MAKA: Provably Secure Multi-factor Authenticated Key Agreement Protocol

Xiaoxue Liu[1], Yanping Li[1], Juan Qu[2], Qi Jiang[3,4]

[1] School of Mathematics and Information Science, Shaanxi Normal University, China
[2] School of Mathematics and Statistics, Chongqing Three Gorges University, China
[3] School of Cyber Engineering, Xidian University, China
[4] School of Computer & Software, Nanjing University of Information Science & Technology, China
862417756@qq.com, lyp@snnu.edu.cn, qulujuan@163.com, jiangqixdu@gmail.com

Abstract

Remote authentication is important to protect a networked server against malicious remote logins in complex systems, it is also the most efficient method to determine the identity of a remote user. Recently, Li et al. proposed an enhanced smart card based remote user password authentication scheme, referred to as LNKL scheme. In this paper, we first analyze LNKL scheme and show their scheme is vulnerable to key compromise impersonation attack and smart card impersonated attack. Besides, LNKL scheme does not provide user's anonymity and privacy protection. LNKL scheme still has some design flaws such as non-repairability. Furthermore, LNKL scheme adopts two-factor authentication (password and smart-card), which are easily compromised. Based on LNKL scheme and biometrics- based multi-factor authentication, an improved multi-factor authentication (short for MAKA) is proposed in this paper, which not only keeps the merits of LNKL scheme, but also achieves more security features. In addition, the MAKA protocol can be formally proved securely against passive and active attacks under the computational Diffie-Hellman problem assumption in the random oracle model. As a result, it is more well-suited for mobile application scenarios where resource is constrained and security is concerned.

Keywords: Multi-factor authentication, Biometrics, Random oracle model, Computational Diffie-Hellman problem (CDHP)

1 Introduction

With the rapid development of ubiquitous computing, most users access remote networks and get services. In order to obtain the trusted services, mutual authentication between the user and the server is the most important mechanism [1]. In distributed systems, single-factor and two-factor authentication are vulnerable to the simple dictionary attack [2]. Hence, multi-factor authentication becomes very important both in theory and in practice. Furthermore, with the development of e-commerce, e-banking, and online shopping, there is a growing demand to protect users' privacy. Currently, anonymity is one of the most important and common ways to preserve user privacy. Traditionally, authentication and anonymity, these two security goals contradict each other in some scenarios. Therefore, authentication protocols with privacy preserving have become a hot research topic.

The first password-based authentication scheme was given by Lamport [3]. Later on, a large number of designs of authenti- cation have been proposed [4-6]. To strengthen security, smartcard-based password authentication has become one of the most common authentication mechanisms. However, passwords might be divulged or forgotten, and smart cards might be shared, lost, or stolen. Compared with them, biometric keys cannot be lost or forgotten, copied or shared, and cannot be guessed easily [7]. Therefore, biometrics-based authentication schemes gain wide attention.

Authenticated key agreement (short for AKA) protocols have been extensively studied since they incorporate authentication and key agreement in one logical step. Any two parties could authenticate mutually and communicate confidentially in open network environment. Due to advantage of bioinformatics, biometrics-based AKA protocols are becoming one of the most widely deployed authentication mechanism [8-11].

A smartcard-based password authentication scheme proposed in [12] claimed it is secure against many known attacks. However, the scheme is proved to be vulnerable to off-line dictionary attack and forgery attack as noted in [13]. A valid but illegal user can extract data from the smart card and execute an impersonation attack. Then Song presented an new scheme [13]. But unfortunately, Chen et al. [14] found it is insecure against off-line dictionary attack,

*Corresponding Author: Yanping Li; E-mail: lyp@snnu.edu.cn
DOI: 10.3966/160792642018051903005

impersonation attack and proposed a new improved scheme. However, in 2013, Li et al. [15] found Chen et al.'s scheme cannot really ensure forward security, and the password change phase is unfriendly and inefficient. Further Li et al. put forward an enhanced AKA scheme (short for LNKL scheme lately). They claimed that their scheme can resist many attacks. Unfortunately, after careful analysis, we found that LNKL scheme still has some security flaws. It is vulnerable to impersonation attack. It also cannot protect users' anonymity and scheme's reparability. In order to overcome the weaknesses of LNKL scheme, a novel provably secure multi-factor authentication key agreement protocol (abbreviation for MAKA) is presented in this paper, which not only inherits the merits of LNKL scheme, but also has the following advantages.

- First, the MAKA protocol can provide multi-factor authentication: the smart card (something the user has), password (something the user knows) and bioinformatics (something the user is). Bioinformatics is believed to be a reliable authentication factor since it provides a potential source of high entropy information and cannot be easily lost, forgotten and faked [7]. The unique bioinformatics is used to activate the smart card. Only the password and bioinformatics both are correct, then the smart card can be activated to help users authentication.
- Second, the MAKA protocol is proven secure in the random oracle model and it can withstand cryptanalytic attacks under the hardness assumption of **CDHP**, over a finite cyclic group. The **CDHP** is one of classical hard problems in cryptology, whose difficulty can be reduced to the discrete logarithm problem (**DLP**). Its computational difficulty is more stable than other derived hard problems.
- Third, the MAKA protocol allows user to register with anonymous ID_i to preserve the user's privacy. In order to prevent the adversary to track the behavior of the user with identity ID_i, the whole process of AKA between the user and server adopts a dynamic blind identity CID_i.

The rest of this paper is organized as follows. Section 2 briefly reviews LNKL scheme and the weaknesses of LNKL scheme is analyzed in Section 3. The proposed MAKA protocol is presented in Section 4. Detailed security analysis and proof are given in Section 5. The comparisons of the performance and security features between our MAKA with other related schemes are shown in Section 6. Section 7 concludes this paper.

