Biometric-based Remote Mutual Authentication Scheme for Mobile Device

Sheng-Kai Chen · Jenq-Shiou Leu · Wen-Bin Hsieh · Jui-Tang Wang · Tian Song

Accepted: 28 May 2022 / Published online: 29 June 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Remote user authentication schemes provide a system to verify the legitimacy of remote users’ authentication request over insecure communication channel. In the past years, many authentication schemes using password and smart card have been proposed. However, password might be guessed, leaked or forgotten and smart card might be shared, lost or stolen. In contrast, the biometrics which utilize biological characteristics, such as face, fingerprint or iris, have no such weakness. With the trend of mobile payment, more and more applications of mobile payment use biometrics to replace password and smart card. In this paper, we propose a biometric-based remote authentication scheme substituting biometric and mobile device bounded by user for password and smart card. This scheme is more convenient, suitable and securer than the schemes using smart cards on mobile payment environment.

Keywords Mutual authentication · Biometric · Remote authentication scheme

Wen-Bin Hsieh
d9802106@mail.ntust.edu.tw
Sheng-Kai Chen
m10302154@mail.ntust.edu.tw
Jenq-Shiou Leu
jsleu@mail.ntust.edu.tw
Jui-Tang Wang
rtwang@mail.ntust.edu.tw
Tian Song
tiansong@ee.tokushima-u.ac.jp

1 Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
2 Tokushima University, Graduate School of Technology, Industrial and Social Sciences, Tokushima, Japan
1 Introduction

With the rapid development of wireless communication networks and e-commerce applications, such as e-banking, mobile payment and other transaction-oriented services, there is a growing demand for protecting user’s credentials privacy. In the recent decades, more and more transactions have been implemented on the internet or wireless network due to the portability property of mobile devices, such as laptops, tablet computers, smart phones and smart watches.

The traditional remote authentication schemes are based on passwords such as [1–3]. In order to enhance the security, some schemes add additional factors such as a smart card [4–6]. Due to reliability of biometric identification is superior to traditional password-based user authentication, biometric-based user authentication schemes, like [7] [8–12], are given rise.

More and more authentication systems use biological characteristics to be the key. The biometrics is the measurement and statistical analysis of people’s physical and behavioral characteristics. The biometrics of physiological characteristics are face, iris, fingerprint, ear, voice, palm print, retina, etc. The biometrics of behavioral characteristics are gait, signature, keystrokes, mouse use characteristics, etc. Compare with traditional secrets such as passwords, biometric-based secrets have many advantages. Several advantages are described as follows [12]:

- It is difficult to lose or forget biometric keys.
- It is difficult to copy or share biometric keys.
- It is difficult to forge or distribute biometrics.
- It is difficult to guess biometric keys.
- It is more difficult to break biometric keys.

Accordingly, biometrics-based authentication is inherently more reliable than traditional password-based authentication. In 2010, Li and Hwang [7] proposed a biometrics-based remote user authentication scheme using smart cards. In their scheme, they substituted nonce for the use of time synchronization. In 2011, Das et al. [8] analyzed Li and Hwang’s scheme and pointed out the security drawbacks. Subsequently, they proposed an efficient remote user authentication scheme to overcome the weaknesses of Li and Hwang’s scheme. In 2012, An et al. [9] analyzed Das et al.’s scheme and showed that it was still vulnerable to the various attacks and does not provide mutual authentication between the user and the server. An et al. proposed a scheme to remove the security problems found in Das et al.’s scheme. In 2013, Khan et al. [10] specified that An et al.’s scheme is vulnerable to the security problems to which Das et al.’s scheme is susceptible to online and offline password guessing attacks, lack of resistance to user and server impersonation attacks, lack of mutual authentication, and lack of user anonymity. Khan et al. remove drawbacks from An et al.’s scheme by means of proposing an improved user authentication scheme.

Nowadays people are increasingly relying on mobile devices, so the mobile payment will be the trend in the nearly future. Payments with smart cards is less convenient than mobile payment. The current mobile payments such as Alipay, LINE pay and Apple Pay adopt virtual currency or credit cards. Users using mobile payment need to store the credit card numbers in their mobile devices. If a thief knows the credit card number, he can complete fraudulent transactions. Every time you use your credit card you
are making your card number available to everyone who is involved in the transaction, from the sales clerk to the billing staff of the creditor. In this paper, we proposed a new remote user authentication scheme based on biometric technology. The architecture is shown in Fig. 1. The mobile device uses the camera, microphone, fingerprint reader or other devices to capture the user’s biometric features like face, iris, fingerprint, voice, etc. The server stores the user’s information and biometric file in the server’s database. In authentication process, biometric information inputted by the user is encrypted by the mobile device and is sent to a remote authentication server. Biometric authentication system verifies the users’ identification by comparing biometric traits to stored data in databases. We substitute bounding with mobile devices for smart cards. The user only completes a transaction by the bounding device.

