_{MAC} denotes the time consumption of *MAC*.
T_{Sym} denotes the time consumption of symmetric encryption or decryption.
T_{Exp} denotes the time consumption of exponential operation.
T_{PM} denotes the time consumption of point multiplicative on elliptic curve.

Table 3: Comparisons

	Security	Robustness	Practicability	Efficiency
[14]	○	○	●	●
[32]	●	○	●	○
[35]	●	○	●	●
ours	●	●	●	●

● denotes offering more advantages.
○ denotes offering less advantages.

proposal provides user anonymity, while maintaining static ID login mode if CID_i is replaced by ID_i. For example, users would like to access the online stores with anonymous mode to protect their privacy on the Internet; but when users pay for their commodities by electronic bank systems, users must access their account with real-identity mode. Therefore, our three-factor authentication can be reduced to two-factor authentication with less security properties, if the scheme gets rid of biometrics w_i and replaces R_i by a random generated number. Moreover, according to the actual conditions, biometrics verification system cannot be available anywhere, thus it is necessary to sacrifice the security attributes appropriately for the requirements of practical application. At last, our designation is practicable for users, because mobile device is more prevalent and convenient to take along with oneself than smart card. However, the above benefits are obtained through sacrificing the computation cost, thus the computation cost of our proposal is increased.

8 Conclusion

In this paper, we analyze Li et al.'s scheme and point out the potential security flaws. In addition, we demonstrate that user impersonation attack and server masquerade attack are effective on their scheme under the situation of smart card breach. Therefore, the countermeasures are presented as an enhanced scheme to overcome the shortcomings. Our enhancement is more secure, robust and practical, while costing more computation. At last, balancing the tradeoff between security and performance is the motivation in our further research.

Acknowledgement

This work is supported by NSFC (Grant Nos. 61300181, 61272057, 61202434, 61170270, 61100203, 61121061), the Fundamental Research Funds for the Central Universities (Grant No. 2012RC0612, 2011YB01), China Postdoctoral Science Foundation (Grant No. 2013M530561).
The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] L. Lamport, Password authentication with insecure communication, Communications of the ACM **24(11)**, 770-772 (1981).

[2] M. Burrows, M. Abadi, R. Needham, A logic of authentication, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 426(1871): 233-271 (1989).

[3] L.H. Li, I.C. Lin, M.S. Hwang, A remote password authentication scheme for multi-server architecture using neural networks, IEEE Transactions on Neural Networks **12(6)**, 1498-1504 (2001).

[4] W.S. Juang, Efficient multi-server password authenticated key agreement using smart cards, IEEE Transactions on Consumer Electronics **50(1)**, 251-255 (2004).

[5] C.C. Chang, J.S. Lee, An efficient and secure multi-server password authentication protocol using smart cards, In Proceedings of the Third International Conference on Cyberworlds **2004**, 417-422 (2004).

[6] W.J. Tsaur, C.C. Wu, W.B. Lee, A smart card-based remote scheme for password authentication in multi-server Internet services, Computer Standards & Interfaces **27(1)** 39-51 (2004).

[7] J.L. Tsai, Efficient multi-server authentication scheme based on one-way hash function without verification table, Computers & Security **27(3-4)**, 115-121 (2008).

[8] K.H. Yeh, N.W. Lo, Y. Li, Cryptanalysis of Hsiang-Shih's authentication scheme for multi-server architecture, International Journal of Communication Systems **24(7)**, 829-836 (2010).

[9] S.K. Sood, A.K. Sarje, K. Singh, A secure dynamic identity based authentication protocol for multi-server architecture, Journal of Network and Computer Applications **34(2)**, 609-618 (2011).

[10] X. Li, Y.P. Xiong, J. Ma, W.D. Wang, An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards, Journal of Network and Computer Applications **35(2)**, 763-769 (2012).

[11] Y.P. Liao, S.S. Wang, A secure dynamic ID based remote user authentication scheme for multi-server environment, Computer Standards & Interfaces **31(1)**, 24-29 (2009).

[12] H.C. Hsiang, W.K. Shih, Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment, Computer Standards & Interfaces **31(6)**, 1118-1123 (2009).

[13] C.C. Lee, T.H. Lin, R.X. Chang, A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards, Expert Systems with Applications **38(11)**, 13863-13870 (2011).

[14] X. Li, J. Ma, W.D. Wang, Y.P. Xiong, J.S. Zhang, A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments, Mathematical and Computer Modelling **58(1-2)**, 85-95 (2012). DOI:10.1016/j.mcm.2012.06.033.

[15] P. Kocher, J. Jaffe, B. Jun, Differential Power Analysis, Advances in Cryptology-CRYPTO'99, LNCS **1666**, 388-397 (1999).

[16] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Transactions on Computers **51(5)**, 541-552 (2002).

[17] X.F. Leng, Smart card application and security, Information Security Technical Report **14(2)**, 36-45 (2009).

[18] X. Huang, Y. Xiang, A Chonka, A generic framework for three-factor authentication: preserving security and privacy in distributed systems, IEEE Transactins on Parallel and Distributed Systems **22(8)**, 1390-1397 (2011).

[19] R. Song, Advanced smart card based password authentication protocol, Computer Standards & Interfaces **32(5)**, 321-325 (2010).