2 Review of LNKL Scheme

LNKL scheme is composed of **Registration, Login, Authentication, Password change** and **User revoking phase** [15]. To simplify the subsequent description, some notations are given in Table 1. Initially, the authentication server S selects the large prime p and q such that $p = 2q+1$, chooses master secret key $x \in Z_q^*$ and a cryptographically secure one-way hash function $h:\{0,1\}^* \rightarrow Z_q^*$. LNKL scheme is briefly reviewed as follows.

Table 1. Notations

Symbol	Description
$E_k(\cdot)/D_k(\cdot)$	Symmetric en/decryption functions with key k
ΔT	The maximum transmission delay
$\oplus$	The bitwise XOR operation
$\|\|$	The string concatenation operation
$\rightarrow$	A common communication channel
$\Rightarrow$	A secure communication channel

2.1 Registration Phase

R1 U_i chooses the identity ID_i and the password PW_i.

Then, $U_i \Rightarrow S: ID_i, PW_i$;

R2 On receiving the registration request, S computes

$A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$, $B_i = h(ID_i)^{x+PW_i} \bmod p$;

R3 S stores $\{A_i, B_i, p, q, h(\cdot)\}$ into a smart card. Then, $S \Rightarrow U_i$: Smart card.

2.2 Login Phase

L1 U_i inserts his/her smart card into the card reader, and inputs his/her ID_i, PW_i;

L2 The smart card computes $A_i ?= h(ID_i \| PW_i)^{PW_i} \bmod p$. If it does not match, the session is terminated [16]. Otherwise;

L3 The smart card generates a random number $\alpha \in Z_q^*$ and computes $C_i = B_i / h(ID_i)^{PW_i} \bmod p$, $D_i = h(ID_i)^{\alpha} \bmod p$,

$M_i = h(ID_i \| C_i \| D_i \| T_i)$, where T_i is the current timestamp. Then $U_i \rightarrow S: ID_i, D_i, M_i, T_i$;

2.3 Authentication Phase

V1 S checks validity of ID_i and insures that $T_i^{'} - T_i \le \Delta T$, where $T_i^{'}$ is the current time of S. If both of them are invalid, the login request is rejected. Otherwise, S computes $C_i = h(ID_i)^x \bmod p$, checks $M_i ?= h(ID_i \| C_i \| D_i \| T_i)$. If it does not match, S terminates the request. Otherwise ;

V2 S generates a random number $\beta \in Z_q^*$, computes

$V_i = h(ID_i)^{\beta} \bmod p$, session key $sk = D_i^{\beta} \bmod p$ and $M_S = h(ID_i \| C_i \| V_i \| sk \| T_S)$, where T_S is the current timestamp. Then, $S \to U_i : ID_i, V_i, M_S, T_S$;

V3 On receiving the message, U_i checks the validity of ID_i and T_S by $T_S^{'} - T_S \le \Delta T$, where $T_S^{'}$ is the current time of U_i, if any of them do not hold, U_i rejects. Otherwise ;

V4 U_i computes $sk = V_i^{\alpha} \bmod p$, checks $M_S ?= h(ID_i \| C_i \| V_i \| sk \| T_S)$; if it is not equal, the session is terminated.

Otherwise, S is authenticated by U_i. At last, U_i and S share the session key $sk = h(ID_i)^{\alpha\beta} \bmod p$.

Due to **Password change phase** and **User revoking phase** have nothing with security analysis of LNKL scheme, they will not be covered again here. For more details, please refer to [15].

3 Cryptanalysis of LNKL Scheme

In this section, we will show that LNKL scheme cannot withstand key compromise impersonation attack. Moreover, LNKL scheme also cannot provide anonymity and reparability.

3.1 Key Compromise Impersonation (KCI) Attack

Suppose the long-term private key x of server S is leaked out by accident or intentionally stolen by an adversary A, A can succeed in impersonating U_i to spoof S. Assume A can obtain all messages transferred on the public communication channel (ID_i, D_i, M_i, T_i), (ID_i, V_i, M_S, T_S).

A chooses a random number $\alpha^* \in Z_q^*$ and computes $C_i = h(ID_i)^x \bmod p$, $D_i^* = h(ID_i)^{\alpha^*} \bmod p$, $M_i^* = h(ID_i \| C_i \| D_i^* \| T_i)$. Then, $A \to S : ID_i, D_i^*, M_i^*, T_i$;

S computes $C_i = h(ID_i)^x \bmod p$, believes M_i^* is U_i 's legal login request. S will accept A 's login request and send back a reply. Then the session key will be built between S and A. Hence, LNKL scheme cannot resist the KCI attack.

3.2 Smart Card Impersonated (SCI) Attack

Suppose A intercepts U_i 's ID_i from **L1** and **V2** since both of them are common communication channels, A can enroll in S by using $PW_{\mathcal{A}}$ which is randomly chosen by A. If U_i did never register at S, A can easily get a smart card in the name of U_i. A computes $A_{\mathcal{A}} = h(ID_i \| PW_{\mathcal{A}})^{PW_{\mathcal{A}}} \bmod p$. Finally, A gets $C_{\mathcal{A}} = B_{\mathcal{A}} / h(ID_i)^{PW_{\mathcal{A}}} \bmod p$. $C_{\mathcal{A}} = C_i$ exactly is the most important information, which can help A to prove himself being U_i. Hence, LNKL scheme also is vulnerable to the SCI attack.

3.3 Non-anonymity

In LNKL scheme, the user's identity ID_i is static in whole phases, which can easily leak U_i 's login history，his preference, hobbies, interest，and even U_i 's real identity. Hence, user's anonymity is not preserved.