This paper is organized as follows: Section 2 gives some background of advanced encryption standard (AES) and one-way hash function. Section 3 describes our new biometric based remote user authentication scheme. Section 4 and 5 are about security analysis and implement result. Finally, a conclusion is offered in Section 6.

2 Related Technology

2.1 Advanced Encryption Standard

The Advanced Encryption Standard is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [13]. AES is based on the Rijndael cipher [14] developed by Joan Daemen and Vincent Rijmen. NIST selected three different.

---

Fig. 1 Architecture
key sizes from Rijndael cipher, each with a block size of 128 bits. The three distinct key sizes are 128, 192 and 256 bits. AES operates on a $4 \times 4$ column-major order matrix of bytes and has the following property.

- AES can be applied to a file of all sizes and types.
- AES is a symmetric-key algorithm.
- AES is a block cipher using an iterative structure known as a substitution-permutation network
- AES is fast in both software and hardware.

The number of repetitions of transformation rounds is specified by the key length used for an AES cipher. AES executes 10, 12 and 14 cycles of repetition for the key size of 128, 192 and 256 bits respectively. Every cycle consists of several processing steps. There are four main steps, which are **SubBytes**, **ShiftRows**, **MixColumns** and **AddRoundKey**. In the **SubBytes** step, each byte in the state matrix is replaced with a byte using a substitution box, the Rijndael S-box. The S-box is derived from the multiplicative inverse over GF($2^8$). The **ShiftRows** operates on the rows of the state. It shifts the bytes in each row by a particular offset cyclically. The first row is left unchanged and each byte of the second row is shifted one position to left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively. The n-th row is shifted left circular by n − 1 bytes. In the **MixColumns** step, the algorithm uses an invertible linear transformation to combine the four bytes of column of the state. During this operation, each column is transformed using a fixed matrix. In the **AddRoundKey** step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael key expansion algorithm. To combine with the state, the corresponding byte of the subkey does a bitwise XOR operation on each byte of the state. The procedure of AES can be divided into four steps, **KeyExpansions**, **InitialRound**, **Rounds** and **FinalRound**. In the **KeyExpansions**, Rijndael key schedule is used to derive round keys from the cipher key. The **InitialRound** execute **AddRoundKey**. The **Rounds** sequential executes **SubBytes**, **ShiftRows**, **MixColumns** and **AddRoundKey**. The **FinalRound** sequential executes **SubBytes**, **ShiftRows** and **AddRoundKey**.

### 2.2 One-Way Hash Function

A one-way collision-resistant hash function $h$ takes an input as arbitrary length binary string $x \in \{0, 1\}^*$ and outputs a binary string $h(x) \in \{0, 1\}^n$ of fixed-length $n$. The hash function may be the fingerprint of a file, a message, or other data block, and has the following attribute [11].

- Hash function can be applied to a data block of all sizes.
- For any given $x$, it is easy to compute the message digest $h(x)$. Its implementation in software and hardware is simple.
- The output length of the message digest $h(x)$ is fixed.
- Given an output hash value $y = h(x)$ and the hash function $y = h(\cdot)$. It is computationally infeasible to derive the input $x$. This property is called the one-way property.
- For any given input $x$, finding any other input $y \neq x$ so that $h(y) = h(x)$ is computationally infeasible. This property is referred to as weak-collision resistant property.
- Finding a pair of input $(x, y)$, with $x \neq y$, so that $h(x) = h(y)$ is computationally infeasible. This property is referred to as strong-collision resistant property.
3 New Biometric-Based Authentication Scheme

In this section, we give the detail of our remote authentication scheme. There are four phases in the proposed scheme, which are registration phase, authentication phase, password change phase and UUID change phase. The notations used throughout this paper are summarized in Table 1.

### 3.1 Registration Phase

In this phase, \( U_i \) sends the registration request to \( S_j \). Then \( U_i \) inputs user’s information and biometric template to accomplish the registration. As shown in Fig. 2, the detail of the phase is presented as follows:

1. \( U_i \) gets a smart card from the service provider through a secure channel (e.g. in person). Each smart card has a unique identification \( SID_j \) and a nonce \( N_s \) which is encrypted and recorded in the service provider’s database. When a smart card is activated, it generates a nonce \( N_s \) and computes \( M_1 = h(SID_j) \odot N_s \) and stores \( M_1 \) in EEPROM.

