# Bio-AKA: a New PUF and Fingerprint based Two Factors User Authentication and Key Agreement Scheme

Weixin Bian, Prosanta Gope, Yongqiang Cheng, Qingde Li

*Abstract*— The fingerprint has long been used as one of the most important biological features in the field of biometrics. It is person-specific and remain identical though out one's lifetime. Physically uncloneable functions (PUFs) have been used in authentication protocols due to the unique physical feature of it. In this paper, we take full advantage of the inherent security features of user's fingerprint biometrics and PUFs to design a new user authentication and key agreement scheme, namely Bio-AKA, which meets the desired security characteristics. To protect the privacy and strengthen the security of biometric data and to improve the robustness of the proposed scheme, the fuzzy extractor is employed. The scheme proposed in the paper can protect user's anonymity without the use of password and allow mutual authentication with key agreement. The experimental results show superior robustness and the simplicity of our proposed scheme has been validated via our performance and security analysis. The scheme can be an ideal candidate for real life applications that requires remote user authentication.

***Index Terms*—Mutual authentication, Key Agreement, Physical Unclonable Functions, Fuzzy extractor, Biometric security and privacy.**

## I. Introduction

Traditionally, remote logins are authenticated through password authentication scheme where identity and password are used by a remote server. The Lamport scheme [1] is a typical example of this type of authentication method. The scheme used an insecure channel to authenticate remote users. Lamport's system stores all passwords in a table for authenticating the legitimacy of users. It could be attacked by modifying the password table. Based on ElGamal's cryptosystem [3], Hwang et al [2] improved the scheme by introducing a smart card without the need for password table being stored. The only requirement is a single secret key. In spite of its simplicity, this kind of system can be easily attacked by using a forged identity [4]. One way of reducing its vulnerability is to introduce a certain fingerprint recognition technique into the system, as is proposed in Lee et al. [5]. Similar to smart card-based method [2], this Fingerprint-based scheme is also based on ElGamal's cryptosystem and smart card except that the two secret keys have been kept. Although the security of the system has been strengthened by verifying the legitimate users using their fingerprints, there are some obvious weaknesses in Lee's scheme. It does not allow users to modify their password and is still subject to masquerade attack [6]. Some improvements to Lee's scheme have made by Lin et al. [6]. One common issue for both the above two schemes [5, 6] is that they are not based on mutual authentications protocols and they only performed user authentication (unilateral authentication protocols). Hence, these schemes cannot withstand server spoofing attacks. Lin's scheme has been further improved in [7] by adding a security patch to provide user and remote server mutual authentications. However, this kind of system is still far from secure. In fact, it can be easily attacked by using guessed passwords, and is vulnerable to both user and server masquerade attacks.

To develop more secure authentication systems, a kind of one-way collision free chaotic hash function was introduced in the scheme proposed by Khan et al. [8] in 2008. In this system, the remote user authentication phase can be performed using biometric fingerprint. Khan's scheme has lower time complexity and less power complexity. So, it can be implemented efficiently on mobile devices. However, due to the fact that the information in mobile device can be extracted by an adversary, this scheme is also vulnerable to the offline attack and the masquerade attack. To address this issue, Chen et al. [9] proposed an authentication scheme by using both fingerprint biometric and password. Further to Chen's work, Truong et al. [10] improved the scheme by bringing in a more robust scheme which are not only more resistant to server and user spoofing attack, replay attack, but can also protect the anonymity of users. However, Truong's solution is far from ideal, relevant security issues are still not fully solved. In fact, due to the failing of the three-way challenge-response handshake and the three-factor authentication, a malicious user can guess the secret key of the server. Hence Khan et al. present a solution in [11] which can take the essence of the schemes proposed in [9, 10] while their vulnerability to malicious attacks is remedied.

W.X. Bian is with the School of Computer and Information, Anhui Normal University, Wuhu, 241002, China, Anhui Province Key Laboratory of Network and Information Security, Wuhu, 241002, China and the Department of Computer Science and Technology, University of Hull, Hull, HU6 7RX, UK (e-mail: bwx2353@ahnu.edu.cn).

P. Gope is with the Department of Computer Science, University of Sheffield, UK (e-mail: p.gope@sheffield.ac.uk)

P. Gope, Y. Cheng and Q. Li are with the Department of Computer Science and Technology, University of Hull, Hull, HU6 7RX, UK (e-mail: p.gope@hull.ac.uk; y.cheng@hull.ac.uk; q.li@hull.ac.uk).

In 2010, Li et al. [12] proposed a smart-card-based solution to solve some essential issues in a remote user authentication system. Later, Das [13] identified two main design flaws in Li et al.'s scheme. One is that the validity of the password is not verified in the login phase, and the other is that the verification of the old password is not considered at the stage of password resetting. To correct these flaws, Das [13] proposed an improvement version to Li et al.'s scheme. However, only limited degree of vulnerability to external attacks can be reduced with Das's scheme. Thereby An [14] explored the design faults in Das's scheme, and put forward an improved method, which was later investigated by Both Khan et al. [15] and Ibjaoun et al. [16]. They identified some issues in An's scheme, and proposed an enhanced version to avoid the potential pitfalls involved in An's scheme. In 2014, Li et al proposed an improved method [17] to Das's scheme which can support the session key agreement at the stage of mutual authentication to make up the lack of this function in Das's scheme. Unfortunately, this improved scheme does not support three-factor authentication and cannot ensure user's privacy. To overcome the weaknesses, Chaturvedi et al. [18] designed an authentication and key agreement protocol, which can inherit all its original merits.