3.4 Non-reparability

As above described in 3.1 and 3.2, there are many ways that may leak C_i. Even if U_i changes his/her password, C_i is static and unchanged. Hence, impersonation attacks cannot be instantly prevented. As C_i is determined only by U_i 's ID_i and S 's secret key x, S cannot change C_i for U_i unless ID_i or x can be changed. However, it is also inefficient to change ID_i, which may be tied to U_i in most application systems. Additionally, since x is commonly used for all users rather than specifically used only for U_i, it is unreasonable and impractical if x is changed to protect the U_i 's security. Thus, the LNKL scheme is not easy to repair [17-18].

4 Our MAKA Protocol

To overcome the afored-discussed security flaws of LNKL scheme, an improved protocol is proposed as shown in Figure 1. Initially, the authentication sever S generates parameters p, q and g like LNKL scheme. S chooses two secure one-way hash functions $h_i : \{0,1\}^* \to \{0,1\}^l (i = 1,2)$ and its secret key $sk = x \in Z_q^*$, then computes the corresponding public key $pk = g^x$.

Here, the fuzzy extractor which would be used in our paper is briefly introduced. It is consisted by two procedures: the probabilistic generation procedure (*Gen*) and the deterministic reproduction procedure (*Rep*) [19].

- *Gen*: On input biometric data ω, *Gen* outputs an extracted string σ and a public auxiliary string θ, where $<\sigma, \theta> = Gen(\omega)$ with $|\sigma| = l$.
- *Rep* can recover σ from the auxiliary string θ and any vector $\omega^{'}$, which is close to ω. For all $\omega, \omega^{'}$ satisfying $dis(\omega, \omega^{'}) \le \lambda$, ($\lambda$ is the tolerance threshold), if $<\sigma, \theta> = Gen(\omega)$, then *Rep* $(\omega^{'}, \theta) = \sigma$. Compared with low-entropy password, the probability to guess the biometric key σ by A is about $\frac{1}{2^l}$, $l = \gamma + 2\log(\varepsilon) + O(1)$, γ is min-

entropy, $\varepsilon \geq 0$ is constant) [19].

4.1 Registration Phase

R1 U_i gets his/her bioinformatics b_i by special equipment, $|b_i| = l$, $Gen(b_i) = <\sigma_i, \theta_i>$, chooses his/her ID_i, PW_i, and computes $h_1(\sigma_i \| PW_i)$. Then, $U_i \Rightarrow S: ID_i, h_1(\sigma_i \| PW_i), \sigma_i$;

R2 On receiving the registration message from U_i, S maintains an account for U_i, which records the U_i's ID_i, biometric key σ_i and registration number T ($T=1$ if it is the first time to registration, or else, $T := T + sl$, where sl is a step length chosen by S). S computes s_i, A_i, B_i and stores $\{A_i, B_i, p, q, h_1(\cdot), h_2(\cdot)\}$ in smart card. Then, $S \Rightarrow U_i$: Smart card;

R3 U_i adds θ_i to the smart card. Finally $\{A_i, B_i, \theta_i, p, q, \ h_1(\cdot), h_2(\cdot)\}$ are stored in U_i's smart card, see Figure 1.