2. After getting message \( \{ M_1 \} \), \( U_i \) computes \( M_2 = M_1 \odot h(SID_j) \) to get \( N_s \) from the smart card. Then \( U_i \) chooses his identity \( ID_i \), password \( PW_i \), device UUID \( UUID_i \) and personal biometric file \( Bi \). Next \( U_i \) encrypts user’s information by computing \( M_3 = h(ID_i) \odot M_2, M_4 = h(PW_i) \odot M_2, M_5 = h(UUID_i) \odot M_2 \) and \( f \). The \( f \) is the AES encryption of \( Bi \) and the key of AES is \( h(UUID_i) \). Finally \( U_i \) sends the message \( \{ M_3, M_4, M_5, f \} \) to \( S_j \).

3. After receiving message \( \{ M_3, M_4, M_5, f \} \), \( S_j \) computes \( M_6 = M_3 \odot N_s, M_7 = M_4 \odot N_s, M_8 = M_5 \odot N_s \) and \( B_i \). The \( B_i \) is the AES decryption of \( f \) and the key is \( M_8 \). Then, \( S_j \) stores the user’s registration information \( \{ M_6, M_7, M_8, B_i \} \) in \( S_j \). Next, \( S_j \) computes \( M_9 = PSK \odot h(M_6 || M_8) \) and sends message \( \{ M_9 \} \) to \( U_i \). Finally, \( U_i \) stores message \( \{ M_9 \} \) in user’s device.

| Table 1 Notation |
|------------------|
| Notation | Description |
| ---------------- | ------------ |
| \( U_i \) | The i-th user |
| \( S_j \) | The j-the server |
| \( ID_i \) | The identity of \( U_i \) |
| \( PW_i \) | The password of \( U_i \) |
| \( UUID_i \) | UUID of the user \( U_i \) ‘s mobile device |
| \( SID_j \) | The identity of the smart card |
| \( B_i \) | The biometric template of \( U_i \) |
| \( PSK \) | Pre-shared key of servers |
| \( N_s \) | A random nonce chosen by \( U_i \) |
| \( N_s \) | A random nonce chosen by \( S_j \) |
| \( h(\cdot) \) | A secure one-way hash function |
| \( AES.E(\cdot) \) | Encryption of Advanced Encryption Standard |
| \( AES.D(\cdot) \) | Decryption of Advanced Encryption Standard |
| \( \odot \) | The bitwise XOR operation |
| \( || \) | The concatenation operation |
3.2 Authentication Phase

In this phase, \( U_i \) and \( S_j \) generate nonce \((N_u, N_s)\) for mutual authentication. After mutual authentication completes, \( U_i \) encrypts biometric template by AES encryption. As shown in Fig. 3, the detail of the phase is presented as follows:

1. \( M_1 = h(SID_j) \oplus N_s \)

2. Compute
\[
\begin{align*}
M_2 &= M_1 \oplus h(SID_j) \\
M_3 &= h(ID_i) \oplus M_2 \\
M_4 &= h(PW_i) \oplus M_2 \\
M_5 &= h(UUID_i) \oplus M_2 \\
f &= AES.E(B_i) \text{ by key } = h(UUID_i)
\end{align*}
\]

3. Compute
\[
\begin{align*}
M_6 &= M_1 \oplus N_s \\
M_7 &= M_4 \oplus N_s \\
M_9 &= M_3 \oplus N_s \\
B_i &= AES.D(f) \text{ by key } = M_8
\end{align*}
\]

4. Store \( \{M_6, M_7, M_9, B_i\} \) in server

5. Compute
\[
M_9 = PSK \oplus h(M_6 || M_8)
\]

6. Store \( \{M_9\} \) in user’s device

(1) \( U_i \) inputs his ID and get the UUID from user’s device. Then \( U_i \) computes \( M_{10} = h(ID_i) || h(UUID_i) \). PSK = \( M_9 \oplus h(M_{10}) \). \( M_{11} = h(ID_i) \oplus h(SID_j) || PSK \). \( M_{15} = h(M_{10}) \oplus N_u \) and \( M_{13} = h(N_u) \). Final \( U_i \) sends the message \( \{M_{11}, M_{12}, M_{13}\} \) to server \( S_j \).