### *A. Related Work*

Unlike identity or password used in traditional remote authentication schemes, biometrics are uniquely associated with an individual as a native alternative to enable a reliable biometric authentication. It offers important advantages over passwords and cryptographic keys [12], such as resistant to copy or being guessed, hard to be forged, and no necessity from users to remember the keys or lose them. For the same reasons, however, it is critical to keep the privacy and security of the biometrics during the authentication process [19-24]. Once the biometrics information is compromised, it is not feasible to generate a second one. Hence, for security reasons, the biometric information should not be stored in either user or server side devices directly which will impose significant risks [5-8]. To mitigate the risks, some researchers [9-15] employed one-way hash function (OWFH) or keyed hash function (KHF) to encrypt biometric before storing it on devices. However, the OWHF is sensitive to subtle changes in the inputs, especially when human's fingerprints continue to change slightly over time. Thus, it is unwise to directly use the biometrics as the inputs of the cryptographic hash function. To address this challenge, the fuzzy extractor (FE) technology [17, 18] are employed to perform biometrics verification. The FE first extracts a nearly random string $K$ from the biometrics information which can tolerate to errors [19, 20]. Then under the help of auxiliary information, FE can extract the same $K$ even if there are changes in the inputs within an acceptable range. However, the auxiliary information is stored in the device directly in their scheme, and thus cannot ensure the security of users' biometric privacy. Moreover, as the commonly used pairwise templates match with the encrypted biometric directly, it has a potential risk of leaking the privacy of user's biometrics information. Zhou et al. [23] proposed a Threshold Predicate Encryption scheme to only reveal the matched result so no biometric data can be learned. The use of adversarial machine learning has been reported in [24], where Wang et al. investigated privacy-malicious attacks on the preserving vulnerability in a biometric database by using critical biometric similarity information in machine learning. As the encrypted biometric data is stored in database in [23, 24], the risks of biometrics information leakage do exist with these approaches.

Recently, a few Internet of Things (IoT) systems have proposed to use physical unclonable functions (PUFs) for mutual authentication schemes [25-27]. PUFs are based on the random differences in Integrated Circuits (ICs) introduced during manufacturing processes. Just like the biometrics of human beings (e.g. fingerprints), each PUF carries its unique physical characteristics from being produced. Hence, to predict and produce a clone of PUF is almost impossible. Researches used PUF enabled devices in a remote authentication scheme to achieve mutual authentication has been very successful benefiting from the fact that this "biometric" cannot be duplicated or cloned. However, it is not able to verify the identity of the user who uses it. An attempt has been made by Gope et al. [26] to input user's biometric thumb impression into the PUF and then generate the biometric key to verify the user identity. In the proposed scheme, the fingerprint biometric is input directly into PUF during the registration phase and the authentication phase. However, it should be noted that this scheme does not support noisy PUF environment. As mentioned above, the fingerprint biometrics changes slightly over time, the result from this scheme is unreliable because the subtle perturbations in inputs will cause unpredictable outputs of a PUF. Furthermore, the user needs to provide password to complete registration and authentication in proposed scheme. This will increase the complexity of system and also produce security risks to system.

### *B. Our Contribution*

To address these flaws in existing biometrics-based remote authentication systems, in this paper we propose a secure and light-weight two factors user authentication and key agreement scheme combining PUFs and fingerprint biometric. To provide two-factor authentication to remote authentication systems, in addition to user's biometrics as the first authentication factor, we proposes the use of PUFs as the second authentication factor. Motivated by the recent success of PUF-based light-weight mutual authentication in IoT systems, PUFs are introduced into our scheme to enhance the security of design. Furthermore, the user can access the remote system using only his fingerprint and does not rely on any passwords. So, the user does not have to remember and update his/her password. Our proposed scheme to combine PUFs and fingerprint biometric can solve the problems mentioned above and also meet the desired requirements of security and efficiency. The key contributions of this paper can be summarized as below:

(1) Design of a two factors user authentication and key agreement scheme combining PUFs and fingerprint biometric, which has high security, convenience and efficiency.

(2) Both user and device have their own unique physical feature, which can provide the key security properties.

(3) No need to store user's biometric information in device, which can completely avoid the risk of leaking biometric information.

(4) Extraction or reconstruction of the key from the biometric or the PUFs response with noise by using the FE.

(5) A password free scheme for registration, login, authentication and secure session key establishment.

The rest of the paper is organized as follows: a brief review of fuzzy extractor and PUFs will be given in Section II. Section III describes the detailed scheme. The proposed two factors user authentication and key agreement are presented in Section IV. Section V analyses the security and performance informally and formally. Section VI concludes the paper.

The notations used in this paper are defined in Table I.

TABLE I
NOTATIONS USED IN THIS PAPER

| Symbol | Description |
|---|---|
| $U_i$ | The user |
| S | Server |
| $ID_i$ | Identity of $U_i$ |
| $h(\cdot)$ | One-way hash function |
| FE | Fuzzy extractor |
| $\|$ | Concatenation operation |
| $PUF_i$ | Physically uncloneable functions of $U_i$ |
| $C_i$ | Challenge of $U_i$ |
| $R_i$ | Response of the respective PUF for $C_i$ |
| $F_i$ | The fingerprint template of $U_i$ |
| $\oplus$ | Exclusive-OR operation |
| $SK$ | Session key between $U_i$ service |

## II. PRELIMINARIES

We briefly introduce the preliminary background of PUF, FE and the System Model in this section.

### A. Fuzzy Extractor

A *fuzzy extractor* [19, 20] is defined as a pair of functions $FE.Gen(\cdot)$ and $FE.Rec(\cdot)$, corresponding to the key generation and reproduction procedures respectively. $FE.Gen(\cdot)$ is a basically a probabilistic function, which generates a key $K$ and an auxiliary data $A$ from an input $D$, i.e., $(K, A) = FE.Gen(D)$. In contrast, $FE.Rec(\cdot)$ is a deterministic function, which reconstructs the key $K$ from a given data $A$ and any noisy input $D'$, where $D'$ is an approximation of $D$ subject to satisfaction of Hamming distance between $D$ and $D'$, i.e. $dis(D,D') \le t$. $t$ is a threshold. Formally, $(K, A) = FE.Gen(D) \rightarrow K=FE.Rec(D', A)$. It's obvious that the success of FE relies on the similarity between original data and the noisy input data.