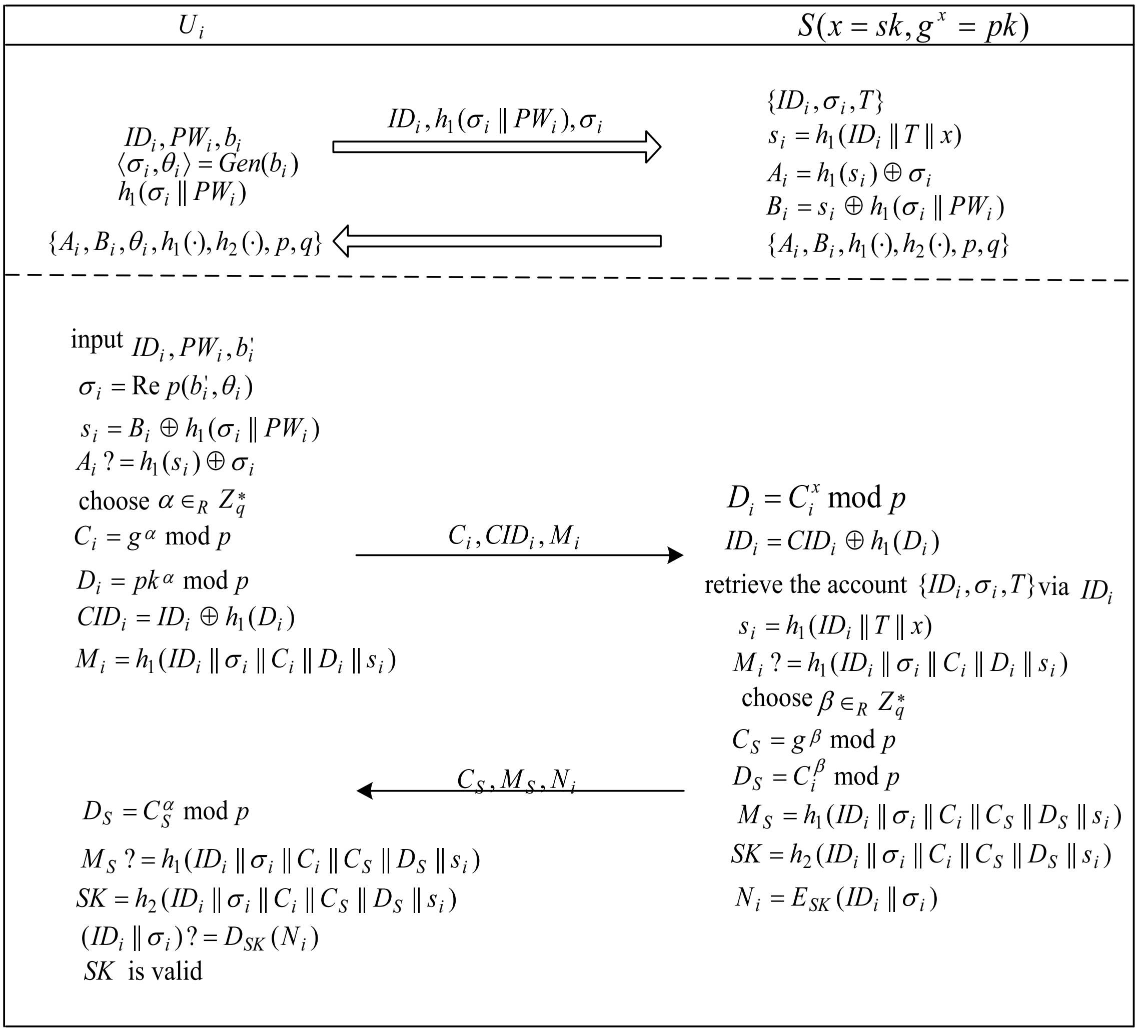

Figure 1. The flow chart of our MAKA protocol

4.2 Login Phase

L1 When U_i logins to S, U_i inserts his/her smart card into the card reader, inputs his/her ID_i, PW_i, and his/her bioinformatics $b_i^{'}$ read by special equipment;

L2 The smart card computes σ_i, s_i as Figure 1, and checks $A_i ?= \ h_1(s_i) \oplus \sigma_i$. If it does not match, the session is terminated. Or else, the smart card generates a random number $\alpha \in Z_q^*$, and computes C_i, D_i, CID_i, M_i. Then, $U_i \rightarrow S: C_i, D_i, CID_i, M_i$.

4.3 Authentication Phase

V1 On receiving the login request from U_i, S computes D_i, ID_i as Figure 1. Then S checks whether ID_i is valid according to the account $\{ID_i, \sigma_i, T\}$. If not, the login request is rejected. Otherwise;

V2 S computes s_i and checks $M_i ?= h_1(ID_i \| \sigma_i \| C_i \| D_i \| s_i)$. If not, the session is terminated. Or else, S chooses a random number $\beta \in Z_q^*$, computes C_S, D_S, M_S, SK, N_i as Figure 1. Then, $S \rightarrow U_i: C_S, M_S, N_i$;

V3 On receiving the message C_S, M_S, N_i, U_i computes D_S, checks $M_S ?= h_1(ID_i \| \sigma_i \| C_i \| C_S \| D_S \| s_i)$. If it does not match, terminates the session. Otherwise, U_i computes SK, checks $(ID_i \| \sigma_i)?=D_{SK}(N_i)$. If not, terminates the session. Otherwise, SK is valid.

4.4 Password Change Phase

C1 The algorithm is invoked whenever U_i wants to change the old PW_i to the new PW_i^{new}. U_i inserts his/her smart card into card reader and inputs ID_i, PW_i, b_i^*;

C2 The smart card computes σ_i and s_i as Figure 1, checks $A_i ?= h_1(s_i) \oplus \sigma_i$. If not match, the session is terminated; Or else, it computes $B_i^{new} = B_i \oplus h_1(\sigma_i \| PW_i) \oplus h_1(\sigma_i \| PW_i^{new})$ according to the new PW_i^{new} and updates B_i with B_i^{new}.

4.5 Smart Card Revocation Phase

If U_i lost his/her smart card, he can ask S to issue a new smart card after checking the validity of U_i's ID_i and σ_i. The process is the same as the registration process, and lets T be $T + sl$. Thus U_i gets a new smart card.

Additionally, no one can activate the smart card without the correct bioinformatics except for the legal smart card owner. So the lost smart card is useless for anybody else.

5 Security Proof of the MAKA Protocol

The hypothesis of the adversary A are given as follows.

- A can eavesdrop, intercept, delete, and modify all messages of the common communication channel;
- A can extract the secret information stored in the smart card, but cannot get the password synchronously [12].
- A cannot get the server's private key x and the U_i's account $\{ID_i, \sigma_i, T\}$ synchronously [12].

5.1 Security Analysis

User anonymity (UA). MAKA adopts a dynamic anonymous blind identity CID_i instead of the static identity ID_i in the common channel. So anybody outside the system cannot get the user's ID_i by tracking CID_i. Suppose that A has intercepted U_i's authentication messages (C_i, CID_i, M_i), (C_S, M_S, N_i). And A recovers $ID_i = CID_i \oplus h_1(D_i)$, if and only if he gets $h_1(D_i)$, which means A has to solve the **CDHP** of (C_i, pk). Additionally, both of above messages are random values relied on the random number α or β. A cannot distinguish the correlation of two messages [20]. Thus, it realizes untraceability of the session. Hence, MAKA realizes the anonymity and avoids the user to be traced.

Perfect forward secrecy (PFS). In MAKA, $SK = h_2(ID_i \| \sigma_i \| C_i \| C_S \| D_S \| s_i)$ is the session key shared between U_i and S, wherein $C_i = g^{\alpha} \bmod p$, $C_S = g^{\beta} \bmod p$, $D_S = g^{\alpha\beta} \bmod p$, α and β are random numbers chosen by U_i and S respectively, which are different in each session run. SK is hash value which cannot disclose any information. Therefore, A cannot infer any valuable information about the forward and backward session keys even if he gets the current session key.

Resistance to key compromise impersonation (KCI). Suppose that the server secret key x is leaked out by accident or intentionally stolen by A, but U_i's account $\{ID_i, \sigma_i, T\}$ is kept secret. Since A always does not know ID_i and T, A cannot compute the secret value $s_i = h_1(ID_i \| T \| x)$ and M_i. So A cannot impersonate U_i to spoof S. Suppose that U_i's password PW_i is leaked out, but A does not know the U_i's biometric key σ_i, and A cannot activate the smart card. Hence, A cannot successfully impersonate U_i.