(2) After receiving message \( \{M_{11}, M_{12}, M_{13}\} \), \( S_j \) computes \( M_{14} = M_{11} \oplus h(SID_j) || PSK \). Then \( S_j \) compares \( M_{14} \) with \( M_6 = h(ID_i) \) stored in server. If it does not match, \( S_j \) rejects the session; otherwise, \( S_j \) can make sure the authentication user \( U_i \) and computes \( M_{15} = M_9 \oplus M_8 \) and \( M_{16} = M_{12} \oplus h(M_{13}) \). \( S_j \) checks if \( h(M_{16}) \) and \( M_{13} \) are equal. If they are not equal, \( S_j \) rejects the session; otherwise, \( S_j \) generates a nonce \( N_s \) and computes \( M_{17} = h(M_{15} || M_{16}) \oplus N_s \) and \( M_{18} = h(N_s) \). Final \( S_j \) sends the message \( \{M_{17}, M_{18}\} \) to \( U_i \).

(3) After receiving message \( \{M_{17}, M_{18}\} \), \( U_i \) computes \( M_{19} = M_{17} \oplus h(M_{16} || N_s) \). Then \( U_i \) checks whether \( h(M_{19}) \) and \( M_{18} \) are equal. If they are not equal, \( U_i \) rejects the session; otherwise, \( U_i \) inputs his biometric file \( B_i \) and computes \( M_{20} = h(N_s || M_{19}) \) and \( f \). The \( f \) is AES encryption of \( B_i \) and the key of AES is \( h(UUID_i) \oplus M_{19} \). Final \( U_i \) sends message \( \{M_{20}, f\} \) to \( S_j \).

(4) After receiving message \( \{M_{20}, f\} \), \( S_j \) computes \( M_{21} = h(M_{18} || R_s) \). Then \( S_j \) check if \( M_{20} \) and \( M_{20} \) are equal. If they are not equal, \( S_j \) rejects the session; otherwise, the user passes the authentication and the mutual authentication completes. Then \( S_j \) gets the \( B_c = AES.D(f) \) and the key of AES is \( M_9 \oplus N_s \). Finally, \( S_j \) recognizes the user by using a recognition system to compare \( B_c \) with \( B_c \). If \( B_c \) and \( B_c \) are matched, \( S_j \) confirms that \( U_i \) is a legal user; otherwise, \( S_j \) stops the session.
### 3.3 Password Change Phase

In this phase, \( U_i \) could change the old password \( PW_{\text{old}}^i \) to the new password \( PW_{\text{new}}^i \). The detail of this phase is illustrated as Fig. 4 and is presented as follows:

1. The \( U_i \) sends the password-change request to \( S_j \). Then \( S_j \) computes \( M_{22} = h(ID_i)||PSK||h(M_{10}) \). Finally, \( S_j \) sends the message \( M_{22} \) to \( U_i \).
2. After receiving message \( M_{22} \), \( U_i \) computes \( M_{23} = M_{22} \oplus h(SID_i||PSK) \), \( M_{24} = h(ID_i)||h(SID_i)||PSK||M_{22} \), \( M_{25} = h(PW_{\text{old}}^i)||h(PW_{\text{new}}^i)||h(ID_i)||h(UUID_i)||M_{23} \) and \( f \). The \( f \) is the AES encryption of \( B_i \) and the key of AES is \( h(PW_{\text{old}}^i)||h(PW_{\text{new}}^i)||h(M_{23}) \).
3. Finally, \( U_i \) sends the message \( \{M_{24}, M_{25}, f\} \) to \( S_j \).
4. After receiving message \( \{M_{24}, M_{25}, f\} \), \( S_j \) computes \( M_{26} = h(M_{10})||M_{13} \) and \( f = \text{AES}.D(f) \) by key = \( h(UUID_i)||M_{26} \).
5. Finally, \( S_j \) recognizes the user by using a recognition system to compare \( B_i \) with \( B_{i'} \). If the recognition is passed, \( S_j \) replaces \( M_{new}^7 = M_7 \oplus N_s \).

![Authentication phase](image)
3.4 Uuid Change Phase

In this phase, $U_i$ could change the old device’s UUID $UUID^{old}_i$ to the new device’s UUID $UUID^{new}_i$. The steps of this phase are similar with password-change phase. The detail of this phase is presented as follows:

1. $M_{22} = h(SID_j || PSK) \oplus N_s$

2. Compute
   
   $M_{23} = M_{22} \oplus h(SID_j || PSK)$
   $M_{24} = h(ID_i) \oplus h(SID_j || PSK) \oplus M_{23}$
   
   $M_{25} = h(PW^{old}_i) \oplus h(PW^{new}_i) \oplus h(h(ID_i)) \oplus h(UUID^{old}_i) \oplus h(UUID^{new}_i) \oplus M_{23}$