### B. PUF

A PUF is a kind of hardware function implementation circuit with intrinsic chip characteristics of uniqueness and randomness. A PUF circuit can be understood as the fingerprint of a chip, which is achieved according to the process parameter deviation introduced in the chip manufacturing process. So, a PUF is a physical system interacting in a very sophisticated way with challenges and produces unique but unpredictable responses. This physical system is built by an uncontrollable random process, and thus is hard to clone. Furthermore, mathematical modeling of the PUF is almost impossible to be built because it is based on many sophisticated interactions. So, it's not possible to use cryptographic primitives to reproduce a PUF. Essentially, a PUF uses an exceedingly complex physical system [27] to generate a set of responses from a set of challenges. Mathematically, the PUF can be expressed in the following form,

$$R = PUF(C) \tag{1}$$

Where $R$ is the corresponding set of responses given $C$ as the set of all possible challenges. The function is derived from the intrinsic randomness of the manufacturing process of IC, and cannot be controlled. So, PUF is entirely relied on the random process in the IC manufacturing, and it is actually impossible to make two entirely identical PUFs. The same response will be produced by a PUF if the challenge is the same; but the responses will be different if two different PUFs are given the same challenge. However, the output of the PUF may contain noise due to the changes of working environment (e.g. temperature, air humidity). To support noisy PUF environment, the concept of FE is introduced. Let us assume $P = \{PUF_1(\cdot), \cdots, PUF_M(\cdot)\}$ as a set of PUFs and $C = \{C_1, \cdots, C_N\} s.t. \forall n C_n \in \{0,1\}^k$ as the set of challenges, then a $(d, h, l, \lambda, \varepsilon)$ -secure PUF needs to meet the following requirements [26, 27]:

$$\mathrm{P_r}\left[ d_H(PUF_m(C_p), PUF_n(C_p)) > d \right] \ge 1-\varepsilon, \quad 1 \le m, n \le M \wedge m \ne n; 1 \le p \le N \tag{2}$$

$$\mathrm{P_r}\left[ d_H(PUF_m(C_p), PUF_m(C_q)) > d \right] \ge 1-\varepsilon, \quad 1 \le m \le M, 1 \le p, q \le N \wedge p \ne q \tag{3}$$

$$\mathrm{P_r}\left[ \hat{H}_\infty(PUF_m(C_p), PUF_n(C_q)) > \lambda \right] \ge 1-\varepsilon, \quad 1 \le m, n \le M \wedge m \ne n; 1 \le p, q \le N \wedge p \ne q \tag{4}$$

Where $d_H$ is the Hamming distance and $\hat{H}_\infty$ is the min-entropy of the PUF output.

### C. System Model

The Fig. 1 describes our system model for mobile user remote authentication. The mobile devices used in the system can be in the form of any smart devices such as tablets, mobile phones and laptops, which are the most frequently used in the remote authentication system. In terms of security, the authentication system can achieve two-way secure communication between user's mobile devices and serve. In this model the mobile device is equipped with a PUF and fingerprint sensor, and the server is considered as the trusted party.

## III. THE PROPOSED SCHEME

Here we describe the details of our novel PUF and fingerprint biometrics based remote user authentication and key agreement (Bio-AKA) scheme. The proposed scheme supports remote user authentication without password and session key agreement between the user and the server after the authentication. We use fuzzy extractor to generate a user secret key $K_u$ from fingerprint biometric template captured in registration phase, and recover $K_u$ from fingerprint biometric template captured in login phase. In our scheme the biometric information of users will not be stored on any devices, which can completely eliminate the risk of biometric information

being leaked. The proposed scheme includes three phases: registration, login and the mutual authentication and key agreement phases.

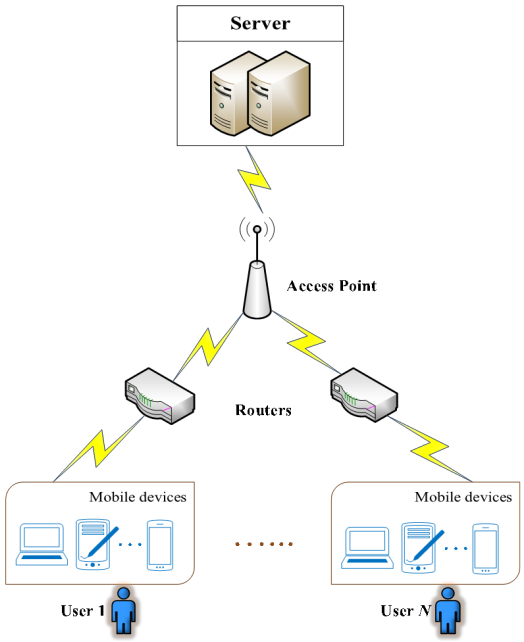


Figure.1. System model.

### A. Registration Phase

User is required to complete registration procedure before being authenticated. The communication between the server $S$ and the user $U_i$ in this stage is required to be completed over a secure channel. Fig. 2 illustrates detailed steps for how to register.

**Step R1:** User $U_i$ picks an identity $ID_i$, and inputs his/her fingerprint on a mobile device. Then $U_i$ extracts fingerprint biometric template $F_i$ from input fingerprint and randomly generates a challenge $C_i$ and a random number $N$.

**Step R2:** $U_i$ computes the PUF outputs $R_i = PUF_i(C_i)$ and obtains the user secret key $K_u$ and the auxiliary data $FA$ from fingerprint biometric template $F_i$ using the procedure $FE.Gen(\cdot)$ i.e., $(K_u, FA) = FE.Gen(F_i)$. And then the $U_i$ computes $W = h(ID_i \| K_u)$, $C_i^* = C_i \oplus h(K_u)$ and $AID_i = ID_i \oplus h(K_u \| N)$. Finally, $U_i$ sends $\{AID_i, (C_i^*, R), W\}$ along with a request for registration $Reg_{req}$ via a secure channel to the server. It is noteworthy that $K_u$ is not disclosed to any others in subsequent communications because it depends only on the biometric of user.

**Step R3:** The server first checks the uniqueness of the $AID_i$ Upon receipt of the $Reg_{req}$ sent by the user $U_i$. Then the server generates randomly a private key $K_s$ and a user unique random number $e$. Next, the server computes $E_i = h(K_s \| e) \oplus W$. Finally, the server completes the registration by sending $E_i$ to $U_i$ by a secure channel.

**Step R4:** Upon receipt of the secret message $E_i$ by the server, $U_i$ computes the secret information $V = h(ID_i \| K_u \| N)$, $N^* = N \oplus h(ID_i)$ and $FA^* = h(ID_i \| N) \oplus FA$ for securing network communications in future. Finally, the user stores $\{h(\cdot), E_i, V, N^*, FA^*\}$ into his mobile device.

As noted from above, we don't store the biometric information $F_i$ directly stored on user's mobile device, and we only stores the encrypted auxiliary data extracted from the fingerprint template using the Fuzzy Extractor. In addition, our scheme does not need to use password. So it can provide a more secure and convenient user authentication scheme.