Known key (KK) attack. In MAKA, all session keys are independent since each key depends on random numbers $\alpha, \beta \in Z_q^*$. A cannot compute other session keys from the current session key. So the MAKA protocol can resist **KK** attack.

Verification account stolen (VAS) attack. Verification account stolen attack denotes that A obtains the U_i's verification account and guesses the U_i's password PW_i, then launches impersonation attack [21]. In MAKA, even if A gets a registration account $\{ID_i, \sigma_i, T\}$, he is still unable to compute parameter $s_i = h_1(ID_i \| T \| x)$ without knowing S's secret key x. And nor can A compute the legitimate authentication item (M_i, M_S, N_i). Therefore, the MAKA is secure against **VAS** attack.

Password guessing (PG) attack. Suppose A can get $\{A_i, B_i, \theta_i, p, q, h_1(\cdot), h_2(\cdot)\}$ from the smart card. However, only $B_i = h_1(ID_i \| T \| x) \oplus h_1(\sigma_i \| PW_i)$ is related to PW_i. A want to guess a PW^* and verify $B_i ?= h_1(ID_i \| T \| x) \oplus h_1(\sigma_i \| PW^*)$. But A does not know (x, ID_i, T) and σ_i. Hence, A is unable to check above equation, further cannot verify the correctness of PW^*. Hence, the MAKA can resist **PG** attack.

Smart card impersonated (SCI) attack. If A registers at the server S in the name of U_i, A must provide S with his/her unique bioinformatics. If U_i found that he has been registered at S, he can make a complaint. And the A 's unique bioinformatics. will be added to the blacklist and A can be traced when necessary. Based on this risk, A is unwilling to do it.

5.2 Security Model and Notations

Firstly, the formal security model for password-based AKA protocols with smart card is described, which is mainly develop-ed from Bellare [22]. Then, a security proof for the MAKA protocol is given under the hardness assumption of **CDHP**.

Participants and initialization. In AKA scheme, each participant is either a user $U_i \in Users$ or server $S \in Servers$. Each participant is modeled as a set of random oracles. Each oracle can be independent and executed concurrently. S holds a secret key sk. Each user U_i chooses a password PW_{U_i} from the dictionary $\mathcal{D}$.

Execution of the protocol. $\mathcal{C}$ is a simulator who simulates the protocol for $\mathcal{A}$. The interaction between $\mathcal{A}$ and $\mathcal{C}$ occurs only via oracle queries, which simulate the adversary capabilities in a real attack.

Execute (U_i, S): This oracle query is used to simulate $\mathcal{A}$'s passive eavesdropping attack. Its output consists of the messages that were exchanged between U_i and S during the real execution of the protocol.

Send $(U_i / S, m)$: This oracle simulates $\mathcal{A}$'s active attack. $\mathcal{A}$ sends a message m to U_i / S. U_i / S give response to m according to the protocol.

Reveal (U_i / S): This oracle query simulates the **KK attack**. It returns to $\mathcal{A}$ the session key negotiated by U_i and S. It helps $\mathcal{A}$ to judge whether two session keys are independent.

Corrupt $(U_i / S, a)$: The oracle query simulates corruption capabili-ties of $\mathcal{A}$. $\mathcal{A}$ can simulate the **KCI attacks**, **VAS attacks**, **PFS** etc. with this oracle.

- ***Corrupt*** (U_i, a), If a=1, it outputs the U_i's password PW_{U_i}; if a=2, it outputs the messages stored in the smart card;
- ***Corrupt*** (S, a), If $a = 1$, it outputs the S 's private key x; if $a = 2$, it outputs the account $\{ID_i, \sigma_i, T\}$.

Ephemeral key reveal (U_i, S): The oracle simulates the key leak attack, by which $\mathcal{A}$ can get U_i / S 's temporary secret information.

DDHP: The decision **DHP** oracle is to verify $(g^a, g^b)? = g^{ab}$, only $\mathcal{C}$ can query the oracle once in one session.

Test (U_i / S): This oracle query is to define semantic security of the session key and can be asked only once. After querying the oracle, a value will be returned according to a predefined random bit b. If $b = 1$, the adversary would get the session key shared by U_i and S, otherwise get a random value.

Security goals.

- ***Partner***(***Par***): We say that U_i and S are partnered if the following conditions are met: (1) Both U_i and S are accepted (it means the session key has been successfully negotiated between them); (2) Both U_i and S share the same session identification sid (which are sent and received by U_i and S in the protocol); (3) U_i's partner only is S and vice-versa, that is to say ***Par*** $S = U_i$ or ***Par*** $U_i = S$.
- ***Freshness***: Say U_i / S is fresh if the following conditions hold: (1) U_i / S has accepted and has session key SK; (2) U_i / S and its partner has been made no ***Reveal queries***. (3) U_i / S is asked ***Corrupt queries*** at most only once.
- ***Semantic security***: It is a significant goal of AKA protocols. During one session of the protocol P, $\mathcal{A}$ can make polynomial times with ***Execute, Send, Reveal, Corrupt queries***, a single ***Test query*** for some fresh instance that has been completed. The output of ***Test query*** is a bit b'. Then $\mathcal{C}$ compares b' with b that was selected in the Test query. If $b' = b$, we say $\mathcal{A}$ wins the game and ***Succ*** stands for this event. Accordingly, the $\mathcal{A}$'s advantages to destroy the semantic security of protocol P is

$$Adv_P^{aka}(\mathcal{A}) \overset{def}{=} 2\Pr[Succ(\mathcal{A})] - 1 = 2\Pr[b' = b] - 1$$

5.3 Security Proof

Theorem 1. Let G be a finite cyclic group and let $\mathcal{D}$ be a uniformly distributed dictionary of size $|\mathcal{D}|$. Let $\mathcal{A}$ be an adversary against the semantic security with time bound t, with less than q_s sessions, q_d ***Send queries***, q_e ***Execution queries***, and q_h ***Hash oracle queries***. Then we have

$$Adv_P^{aka}(\mathcal{A}) \le \frac{q_h^2}{2^{l+1}} + \frac{(q_d + q_e)^2}{2q} + 4\frac{q_d}{2^l} + \frac{q_d}{|\mathcal{D}|}$$
$$+ q_s^2 q_h Adv_G^{DLP}(t) + q_s q_h Adv_G^{CDH}(t + (q_d + q_e)t_e)$$

where t_e demotes the exponentiation computation time in G.