   $f = AES.E(B_i) by key \equiv h(PW^{old}_i) \oplus h(PW^{new}_i) \oplus M_{23}$

3. $M_{26} = M_{24} \oplus h(SID_j || PSK) \oplus N_s$

4. Compare $M_{26}$ with $M_8$ stored in server
   
   If it matches, then computes
   
   $M_{27} = M_{26} \oplus h(M_8 || N_s)$
   $B_c = AES.D(f) by key \equiv M_{27} \oplus N_s$

5. Recognize the user by using a recognition system
   
   If $B_c$ matches $B_i$, server replaces $M_7$ with $M^{new}_7 = M_7 \oplus M_{27}$

rejests the session; otherwise, $S_j$ computes $M_{33} = M_{32} \oplus h(M_8 || M_9 || N_s)$ and $B_c = AES.D(f)$. The key of AES is $M_{33} \oplus N_s$. Next, $S_j$ recognizes the user by using a recognition system to compare $B_c$ with $B_i$. If it is recognized successfully, $S_j$ replaces $M^{new}_8 = M_8 \oplus M_{33}$ and replies to $U_i$ with the information of a success. Finally, $U_i$ replaces $M^{new}_9 = PSK \oplus h(h(ID_i) \oplus h(UUID^{new}_i))$. 

Fig. 4 Password change phase
4 Security Analysis

4.1 Informal Security Analysis

4.1.1 Mutual Authentication

In Step 1 of the authentication phase, the $U_i$ regenerates a nonce $N_u$ and computes $M_{12} = h(h(ID_i)||h(UUID_i))||N_u$, $M_{12} = h(h(ID_i))||h(UUID_i) || h(N_u)$. Then in Step 2, $S_j$ could recover $N_u$ from $M_{12}$ and authenticate $U_i$ by checking whether $h(N_u)$ is equal to $M_{13}$ or not. In Step 3, $U_i$ could recover $N_u$ and $N_s$ sent by $S_j$ are correct by checking whether $h(N_u)||N_s$ computed by $S_j$ and $M_{20}$ are equal. Therefore, the proposed scheme could provide mutual authentication between $U_i$ and $S_j$.

4.1.2 Anonymity

In the authentication process, all the information ($ID_i$, $UUID_i$, $SID_j$) are protected by hash function. In message $M_{12} = h(h(ID_i))||h(UUID_i)||N_u$, the $ID_i$ and $UUID_i$ are protected by nonce $N_u$. The adversary must have the $N_u$, but $N_u$ changes over sessions. Even if the adversary has the $N_u$, he is hard to recover $ID_i$ and $UUID_i$ from $h(h(ID_i))||h(UUID_i)$. As the result, our scheme could preserve the user anonymity property.

4.1.3 User Impersonation Attack

To impersonate as a legal user, the adversary must be able to generate the messages {$M_9$, $M_{11}$, $M_{12}$}. The adversary must know the user information $ID_i$ and $UUID_i$. But the $ID_i$ and $UUID_i$ are protected in message $M_{11} = h(h(ID_i)||h(UUID_i))||N_u$, the $ID_i$ and $UUID_i$ are protected by nonce $N_u$. Therefore, our scheme could withstand the user impersonation attack.

4.1.4 Man-in-the-Middle Attack

MITM is an active attack which the adversary eavesdrops the communication and tries to extract the information to complete the authentication. The adversary intercepts the messages {$M_{11}$, $M_{12}$, $M_{13}$}, {$M_{17}$, $M_{18}$} and {$M_{20}$, $f$} and uses the previous messages to pass the authentication. Although the adversary has $M_7 = h(ID_j)||h(SID_i||PSK)$, he does not have the server’s secret $h(SID_i||PSK)$. Therefore, he cannot extract $h(ID_j)$ from $M_7$ and obtains other information. Therefore, the proposed scheme can withstand the MITM attack.

4.1.5 Sever Spoofing Attack

Under this attack, the adversary attempts to masquerade as a server $S_j$. When the user $U_i$ sends the messages {$M_{11}$, $M_{12}$, $M_{13}$} to the server. The adversary intercepts that message, where $M_{11} = h(ID_j)||h(SID_i||PSK)$, $M_{12} = h(ID_j)||h(UUID_i) || h(N_u)$ and $M_{13} = h(N_u)$. The adversary can try to replay the old message {$M_7$, $M_{18}$}, where $M_17 = h(ID_j)||h(UUID_i)$
This attempt will not succeed, since the different session uses different nonces, that is $N_u \neq N'_u$ and the session will reject by $U_i$. Therefore, our scheme could resist the server spoofing attack.

### 4.1.6 Password Guessing Attack

We have made use of a randomly-generated nonce to protect users’ passwords. Even if an attacker intercepts the message $M_4 = h(PW_i) \oplus M_2$ in registration phase. The complexity of combination of a password and a nonce makes the attacker cannot guess the password and nonce at the same time. Thus, it is computationally infeasible for an attacker to guess the user’s credentials. Thus, our scheme is free from the password guessing attack.