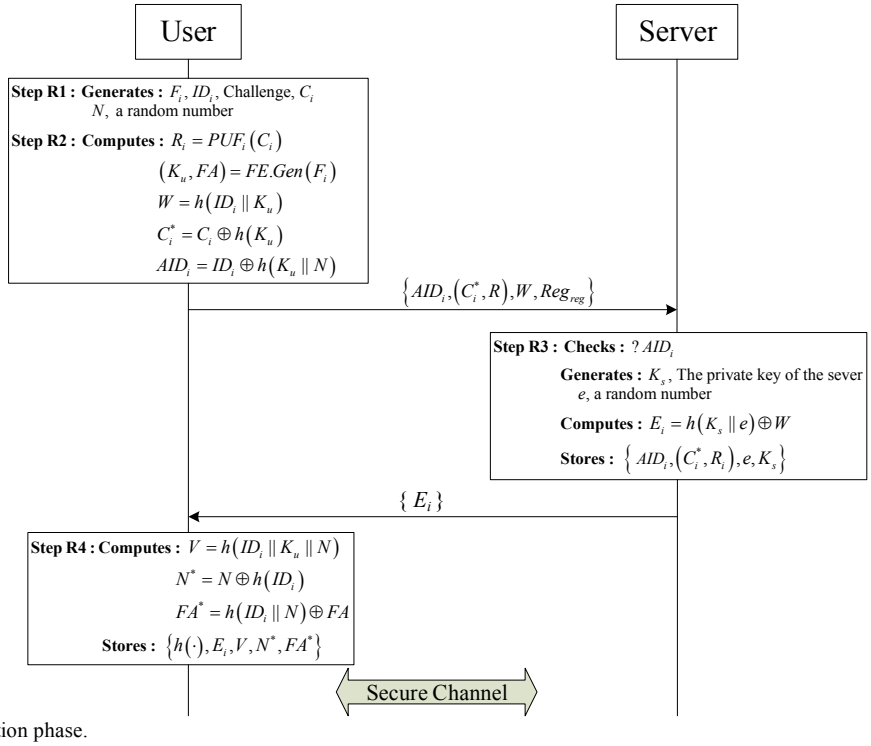


Figure.2. Bio-AKA: Registration phase.

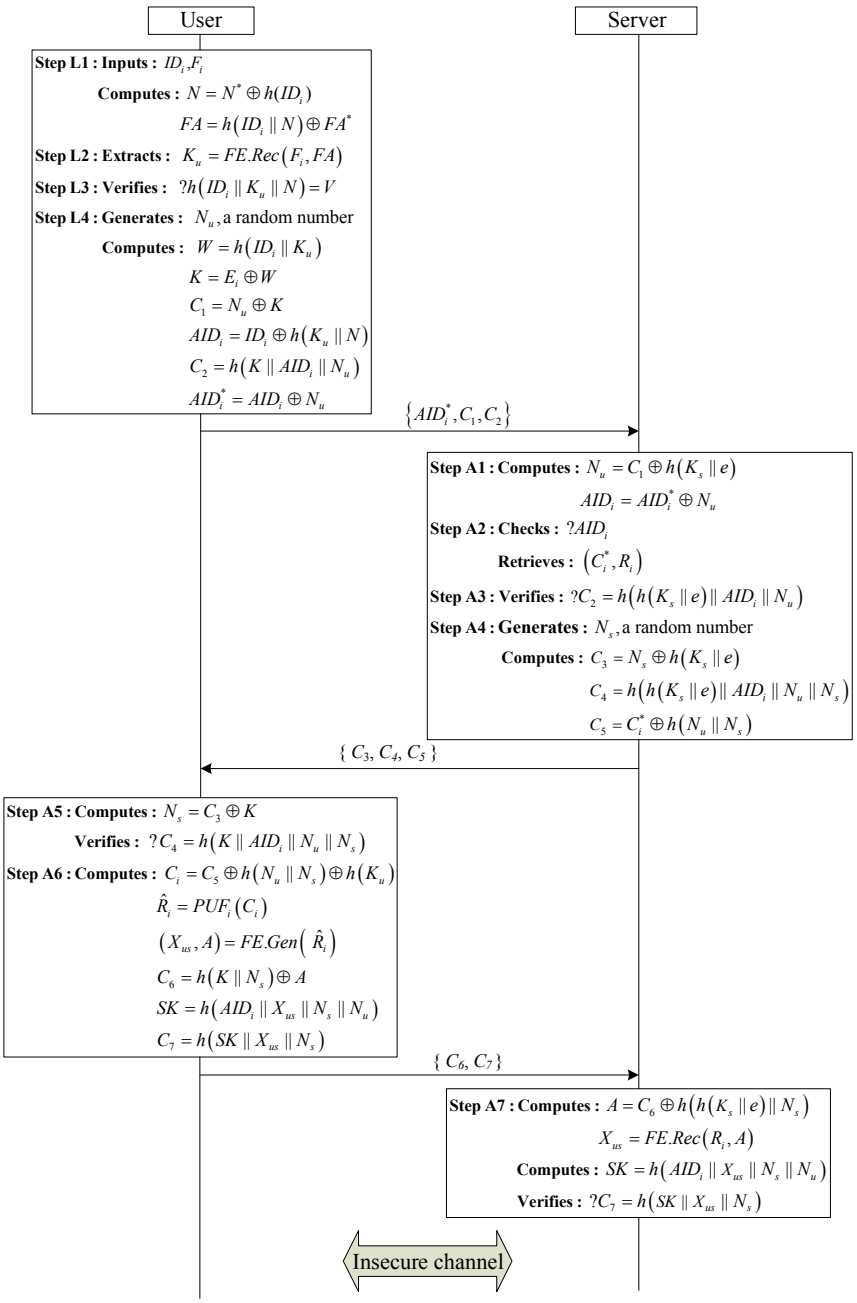


Figure.3. Bio-AKA: Login, authentication and key agreement phases.

### *B. Login Phase*

Fig. 3 shows the login phase procedures. First a user attempting to login a server *S* remotely, he/she has to enter his/her identity $ID_i$, and imprints fingerprint via his/her mobile device. Then the following steps will be executed by the mobile device with a fingerprint template $F_i$ extracted by the device.