Proof. The main idea is that if $\mathcal{A}$ destroyed semantic security of the protocol successfully, $\mathcal{C}$ can solve the **CDHP** by $\mathcal{A}$'s answers. Our proof defines a sequence of hybrid games, starting with the real attack and ending with a game in which $\mathcal{A}$ has no advantage. For each $Game_n$, we define events $Succ_n$ corresponding the case in which $\mathcal{A}$ correctly guesses the bit b involved in the ***Test query***. $AskPara_n$ denotes $\mathcal{A}$

successfully computes the secret value s_i by h_1 queries with $(\sigma_i \| PW_i)$ or $(ID_i \| T \| x)$, and $AskH_n$ denotes $\mathcal{A}$ successfully computes the secret value s_i and queries $h_i(\cdot)(i=1,2)$ with ID_i, σ_i, C_i, D_i, C_S, D_S, s_i.

***Game*₀**: This game is the real attack without any limits, where $Succ_0 = \{b' = b\}$. By the definition, we have

$$Adv_P^{aka}(\mathcal{A}) = 2\Pr[Succ_0] - 1 \quad (1)$$

***Game*₁**: In this game, $\mathcal{C}$ simulates the random oracle h_1 and keeps a hash list $L_h=(i,m,n)$ which is empty at the beginning. h_2 only be used once to generate the session key at the end of P. When $\mathcal{A}$ initiates a query m, the same answer n from the list L_h will be given if the request has been asked before. Otherwise, $\mathcal{C}$ chooses $n \in_R \{0,1\}^l$, and returns n as answer, adds this new record (i,m,n) to L_h, where i is the query time, m is the content set, n is the corresponding answer set.

The ***Execute, Reveal, Send, Corrupt, Test*** oracles are also simulated as real attack. Compared with ***Game*₀**, $\mathcal{C}$ just makes the relevant records in ***Game*₁**, it can easily see that this game is completely indistinguishable from the real game. Hence,

$$\Pr[Succ_1] - \Pr[Succ_0] = 0 \quad (2)$$

***Game*₂**: In this game, $\mathcal{C}$ simulates all oracles as in ***Game*₁**. All the executions will be terminated if a collision occurs. According to the birthday paradox, the collision probability in the output of h oracle is at most $\frac{q_h^2}{2^{l+1}}$. Similarly, the collision probability of the messages (C_i, CID_i, M_i), (C_S, M_S, N_i) is at most $\frac{(q_d+q_e)^2}{2q}$. Hence

$$\Pr[Succ_2] - \Pr[Succ_1] \le \frac{q_h^2}{2^{l+1}} + \frac{(q_d+q_e)^2}{2q} \quad (3)$$

Game*₃**: The executions are finished if $\mathcal{A}$ luckily guesses the authentication values (M_i, M_S) without the ***hash query. The ***Game*₂** has removed the collision possibility, and $\mathcal{A}$ guessed the value is exactly the original value with the probability $\frac{q_d}{2^l}$. Hence, ***Game*₃** and ***Game*₂** are indistinguishable, so

$$\Pr[Succ_3] - \Pr[Succ_2] \le \frac{q_d}{2^l} \quad (4)$$

Game*₄**: The executions are halted if $\mathcal{A}$ luckily guessed s_i without the ***hash query and spoofed the U_i or S successfully. Hence, ***Game*₄** and ***Game*₃** are indistinguishable, therefore

$$\Pr[Succ_4] - \Pr[Succ_3] \le \frac{q_d}{2^l} \quad (5)$$

Game*₅**: In this game, the executions are terminated if $\mathcal{A}$ computes the secret parameter s_i by querying ***hash query with $(\sigma_i \| PW_i)$ or $(ID_i \| T \| x)$. If the event $AskPara_5$ does not happen, ***Game*₅** and ***Game*₄** are indistinguishable. Hence,

$$\Pr[Succ_5] - \Pr[Succ_4] \le \Pr[AskPara_5] \quad (6)$$

From the security model, $\mathcal{A}$ can query both ***Corrupt*** (U_i, a) and ***Corrupt*** (S, a). Hence, it is easy to get that:

$$\begin{aligned}
&\Pr[AskPara_5] = \\
&\Pr[AskPara_5WithCorrupt(U_i,1)] \\
&+\Pr[AskPara_5WithCorrupt(U_i,2)] \\
&+\Pr[AskPara_5WithCorrupt(S,1)] \\
&+\Pr[AskPara_5WithCorrupt(S,2)] \\
&=\Pr[*\|PW_i] + \Pr[\sigma_i\|*] + \Pr[ID_i\|*\|PW_i] + \Pr[ID_i\|*\|x] \\
&\le \frac{q_d}{2^l} + \frac{q_d}{|\mathcal{D}|} + \frac{q_d}{2^l} + q_s^2 q_h Adv_G^{DLP}(t) = 2\frac{q_d}{2^l} + \frac{q_d}{|\mathcal{D}|} + q_s^2 q_h Adv_G^{DLP}(t)
\end{aligned}$$

***Game*₆**: The executions are ended if $\mathcal{A}$ calculates the values (M_i, M_S, SK) in this game. If the event $AskH_6$ occurs, A queries the hash function with $(ID_i \| \sigma_i \| C_i \| C_S \| D_S \| s_i)$, where **CDHP** $(C_i, C_S) = D_S$. $\mathcal{C}$ chooses one session of $[1,2,\cdots,q_s]$ as the test session and inserts the **CDHP** parameters. Then $\mathcal{C}$ can use A to solve the **CDHP**.