The security analysis of the related scheme and the proposed scheme is summarized in Table 2. The proposed scheme is more secure than Das’s and Li-Hwang’s scheme relatively. In addition, the proposed scheme provides mutual authentication between the user and the server that can verify each other’s identity before transmitting data.

In Table 3, we have compared the computational overhead of the proposed scheme with Das’s scheme and Li-Hwang’s scheme. Though our scheme requires more computational
overheads, but providing more security features. Besides, many operations in our scheme can be pre-computed to cut down the amount of time and the performance EVALUATION in section 5 also shows that the execution time is acceptable. We conclude that the proposed scheme is superior to the other schemes.

4.2 Formal Security Analysis

4.2.1 Adversary Model

According to the classic Dolev-Yao model [17], Das et al.’s threat model [18] and Yu et al.’s [19] assumptions, we improve and propose the hypothesis about the adversary’s abilities which is enumerated in Table 4.

Definition 1 (one-way hash function). We define a one-way collision-resistant hash function $h: \{0, 1\}^\ast \rightarrow \{0, 1\}^n$ that takes an arbitrary length binary string $x \in \{0, 1\}^\ast$ as input and outputs a fixed-length binary string $y = h(x) \in \{0, 1\}^n$. The formula of advantage that an adversary in finding collision is defined as follows:

$$Adv_H^A(t) = \Pr[(x, x') \leftarrow A : x \neq x' \land h(x) = h(x')]$$

where $Pr[E]$ denotes the probability of an event $E$, and $(x, x') \leftarrow A$ indicates the pair $(x, x')$ which is selected randomly by the adversary. The $Adv_H^A(t)$ stands for the probability in the advantage over the random choices made by the adversary $A$ with the execution time $t$. The hash function is considered to be collision-resistant if $Adv_H^A(t) \leq \varepsilon$, for $\varepsilon$ is negligible small. Then we define the random oracle as follows:

Reveal: The random oracle will output the input $x$ from the corresponding hash value $y = h(x)$ unconditionally.

The adversary must derive the biometric template $B_i$ to masquerade as the user to pass the authentication. The experimental algorithm is given in Algorithm $EXP_{A, auth}$.

| Capability | Description |
|------------|-------------|
| Cap. 1     | The adversary can eavesdrop, intercept, insert, delete, or block messages transmitting via the public channel |
| Cap. 2     | Under corrupting mobile devices or smart cards, an adversary can extract all sensitive secret credentials stored in it |
| Cap. 3     | For the n-factor protocol, the adversary can get n-1 of the b authentication factors at the same time |
| Cap. 4     | The adversary can obtain user ID (When evaluating the anonymity of the protocol, the user ID should be assumed to be sensitive information) |
Algorithm $\text{EXP}_{A, \text{auth}}$

1. Eavesdrop the login request message $\{M_{11}, M_{12}, M_{13}\}$ during the authentication phase.
2. Call Reveal oracle on input $M_{13}$ to retrieve the information $N_u$.
   Let $\text{Reveal}(M_{13}) \rightarrow N_u^\prime$.
3. Compute $M_{10}^\prime = h(ID) \| h(UID) = M_{12} \delta N_u$.
4. Eavesdrop the authentication response message $\{M_{17}, M_{18}\}$ from the server.
5. Compute $N_s^\prime = M_{17} \delta h(M_{10}^\prime \| N_u)$ and $h(N_s^\prime)$ if $h(N_s^\prime)$ matches with $M_{18}$, accept $N_s^\prime$ and $N_u^\prime$.
6. Use $h(UID)$, $N_s^\prime$ and $N_u^\prime$ to compute AES encryption key $f = AES.E(B_i)$ and $M_{20} = (N_u^\prime \| N_s^\prime)$.
7. Send $\{M_{20}, f\}$ to the server.
   If $M_{20} = M_{21}$ (computed on server side), pass authentication retrieve $B_s = AES.D(f)$
   return 1 (Success)
   else
     return 0 (Failure)
end if

**Proof** In this proof, we need to construct a model that the adversary can derive the encrypted biometric feature to pass the authentication of the server. For this purpose, the adversary executes the experimental algorithm $\text{EXP}_{A, \text{auth}}$. The success probability for $\text{EXP}_{A, \text{auth}}$ is defined as $\text{Succ}_{A, \text{auth}} = \Pr[\text{EXP}_{A, \text{auth}} = 1] - 1$. The advantage formula of an adversary for this experiment becomes $\text{Adv}_{1}^{H}(t, q_R) = \max \{ \text{Succ}_{A, \text{auth}} \}$, where the maximum probability that adversary takes with the execution time $t$ and the number of queries $q_R$. Our scheme is said to be secure against an adversary for masquerading as a user to pass the authentication of the server, if $\text{Adv}_{1}^{H}(t, q_R) \leq \epsilon$, for $\epsilon$ is negligible small.