[20] J. Xu, W.T. Zhu, D.G. Feng, An improved smart card based password authentication scheme with provable security, Computer Standards & Interfaces **31(4)**, 723-728 (2009).

[21] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, Journal of Network and Computer Applications **36(5)**, 1365-1371 (2013).

[22] C.T. Li, A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card, IET Information Security **7(1)**, 3-10 (2013).

[23] C.T. Li, M.S. Hwang, An efficient biometrics-based remote user authentication scheme using smart cards, Journal of Network and Computer Applications **33(1)**, 1-5 (2010).

[24] F. Wen, W. Susilo, G. Yang, A robust smart card-based anonymous user authentication protocol for wireless communications, Security and Communication Networks (2013) doi: 10.1002/sec.816.

[25] X. Li, F. Wen, S. Cui, A strong password-based remote mutual authentication with key agreement scheme on elliptic curve cryptosystem for portable devices, Appl. Math **6(2)**, 217-222 (2012).

[26] A.K. Awasthi, K. Srivastava, A Biometric Authentication Scheme for Telecare Medicine Information Systems with Nonce, Journal of medical systems **37(5)**, 1-4 (2013).

[27] Y. An, Improved Biometrics-Based Remote User Authentication Scheme with Session Key Agreement, Computer Applications for Graphics, Grid Computing, and Industrial Environment. Springer Berlin Heidelberg, 307-315 (2012).

[28] L. Chen, P. Morrissey, N.P. Smart, et al., Security notions and generic constructions for client puzzles, Advances in CryptologyCASIACRYPT 2009, Springer Berlin Heidelberg, 505-523 (2009).

[29] L. Kuppusamy, J. Rangasamy, D. Stebila, et al., Practical client puzzles in the standard model, Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ACM, 42-43 (2012).

[30] L. Von Ahn, M. Blum, N.J. Hopper, et al., CAPTCHA: Using hard AI problems for security, Advances in CryptologyłEUROCRYPT 2003, Springer Berlin Heidelberg, 294-311 (2003).

[31] Y. Dodis, R. Ostrovsky, L. Reyzin, et al., Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, SIAM Journal on Computing, **38(1)**, 97-139 (2008).

[32] D. He, D. Wang, Robust Biometrics-Based Authentication Scheme for Multiserver Environment, Systems Journal, IEEE, **99**: 1-8 (2013). DOI:10.1109/JSYST.2014.2301517

[33] M.C. Chuang, M. C. Chen, An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics, Expert Systems with Applications, **41(4)**, 1411-1418 (2014).

[34] V. Odelu, A.K. Das, A. Goswami, Cryptanalysis on Robust Biometrics-Based Authentication Scheme for Multi-server Environment. http://eprint.iacr.org/2014/715.pdf

[35] D. Guo, F. Wen, Analysis and Improvement of a Robust Smart Card Based-Authentication Scheme for Multi-Server Architecture, Wireless Personal Communications, **78(1)**, 475-490 (2014).

[36] ML Das, A Saxena, VP Gulati, A dynamic ID-based remote user authentication scheme, Consumer Electronics, IEEE Transactions on, **50**, 629-631 (2004).

[37] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, An Enhancement of a Smart Card Authentication Scheme for Multi-server Architecture, Wireless Personal Communications, 1-18 (2014). DOI:10.1007/s11277-014-2002-x

[38] D. Mishra, A.K. Das, S. Mukhopadhyay, A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards, Expert Systems with Applications, **41(18)**, 8129-8143 (2014).

[39] IE Liao, CC Lee, MS Hwang, Security enhancement for a dynamic ID-based remote user authentication scheme, Next Generation Web Services Practices, 2005. NWeSP 2005. International Conference on, (2005)

Xuelei Li is a PhD candidate in State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He received the B.S. and M.S. degrees from Dezhou College in 2009 and University of Jinan in 2012, respectively. His interests inlude cryptography and information security.

Qiaoyan Wen received the B.S. and M.S. degrees from Shaanxi normal University in 1981 and 1984, respectively, and the Ph.D. degree from Xidian University in 1997. Now, she is a professor of Beijing University of Posts and Telecommunications. Her present research interests include cryptography and information security.

Wenmin Li received the B.S. and M.S. degrees in Mathematics and Applied Mathematics from Shaanxi Normal University, Xian, Shaanxi, China, in 2004 and 2007, respectively, and the Ph.D. degree in Cryptology from Beijing University of Posts and Telecommunications, Beijing, China, in 2012. She is currently a post-doctoral in Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include cryptography and information security.

Hua Zhang received the B.S. and M.S. degrees from Xidian University in 2002 and 2005, respectively, and the Ph.D. degree from Beijing University of Posts and Telecommunications in 2008. Now she is an associate professor of Beijing University of Posts and Telecommunications. Her research interests include cryptographic protocols, and security in IoT, cloud computing, industrial control system and Mobile Internet.

Zhengping Jin received the B.S. and M.S. degrees from Anhui Normal University in 2004 and 2007, respectively, and the Ph.D. degree from Beijing University of Posts and Telecommunications in 2010. Now he is a lecturer of Beijing University of Posts and Telecommunications. His research interests include design and analysis of cryptographic protocols, and security in IoT.