In non-test sessions, when A queries hash h with (M_i, M_S). Whenever receiving such query, $\mathcal{C}$ checks the list L_h and the same answer will be given if the request has been asked before. Otherwise, $\mathcal{C}$ uses **DDHP** oracle to verify $D_S ?= (C_i, C_S)$, if it is not equal, return $\perp$, or else a random number $\in Z_q^*$ to $\mathcal{A}$ because $\mathcal{C}$ does not know α and β in (M_i, M_S). ***In test session,*** $\mathcal{C}$ chooses g^a and g^b randomly, lets $C_i = g^a$, $C_S = g^b$. $\mathcal{A}$ queries the hash function on $(ID_i \| \sigma_i \| C_i \| C_S \| g^{ab} \| s_i)$, where g^{ab} = **CDHP** (C_i, C_S). If the event $AskH_6$ does not happen, ***Game*₆** and ***Game*₅** are indistinguishable. Hence,

$$\Pr[Succ_6] - \Pr[Succ_5] \le \Pr[AskH_6] \quad (7)$$

and $\Pr[AskH_6] \le q_s q_h Adv_G^{CDH}(t + (q_d+q_e)t_e)$

$\Pr[Succ_6] = \frac{1}{2}$

Hence, it is easy to conclude that:

$$\begin{aligned}
&Adv_P^{aka}(\mathcal{A}) = 2\Pr[Succ_0] - 1 \\
&= 2\Pr[Succ_6] - 1 + 2(\Pr[Succ_0] - \Pr[Succ_6]) \\
&\le 2\{|\Pr[Succ_1] - \Pr[Succ_0]| + |\Pr[Succ_2] - \Pr[Succ_1]|
\end{aligned}$$

$$+|\Pr[Succ_3]-\Pr[Succ_2]|+|\Pr[Succ_4]-\Pr[Succ_3]|$$
$$+|\Pr[Succ_5]-\Pr[Succ_4]|+|\Pr[Succ_6]-\Pr[Succ_5]|\}$$
$$\leq \frac{q_h^2}{2^{l+1}}+\frac{(q_d+q_e)^2}{2q}+4\frac{q_d}{2^l}+\frac{q_d}{|\mathcal{D}|}+q_s^2 q_h Adv_G^{DLP}(t)$$
$$+q_s q_h Adv_G^{CDH}(t+(q_d+q_e)t_e). \quad \square$$

6 Performance Evaluation

The performance and security of the MAKA protocol and other related protocols are given in Table 2 and Table 3.

Table 2. Performance comparison among related protocols

	[12]	[13]	[14]	[15]	Ours
t_{cc}	$5t_e$	t_e+2t_s	$6t_e+2t_m$	$11t_e+t_m$	$6t_e+2t_m$
cc	2816	896	896	1024	768
sc	3200	256	384	640	896

Table 3. Security comparison among related protocols

	[12]	[13]	[14]	[15]	Ours
UA	No	No	No	No	Yes
FSS	Yes	No	Yes	Yes	Yes
SAK	No	Yes	Yes	Yes	Yes
Reparability	No	No	No	No	Yes
PG	Yes	No	No	Yes	Yes
VAS	Yes	Yes	No	No	Yes
KCI	No	Yes	No	No	Yes
KK	Yes	Yes	Yes	Yes	Yes
PS	Yes	No	No	No	Yes

Let t_e, t_m, t_s be the time complexity for exponential operation, multiplication operation, symmetric key encryption /decryption operation, respectively. The time of " $h(\cdot)$ ", "||" and " $\oplus$ " are negligible as compared with the other time-consuming operations [19]. An efficient AKA protocol must take total computation cost(t_{cc}), communication cost (*cc*/bit) and storage cost (*sc*/bit) into consideration. We mainly focus on the efficiency of login and authentication phases since these two phases are the main body of an AKA protocol and are executed much more frequently.

7 Conclusion

System security and user privacy-preserved are challenging issues in distributed authentication systems. The MAKA protocol investigates a systematic approach for authentication and key agreement by multi-factors: password, smart card, bioinformatics. Meanwhile our MAKA protocol is proven secure in the random oracle model under the hardness assumption of **CDHP**. The MAKA protocol not only realizes anonymity to protect user's privacy, but also addresses the error-tolerance issues of bioinformatics. Compared with the recently relevant protocols, the MAKA protocol has better performance and better security features. The future work is to develop concrete multi-factor AKA protocols in multi-server environment with better performance.

Acknowledgement

This work was partly supported by the Natural Science Foundation of China under Grant 61402275, 61402015, 61572246, 615723036, Shaanxi Province Natural Science Basic Research Program Funded Project 2016JM6069, 2016JM6005, the Scientific Research Foundation for the Returned Overseas Chinese Scholars of MOHRSS, the Fundamental Research Funds for the Central Universities under Grant GK201803005, GK201402004, the Innovation Fund Designated for Graduate Students of Shaanxi Normal University (2015CXS022).

References

[1] B. L. Chen, W. C. Kuo, L. C. Wuu, A Secure Password-based Remote User Authentication Scheme Without Smart Cards, *Information Technology and Control*, Vol. 41, No. 1, pp. 53-59, March, 2012.

[2] D. V. Klein, Foiling the Cracker: A Survey of, and Improvements to, Password Security, *Proceedings of the 2nd USENIX Security Workshop*, Portland, OR, 1990, pp. 5-14.