Consider the experiment $\text{EXP}_{A, \text{auth}}$, if the adversary can invert the one-way hash function, he/she can obtain key materials to derive AES key and encrypt the biometric feature that can pass the server’s authentication. However, by definition 1, $\text{Adv}_{1}^{H}(t) \leq \epsilon$, for $\epsilon$ is negligible small. Therefore, the adversary has a tiny advantage $\text{Adv}_{1}^{H}(t, q_R) \leq \epsilon$. As a result, the proposed scheme is provably secure against an adversary in the model.
5 Performance Evaluation

5.1 Environment

To realize the performance, we conduct an implementation of a face-based remote authentication scheme on smart phones. We use three different smart phones, HTC One M8, HTC One M7 and Samsung Galaxy S4, to evaluate the execution time of hash function and encryption of AES. We use the jpg images of the size is around 1 MB to calculate the average time of AES encryption. We also test the execution time of hash function and AES decryption on our server. The specification of our server is Intel Core i7-4790 and 16 GB RAM and the server codes by PHP and shell script. We implement AES encryption by Magic Crypt library [15] which shared by Magic Len and the face recognition by openBR library [16]. In our implementation we adopt that the hash function is sha-256 and the key length of AES is 256 bits.

5.2 Performance Result

In our proposed scheme the user need to use 4 times hash function and 1 time encryption of AES in registration phase as well as 8 times hash function and 1 time encryption of AES in authentication phase. The server need to use 2 times hash function in registration phase as well as 6 times hash function and 1 time decryption of AES in authentication phase. The user side result, which is the average execution time of a cryptographic hash function of SHA256 and encryption of AES algorithm, is showed in Table 4. The server side result showed in Table 5 is the average execution time of once hash function and decryption of AES. In our implementation we directly encrypt the image and the file size encrypting by AES is a little larger than original file size. If you want to decrease the transmission data size, you can extract the biometric feature before carrying out the AES encryption. The computation formula is shown as follows:

$$\text{Average Execution Time} = \frac{\sum_{i=0}^{n} \text{ImageSize}_{\text{CryptAlgo}}}{n}$$

In Table 6, we present the execution time of encrypting and decrypting images with an average size 1Mbytes on client side and server side. The size is almost equal to a 512×512 bmp image which can contain critical biometric features.

In order to increase the comparison of testing platforms, we refer to Nur et al.’s research to present the test result as Table 7 [20]. The testing scenario is the AES algorithm with 256-bit key length is carried out 20 times. The performance of Redmi Note 5 is slightly inferior to ones of HTC and Galaxy.
6 Conclusion

In this paper, we propose a biometric-based remote authentication scheme between mobile devices and cloud servers using AES encryption. Security analysis shows that the proposed scheme could satisfy security requirement of remote authentication system. In our proposed scheme, we substitute bounding mobile device for smart card. The proposal is more convenient and suitable for mobile payment environment. Therefore, our scheme provides security and convenience for authentication scheme of mobile payment.

Acknowledgement The authors gratefully acknowledge the support by the National Taiwan University of Science and Technology-Tokushima University Joint Research Program, under Grant TU-NTUST-108-02.