**Step L1:** Using the $ID_i$ entered by user, $U_i$ first computes $N = N^* \oplus h(ID_i)$, and then decodes $FA = h(ID_i \| N) \oplus FA^*$, obtains the auxiliary data *FA*.

**Step L2:** Extracting the secret key $X_u$ using the procedure $FE.Rec(\cdot)$ i.e., $K_u = FE.Rec(F_i, FA)$.

**Step L3:** $U_i$ computes a key-hash response $h(ID_i \| K_u \| N)$, and then verifies if the response match *V*. If they are not match, the user login is terminated. Only when the user enters identity correctly and offers correct his personal biometric template which is close to that used in registration phase will the verification be successfully passed.

**Step L4:** $U_i$ generates a nonce $N_u$, and then comutes $W = h(ID_i \| K_u)$. After computing *W*, it computes $K = E_i \oplus W$, which is equal to $h(K_s \| e)$. Next, it immediately computes $C_1 = N_u \oplus K$, $AID_i = ID_i \oplus h(K_u \| N)$ and $C_2 = h(K \| AID_i \| N_u)$, $AID_i^* = AID_i \oplus N_u$. The secret value maintained by *S* is hashed and used to mask $N_u$.

$U_i$ completes the login request by sending the message $\{AID_i^*, C_1, C_2\}$ to *S*.

### *C. Mutual Authentication and Key Agreement Phase*

Following the message $\{AID_i^*, C_1, C_2\}$ to request login, Step A1 to Step A7 will be executed by the remote server *S* to realize the mutual authentication and session key agreement as shown in Fig.3.

**Step A1:** *S* first decodes $N_u$ by computing $N_u = C_2 \oplus h(K_s \| e)$ followed by computing $AID_i$ from the secrete message $AID_i^*$ i.e., $AID_i = AID_i^* \oplus N_u$.

**Step A2:** The server checks the validity of $AID_i$. If $AID_i$ is valid, then it locates user's identity $AID_i$ in its database and instantly retrieve and loads the challenge-response pair $(C_i^*, R_i)$. Otherwise, *S* terminates this session.

**Step A3:** *S* first computes a key-hash response $h(h(K_s \| e) \| \| AID_i \| N_u)$, and then verifies whether it is equal to $C_2$. If this verification fails, S terminates the session.

**Step A4:** *S* generates a nonce $N_s$, and then computes $C_3 = N_s \oplus h(K_s \| e)$ using the secret values maintained by it and subsequently computes $C_4 = h(h(K_s \| e) \| AID_i \| N_u \| N_s)$, $C_5 = C_i^* \oplus h(N_u \| N_s)$ and then sends a composite response message $\{C_3, C_4, C_5\}$ to $U_i$.

**Step A5:** Upon receipt of the message in step 4, $U_i$ first decodes $N_s$ by computing $N_s = C_3 \oplus K$, and then verifies whether the key-hash response $h(K \| AID_i \| N_u \| N_s)$ is equal to $C_4$. If not, $U_i$ terminate the session.

**Step A6:** $U_i$ obtains original challenge $C_i$ by computing $C_i = C_5 \oplus h(N_u \| N_s) \oplus h(K_u)$ using the secret values maintained by it and *S*. Next, $U_i$ computes the PUF response $\hat{R}_i$ to challenge $C_i$ i.e., $\hat{R}_i = PUF_i(C_i)$, and subsequently obtains the key-element $X_{us}$ and auxiliary data $A$ using the procedure $FE.Gen(\cdot)$ i.e., $(X_{us}, A) = FE.Gen(\hat{R}_i)$. Next, the mobile device computes $C_6 = h(K \| N_s) \oplus A$, the session key $SK = h(AID_i \| X_{us} \| N_s \| N_u)$ and $C_7 = h(SK \| X_{us} \| N_s)$. Finally, $U_i$ sends the composite message $\{C_6, C_7\}$ to *S*.

**Step A7:** Upon receiving $U_i$'s message $\{C_6, C_7\}$, *S* first computes and decodes the auxiliary data $A = C_6 \oplus h(h(K_s \| e) \| N_s)$ using the secret values maintained by it, then using the reconstruction function $FE.Rec(\cdot)$ to obtains the key-element $X_{us} = FE.Rec(R_i, A)$. And then *S* achieves the session key *SK* by computing $SK = h(AID_i \| X_{us} \| N_s \| N_u)$. Finally, *S* verifies whether the key-hash response $h(SK \| X_{us} \| N_s)$ is equal to $C_7$. If they are different, then *S* terminates this session.

## IV. SECURITY ANALYSIS AND DISCUSSION

Both informal and formal analyses have been performed to ensure the security of our scheme. The security of our scheme is proved theoretically by using the BAN logic [28].

### *A. Informal Security Analysis*

In the analysis, we have respected the facts that the smart mobile devices are prone to be tampered [29-33], as a result, the information stored in the devices are insecure. Hence it is reasonable to assume the information on mobile device can be stolen by a hacker $U_a$. Furthermore, it's possible the communication channel between the server and the device can be intercepted and controlled by the hacker as the transmission is made over public channels.

We claim our scheme is secure and can satisfy the following propositions.

**Proposition 1** User anonymity and untraceability protection.

**Proof.** In our scheme, the real identity $ID_i$ of the user is enciphered with user's secret key $K_u$ extracted from biometric and a random secret number *N* in registration phase. To obtain $ID_i$ from $AID_i = ID_i \oplus h(K_u \| N)$, $K_u$ and *N* are essential which are known only to the user. As the biometric information is unique secret to the user only. So, an attacker cannot obtain the real $ID_i$ value, even the server cannot retrieve it. In login phase, $U_i$ sends $\{AID_i^*, C_1, C_2\}$ to *S*, and the parameters in this message are dynamic because user's mobile device generates $N_u$ randomly for each session, which reduces the possibility of traceability. Thus the proposed scheme can provide untraceability and user anonymity hence further protect the user's privacy.

**Proposition 2** Withstanding both online and offline password guessing attacks.

**Proof.** The proposed scheme do not need password support. It only use the secret key $K_u$ extracted from biometric of user to encrypt the user-end data. Therefore, an attacker never have chances to perform password guessing attacks. As a result, there is no risk for password guessing attacks.