[3] L. Lamport, Password Authentication with Insecure Communication, *Communications of the ACM*, Vol. 21, No. 11, pp. 770-772, November, 1981.

[4] D. He, S. Zeadally, Authentication Protocol for Ambient Assisted Living System, *IEEE Communications Magazine*, Vol. 53, No. 1, pp. 71-77, January, 2015.

[5] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, S.-S. Yeo, Robust Anonymous Authentication Protocol for Health-care Applications Using Wireless Medical Sensor Networks, *Multimedia Systems*, Vol. 21, No. 1, pp. 49-60, February, 2015.

[6] J.-L. Tsai, N.-W. Lo., T.-C. Wu, Novel Anonymous Authentication Scheme Using Smart Cards, *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 4, pp. 2004-2013, November, 2013.

[7] N. Zhang, Y.-L. Zang, J. Tian, The Integration of Biometrics and Cryptography: A New Solution for Secure Identity Authentication, *Journal of Cryptologic Research*, Vol. 2, No. 2, pp. 159- 176, April, 2015.

[8] C.-H. Chou, K.-Y. Tsai, T.-C. Wu, Robust Remote Mutual Authentication Scheme with Key Agreement, *Journal of Internet Technology*, Vol. 16, No. 7, pp. 1283-1289, December, 2015.

[9] R. Guo, Q. Wen, Z. Jin, H. Zhang, J. Zhao, An Efficient and Provably-Secure Broadcast Authentication Scheme in Wireless Sensor Networks, *Journal of Internet Technology*,

Vol. 16, No. 6, pp. 977-985, November, 2015.

[10] L. Yao, C. Lin, G. Wu, T. Jung, K. Yim, An Anonymous Authentication Scheme in Data-link Layer for VANETs, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 22, No. 1, pp. 1-13, May, 2016.

[11] D. Mishra, A. Chaturvedi, S. Mukhopadhyay, An Improved Biometric–based Remote User Authentication Scheme for Connected Healthcare, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol.18, No.1-2, pp. 75-84, March, 2015.

[12] J. Xu, W.-T. Zhu, D.-G. Feng, An Improved Smart Card Based Password Authentication Scheme with Provable Security, *Computer Standards & Interfaces*, Vol. 31, No. 4, pp. 723-728, June, 2009.

[13] R. Song, Advanced Smart Card Based Password Authentication Protocol, *Computer Standards & Interfaces*, Vol. 32, No. 5, pp. 321-325, October, 2010.

[14] B.-L. Chen, W.-C. Kuo, L.-C. Wuu, Robust Smart-card-based Remote User Password Authentication Scheme, *International Journal of Communication Systems*, Vol. 27, No. 2, pp. 377-389, February, 2014.

[15] X. Li, J. W. Niu, M. K. Khan, J. Liao, An Enhanced Smart Card Based Remote User Password Authentication Scheme, *Journal of Network & Computer Applications*, Vol. 36, No. 5, pp. 1365-1371, September, 2013.

[16] J. Bringer, H. Chabanne, T. A. M. Kevenaar, B. Kindarji, Extending Match-on-card to Local Biometric Identification, *Biometric ID Management and Multimodal Communication*, Madrid, Spain, 2009, pp. 178-186.

[17] R.-C. Wang, W.-S. Juang, C.-L. Lei, Robust Authentication and Key Agreement Scheme Preserving the Privacy of Secure Key, *Computer Communications*, Vol. 34, No. 3, pp. 274-280, March, 2011.

[18] H.-R. Chung, W.-C. Ku, M.-J. Tsaur, Weaknesses and Improvement of Wang et al.'s Remote User Password Authentication Scheme for Resource-limited Environments, *Computer Standards & Interfaces*, Vol. 31, No. 4, pp. 863-868, June, 2009.

[19] V. Odelu, A. K. Das, A. Goswami, A Secure Biometrics-based Multi-server Authentication Protocol Using Smart Cards, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 9, pp. 1953-1966, September, 2015.

[20] M. A. Pathak, B. Raj, S. D. Rane, P. Smaragdis, Privacy-preserving Speech Processing: Cryptographic and String-matching Frameworks Show Promise, *IEEE Signal Processing Magazine*, Vol. 30, No. 2, pp. 62-74, March, 2013.

[21] S. K. Sood, Secure Dynamic Identity-based Authentication Scheme Using Smart Cards, *Information Security Journal: A Global Perspective*, Vol. 20, No. 2, pp. 67-77, January, 2011.

[22] M. Bellare, D. Pointcheval, P. Rogaway, Authenticated Key Exchange Secure Against Dictionary Attacks, *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2000)*, Bruges, Belgium, 2000, pp. 139-155.

Biographies

Xiaoxue Liu received the B.S. degree from Bohai Univ. in 2014. She now is a M.S. degree candidate in Applied Mathematics with the Sc -hool of Maths. and inf. Sci., Shaanxi Normal Univ., Xi'an, China. Her research interests include security protocols and its analysis.

Yanping Li received the M.S. degree from Shaanxi Normal Univ. in 2004 and Ph. D. degree from Xidian Univ. in 2009, Xian, China. She now is an associate professor with the School of Maths. and Inf. Sci., Shaanxi Normal Univ.. Her research interests include public key cryptogra-phy and its applications.

Juan Qu received the M.S. degree in Applied Mathematics from Shaanxi Normal University in 2009. She currently is a lecture at School of Maths. and Stats., Chongqing Three Gorges Univ. Her research interests include security protocols and its security analysis.

Qi Jiang received the B.S. degree in Computer Sci. from Shaanxi Normal Univ. in 2005 and Ph.D. degree from Xidian Univ. in 2011. He is now an associate professor at School of Cyber Eng., Xidian Univ.. His research interests are wireless network and cloud security, etc.