References

1. Lamport, L. (1981). Password authentication with insecure communication. Communications of the ACM, 24(11), 770–772.
2. Haller, Neil. “The S/KEY one-time password system,” IETF RFC 1760, 1995.
3. Gwoboa, H. (1995). Password authentication without using a password table. Information Processing Letters, 55(5), 247–250.
4. Hwang, M.-S., & Li, L.-H. (2000). A new remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics, 46(1), 28–30.
5. Li, Chun-Ta, Lee, Cheng-Chi, Liu, Chen-Ju, amd Lee, Chin-Wen. “A robust remote user authentication scheme against smart card security breach,” 3th Data and Applications Security (DBSec), Jul 2011, Richmond, VA, United States. pp.231–238.
6. Kumari, Saru, and Muhammad Khurram Khan. “Cryptanalysis and improvement of ‘a robust smart-card-based remote user password authentication scheme’,” International Journal of Communication Systems 27.12, pp. 3939–3955, 2014.
7. Li, C.-T., & Hwang, M.-S. (2010). An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and computer applications, 33(1), 1–5.
8. Das, Amal K. “Analysis and improvement on an efficient biometric based remote user authentication scheme using smart cards,” Information Security, IET 5.3, pp. 145–151, 2011.
9. An, Younghwa. “Security Analysis and Enhancements of an Effective Biometric-Based Remote User Authentication Scheme Using Smart Cards,” BioMed Research International, vol. 2012, Article ID 519723, 6 pages, 2012.
10. Khan, Muhammad Khurram, and Saru Kumari. “An improved biometrics-based remote user authentication scheme with user anonymity,” BioMed Research International, vol. 2013, Article ID 491289, 9 pages, 2013.
11. Mishra, Dheerendra, Ashok Kumar Das, and Sourav Mukhopadhyay. “A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards,” Expert Systems with Applications 41.18, pp. 8129–8143, 2014.
12. He, Debiao, and Ding Wang. “Robust biometrics-based authentication scheme for multiserver environment,” Systems Journal, IEEE 9.3, pp. 816823, 2015.
13. Standard, NIST-FIPS. “Announcing the advanced encryption standard (AES),” Federal Information Processing Standards Publication 197 (2001), pp. 1–51, 2001.
14. Daemen, Joan, and Vincent Rijmen. “AES proposal: Rijndael,” 1999.
15. Magic Len, MagicCrypt Library [Online]. Available: https://magiclen.org/aes/.
16. J. C. Klontz, B. F. Klare, S. Klum, A. K. Jain and M. J. Burge, “Open source biometric recognition,” 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013
17. Dolev, D., & Yao, A. (1983). On the security of public key protocols. IEEE Transactions on Information Theory, 29, 198–208.
18. Das, A. K., & Goswami, A. (2015). A robust anonymous biometric-based remote user authentication scheme using smart cards. J King Saud Univ Comput Inf Sci, 27(2), 193–210.
19. Yicheng, Y., Liang, H., & Chu, J. (2020). A secure authentication and key agreement scheme for IOT-based cloud computing environment. Symmetry, 12(1), 150. https://doi.org/10.3390/sym12010150
20. Rachmat, Nur & Samsuryadi, Samsuryadi. (2019). Performance Analysis of 256-bit AES Encryption Algorithm on Android Smartphone. Journal of Physics: Conference Series.

**Sheng-Kai Chen** received is B.E. degree and M.S. degree from the Department of Electronic and Computer Engineering at National Taiwan University of Science and Technology, Taipei, Taiwan, in 2014 and in 2016 respectively. His research focuses on multimedia security development and mobile application authentication.

**Jenq-Shiou Leu** received the BS degree in mathematics and the MS degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1991 and 1993, respectively, and the PhD degree on a part-time basis in computer science from National Tsing Hua University, HsinChu, Taiwan, in 2006. He was with Rising Star Technology, Taiwan, as an R&D Engineer from 1995 to 1997, and worked in the telecommunication industry (Mobitai Communications and Taiwan Mobile) from 1997 to 2007 as an Assistant Manager. In February 2007, he joined the Department of Electronic and Computer Engineering at National Taiwan University of Science and Technology as an Assistant Professor. From February 2011 to January 2014, he was an Associate Professor. Since February 2014, he is a Professor. His research interests include mobile service and platform design and application development of computational intelligence He is a senior member of IEEE.
Wen-Bin Hsieh received his BS degree in Computer Science and Information Engineering from Tamkang University, Taipei, Taiwan and his Ph.D. degree in Electronic Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan. He worked in the information department of Landbank from 2006 to 2009, as an engineer. Now he serves in a government research institute as a senior engineer and a project manager. His research interests include cryptography, communication protocols, mobile communication, clouding computing and machine learning.

Jui-Tang Wang received his Master degree from the Department of Computer Science and Information Engineering at National Cheng-Kung University in 2000 and Ph.D. degree from the Department of Computer Science and Information Engineering at National Chiao-Tung University, Taiwan in 2008. In February 2019, he joined the Department of Electronic and Computer Engineering at National Taiwan University of Science and Technology as an Assistant Professor. His research focuses on wireless communication and security protocol.

Tian Song received his B.E. degree from Dalian University of Technology, China, in 1995, and his M.E. and Dr.E. degrees from Osaka University in 2001 and 2004, respectively. He joined Tokushima University in 2004 as an Assistant Professor. Presently, he is an Associate Professor of the Department of Electrical and Electronic Engineering, Graduate School of Advanced Technology and Science, Tokushima University. He is a member of IEICE and IEEE. His current research interests include video coding algorithms, VLSI architectures, and system design methodology. His recent research also include smart coding algorithms and underwater video technologies.