**Proposition 3** Withstanding stolen mobile device attacks.

**Proof.** Granting that an attacker $U_a$ has stolen the mobile device of $U_i$, he/she can extract the information $\{h(\cdot), E_i, V, N^*, FA^*\}$ stored in it. As in our scheme, neither the user identity or biometric information will be stored on the device, hence no valuable data can be obtained from the device. In addition, the user identify is not contained in any message in plain text forms, this prevents sensitive information leakage. All the obtained values such as $E_i$, $V$, $N^*$, $FA^*$ are safeguarded using OWFH, where $E_i = h(K_s \| e) \oplus W$, $V = h(ID_i \| K_u \| N)$, $N^* = N \oplus h(ID_i)$, $FA^* = h(ID_i \| N) \oplus FA$. In order to get $ID_i$ from $N^* = N \oplus h(ID_i)$, $U_a$ requires *N* values, the values of *N*, in turn, depends on the value of $ID_i$. As a result, this becomes a paradoxical chicken-egg problem. If $U_a$ would like to obtain $ID_i$ from other data extracted from mobile device, he not only requires *N* but also requires *FA* or $K_u$. It is impossible to get *FA* or $K_u$ without obtaining user biometric. So, the proposed scheme is secure against mobile device stolen attack.

**Proposition 4** Withstanding insider attacks.

**Proof.** In our scheme, $U_i$ freely selects his identity and nonce *N* and does not send $ID_i$ in plain text during user registration phase, and never leaks his biometric $F_i$. $U_i$ submits only $AID_i$, *W* to server, where $AID_i = ID_i \oplus h(K_u \| N)$, $W = h(ID_i \| K_u)$. It is impossible in proposed scheme to obtain secret values $\{K_u, N\}$ because this is protected by the one-way hash. Therefore, no insider can extract the user's secret information. In addition, the secret value $K_u$ has to be extracted through biometric information which is user's unique secret that the insider cannot obtain it. So, even anyone who has access to the server is unable to get user's $ID_i$ without the secret information $K_u$ and nonce *N*. This proves our scheme is free of insider attacks.

**Proposition 5** Withstanding replay attacks.

**Proof.** We assume that $U_a$ somehow intercepts message $\{AID_i^*, C_1, C_2\}$ from an insecure channel sent by user $U_i$ to server *S*. Then, $U_a$ wants to gain access to the server *S* by replaying this message. However, $U_a$ will fail to achieve his aim by this

approach because the nonce $N_u$ changes in each session. The message $\{AID_i^*, C_1, C_2\}$ received by $S$, where $AID_i^* = AID_i \oplus N_u$, $C_1 = N_u \oplus K$ and $C_2 = h(K \| AID_i \| N_u)$, obviously includes a nonce $N_u$ from $U_i$. In order to complete authentication, $S$ must send back $N_u$ to $U_i$ as the response nonce and $N_u$ is hidden in the messages $C_4$, $C_5$, where $C_4 = h(h(K_s \| e) \| AID_i \| N_u \| N_s)$, $C_5 = C_i^* \oplus h(N_u \| N_s)$. In this way, the proposed scheme can withstand replay attacks.

**Proposition 6** Withstanding user masquerade attacks.

**Proof.** Assuming a legitimate but malicious user $U_a$ wants to masquerade another legitimate user $U_i$, the proposed scheme withstands the masquerade attack even if $U_a$ has obtained the information $\{h(\cdot), E_i, V_i, N_i^*, FA_i^*\}$ stored in the mobile device of $U_i$, (the symbols of items are distinguished by corresponding subscript, the same below.) In addition, $U_a$ can obtain these information $\{h(\cdot), E_a, V_a, N_a^*, FA_a^*\}$ from his mobile device. Using these information, $U_a$ only can get $h(K_s \| e_a)$ by computing $E_a \oplus h(ID_a \| K_{ua})$ using his secret values $ID_a$ and $F_a$. However, a valid login message has to be generated in order to impersonate the legitimate user $U_i$. However, $U_a$ is not able to obtain the secret information $ID_i$ of $U_i$, this further protects the more secret values as $N_i$, $FA_i$ to be breached. Moreover, even if $U_a$ has obtained the key secret values $N_i$ and $FA_i$, he still fails to generate a valid login message. $U_a$ practically impossible to obtain the secret key $K_{ui}$ since the $F_i$ is belong to user only. As a result, $U_a$ cannot compute $AID_i^*$, $C_1$ and $C_2$. So, the proposed scheme is secure to user masquerade attacks.

**Proposition 7** Withstanding server spoofing attack.

**Proof.** Even if $U_a$ is a legitimate user and he can intercept all messages from a public channel, he still fails to masquerade as $S$ in proposed scheme. If $U_a$ wants to masquerade as $S$ to spoof $U_i$, he needs to generate a valid response message $\{C_3, C_4, C_5\}$, where $C_3 = N_s \oplus h(K_s \| e_i)$, $C_4 = h(h(K_s \| e_i) \| AID_i \| N_{ui} \| N_s)$, $C_5 = C_i^* \oplus h(N_{ui} \| N_s)$. However, $U_a$ is impossible to obtain $h(K_s \| e_i)$ even if he can extract all information stored in mobile device of $U_i$ and intercept all messages sent from $U_i$. This is due to the fact that $e_i$ is impossible to get. In addition, $U_a$ unable to obtain $C_i^*$. Moreover, even if $U_a$ has obtained the key secret values $e_i$ and $C_i^*$, he still fails to generate session key $SK$ and spoof user $U_i$. This is because he cannot obtain $R_i$ so that he cannot reconstruct the secret key value $X_{us}$ shared by both $U_i$ and $S$.

**Proposition 8** Withstanding man-in-the-middle attacks.

**Proof.** Suppose the user login message $\{AID_i^*, C_1, C_2\}$ has been intercepted by an attacker $U_a$ during the login phase, and he/she intends to perform a man-in-the-middle attack by modifying the $\{AID_i^*, C_1^{new}, C_2^{new}\}$ message. Note that $C_1 = N_u \oplus K = N_u \oplus h(K_s \| e)$ and $C_2 = h(K \| AID_i \| N_u) = h(h(K_s \| e) \| AID_i \| N_u)$. If $U_a$ can guess $h(K_s \| e)$ and $AID_i$ correctly, he can modify $C_1^{new} = N_u^{new} \oplus h(K_s \| e)$ and $C_2^{new} = h(h(K_s \| e) \| AID_i \| N_u^{new})$ and sends this message $\{AID_i^*, C_1^{new}, C_2^{new}\}$ to $S$ to complete the authentication at the server side. However, it's impossible to get both $AID_i$ and $h(K_s \| e)$ for an attacker $U_a$. As a result, $U_a$ has no way to modify properly all the transmitted messages between $S$ and $U_i$ during either the login or authentication phases. So, our scheme withstands man-in-the-middle attacks.

**Proposition 9** Withstanding known key secrecy attacks.

**Proof.** Suppose an attacker $U_a$ has obtained a session key $SK^*$ being communicated in previous communications, but it is still not possible for him to construct the current session key $SK = h(AID_i \| X_{us} \| N_s \| N_u)$ between $U_i$ and $S$. To construct the current session key, $U_a$ needs to get the nonce values $N_u$ and $N_s$ generated by $U_i$ and $S$. In our scheme, new nonce values are generated in each session. In addition, $N_u$ and $N_s$ cannot be computed without knowing the value $h(K_s \| e)$. In addition, $U_a$ also cannot obtain $AID_i$ and $X_{us}$ because they are computed in each session using the biometric and PUF of user. So, it is impossible to obtain these secret values simultaneously for an attacker. Therefore, our scheme is against the known key secrecy attacks.

**Proposition 10** Withstanding session-specific temporary information attacks.

**Proof.** Assume an attacker $U_a$ has managed to know the temporary information $N_u$ and $N_s$, and then he intends to construct the session key $SK$. However, the session key $SK = h(AID_i \| X_{us} \| N_s \| N_u)$ cannot be computed without knowing the value $X_{us}$. Since the value of $X_{us}$ is the shared secret of $U_i$ and $S$ and it must be generated by using user's PUF in each session. So, $U_a$ is not able to get it. This shows that the proposed scheme is secure from the session-specific temporary information attack.

**Proposition 11** Perfect forward secrecy.

**Proof.** Perfect forward secrecy ensures that $U_a$ cannot construct the session keys generated in previous sessions, even if the long term key $K$ of $U_i$ is stolen. We use the session nonce values $N_u$ and $N_s$ to compute the session key $SK = h(AID_i \| X_{us} \| N_s \| N_u)$ and use different nonce values in each session. Therefore, it is impossible to compute the session key $SK$ without knowing $N_u$ and $N_s$. Moreover, even if $U_a$ intercepts all the information communicated over the public channels, he/she cannot compute $SK$ because the values $X_{us}$ is not known to him/her. Even if he get the challenge $C_i$, he needs to use the $PUF_i$ equipped in mobile device of user to compute response $\hat{R}_i$ in each session and thus to obtain $X_{us}$. But all this is impossible. Therefore, $U_a$ cannot reconstruct the established session key. Hence our scheme has perfect forward secrecy.

**Proposition 12** Supports mutual authentication.

**Proof.** In our scheme, the user $U_i$ challenges the server $S$ by sending the message $C_1$ to $S$ and $S$ responds to the challenge by sending the message $C_4$ to $U_i$. On the other hand, the server $S$ challenges the user $U_i$ by sending the message $C_4$ to $U_i$ and $U_i$ responds to the challenge by sending the massage $C_7$ to $S$. It is clear that only the legitimate user $U_i$ with the correct biometric $F_i$ and PUF $PUF_i$ can successfully complete these tasks. Moreover, the proposed scheme withstands masquerade attack in both user side and server side, so it ensures mutual authentication.

**Proposition 13** Provision of session key agreement.

**Proof.** In proposed scheme, both user $U_i$ and server $S$ need to independently compute the session $SK = h(AID_i \| X_{us} \| N_s \| N_u)$ for subsequent communications. It was only when the session keys computed by $U_i$ and $S$ are identical that they can communicate securely using $SK$. Therefore, our scheme can provide session key agreement.

### A. Formal Analysis

We use the BAN logic to formally analyze the security features of our scheme. The BAN logic can be constructed based on some basic postulates and assumptions and it is commonly used in communication protocol analysis. Table II has summarized the various basic notations in our logic analysis used by the BAN logic.

TABLE II
BASIC NOTATIONS OF THE BAN LOGIC

| Basic notation | Meaning |
|---|---|
| $P \mid\equiv X$ | Principal $P$ believes the statement $X$ |
| $P \lhd X$ | Principal $P$ sees the statement $X$ |
| $P \mid\sim X$ | Principal $P$ once said the statement $X$ |
| $P \mid\Rightarrow X$ | Principal $P$ has jurisdiction over the statement $X$ |
| $\#(X)$ | Formula $X$ is fresh |
| $P \overset{K}{\longleftrightarrow} Q$ | $P$ and $Q$ use the shared key $K$ to communicate. |
| $\overset{K}{\mapsto} P$ | $K$ is a public key of $P$ |
| $P \overset{X}{} Q$ | Formula $X$ is a secret known only to $P$ and $Q$ |
| $\{X\}_K$ | Formula $X$ is encrypted using key $K$ |
| $\langle X \rangle_Y$ | Formula $X$ is combined with formula $Y$. |

The BAN logic uses the following logical postulates as formal rules:

R1. Message meaning rule 1: $\dfrac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$

R2. Message meaning rule 2: $\dfrac{P \mid\equiv P \overset{Y}{} Q, P \lhd \{X\}_Y}{P \mid\equiv Q \mid\sim X}$

R3. Nonce verification rule: $\dfrac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim (X)}{P \mid\equiv Q \mid\equiv X}$

R4. Jurisdiction rule: $\dfrac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$

R5: Freshness rule: $\dfrac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$

R6. Belief rule: $\dfrac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv (X)}$

Below is the initial security assumptions made for the protocols:

A1: $U_i \mid\equiv \#N_u$ A2: $S \mid\equiv \#N_s$

A3: $U_i \mid\equiv U_i \overset{h(K_s \| e)}{\longleftrightarrow} S$ A4: $S \mid\equiv U_i \overset{h(K_s \| e)}{\longleftrightarrow} S$

A5: $U_i \mid\equiv U_i \overset{X_{us}}{\longleftrightarrow} S$ A6: $S \mid\equiv U_i \overset{X_{us}}{\longleftrightarrow} S$

A7: $U_i \mid\equiv S \Rightarrow U_i \overset{SK}{\longleftrightarrow} S$ A8: $S \mid\equiv U_i \Rightarrow U_i \overset{SK}{\longleftrightarrow} S$

The following four security goals are expected to achieve the mutual authentication between a server $S$ and a user $U_i$:

G1. $U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$

G2. $S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$

G3. $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$

G4. $S \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$

The main steps to prove that the proposed protocol has achieved mutual authentication between $U_i$ and $S$ using above assumptions and rules are as follows:

S1. $U_i \lhd \langle AID_i, N_u, N_s \rangle_{h(K_s \| e)}$

Based on the assumed A3, the message meaning rule R1 can be applied along with S1 to yield the following:

S2. $U_i \mid\equiv S \mid\sim (AID_i, N_u, N_s)$

Moreover, we have the assumption A1, the freshness rule R5 applies and yields:

S3. $U_i \mid\equiv \#(AID_i, N_u, N_s)$

According to S2 and S3, the nonce-verification rule R3 applies and yields:

S4. $U_i \mid\equiv S \mid\equiv (AID_i, N_u, N_s)$

According to S4 and A1, A2, A5 and $SK = h(AID_i \| X_{us} \| N_u \| N_s)$, the freshness rule R5 and nonce-verification rule R3 apply and yield:

S5. $U_i \mid\equiv S \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$ **(Goal 1)**

According to S5 and A7, jurisdiction rule R4 applies and yields:

S6. $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$ **(Goal 3)**

S7. $S \lhd \langle U_i \overset{SK}{\longleftrightarrow} S, N_s \rangle_{X_{us}}$

According to A6 and S7, the message meaning rule R2 apply and yield:

S8. $S \mid\equiv U_i \mid\sim (U_i \overset{SK}{\longleftrightarrow} S, N_s)$

Moreover, we have the assumption A2, the freshness rule R5 applies and yields:

S9. $S \mid\equiv \#(U_i \overset{SK}{\longleftrightarrow} S, N_s)$

According to S8 and S9, the nonce verification rule R3 applies and yields:

S10. $S \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S, N_s)$

According to belief rule R6 and S10, we can obtain:

S11. $S \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$ **(Goal 2)**

According to S11 and A8, jurisdiction rule R4 applies and yields:

S12. $S \mid\equiv U_i \overset{SK}{\longleftrightarrow} S$ **(Goal 4)**

The above goals 1-4 clearly indicate the our scheme achieves the mutual authentication between *Ui* and *S*.

## V. PERFORMANCE ANALYSIS AND DISCUSSION

In this section, we will evaluate the performance of our scheme. Comparisons have been made with other related biometrics-based schemes to show the effectiveness and excellent security features of our scheme.

### A. Discussion on User's Biometric and Device's PUF

The biometrics are uniquely associated with an individual.

Fingerprint is one of the most important biometrics and it has stability, uniqueness and convenience. Unlike traditional password, it is hard to copy and forge. However, if you lose it, it is lost forever. So, great loss will be brought to these users once their fingerprint information are leaked. Many researchers tried everything to protect the fingerprint information from the attacker in the various attack environment. Obviously, these methods [5-8] stored fingerprint template directly into device are most at risk. In order to avoid the risk, the authors [9-15] store fingerprint template encrypted by hash function into devices. However, as discussed in Section I A, the fingerprints may change slightly over time and this subtle change may cause changes in the output of the sensitive secure one way hash function. Thus, these schemes are not realistic. In [17, 18], the fuzzy extractor technology are employed to extract the secret key and auxiliary information from fingerprint template and store them into device. However, the auxiliary information is stored in the device directly in their scheme, and thus cannot ensure the privacy of the users. In [26], the user's biometric thumb impression is input directly into the PUF as the challenge to generate the biometric key. It is unrealistic due to the slight variations in biometrics fingerprints. Furthermore, the user need to provide password in this scheme.

In our proposed scheme, the FE technology are employed to improve the robustness of mutual authentication and key agreement. The security and privacy of user's biometrics can be further improved by extracting the secret key and auxiliary information from fingerprint template using FE. Both the data extracted from fingerprint template are encrypted and stored into device. As described in Section II A, the robust stability of the FE algorithm are largely dependent on the Hamming distance between registered fingerprint template and authenticated fingerprint template. The distance need to be small enough in terms of the success of the proposed scheme. In fact the accuracy and reliability of the extracted fingerprint template depend on the quality of input fingerprint image. To meet this requirement, the input fingerprint quality need to be improved. Luckily we can obtain the desired fingerprint with high quality from the input fingerprint with noise by using fingerprint enhancement methods [34].

The proposed scheme has been designed without providing a password for the convenience of the users. In the proposed scheme, the user's mobile device needs to equip a PUF which can generate a shared secret key for the user and the server. To improve the robustness of the proposed scheme, we first extract a key from user's fingerprint template, then use this key to encrypt the challenge and store the encrypted challenge on the server side which the user knows nothing about it. When mutual authentication and key agreement phases are executed, the server sends the encrypted challenge to the user for the user to decrypt using his/her biometrics information. The user can get the challenge only when he imprint his fingerprint correctly. Then the user generates the exact response using the PUF equipped in his mobile device and extracts the secret key and an auxiliary vector from the response using $FE.Gen(\cdot)$. The user sends the encrypted auxiliary vector to the server and the server needs to decrypt it using its private key. The server then uses the stored response and the decrypted auxiliary vector using the $FE.Rec(\cdot)$ to generate the secret key. Only when the response-auxiliary pair matches, the secret keys generated by user and server are identical. Only in this way the session key can be generated to complete the mutual authentication and key agreement phases. By doing so, our scheme can provide robust authentication and key agreement scheme without using password. Comparison for protecting user's biometrics and password is provided in Table III.

Biometric technologies has achieved great progress in the recent years, and has been successfully applied in user authentication scheme to authenticate the user's identity. PUF is a kind of device-unique fingerprint, it provide an effective way to uniquely identify each device and to extract cryptographic keys used for strong device authentication. Taken together, the feasibility of such a secure two factors user authentication and key agreement scheme proposed in this paper can be verified.

### *B. Performance Comparisons*

The list of desired security properties is shown in Table IV. We compare our Bio-AKA scheme with state of the art biometrics-based schemes against this security properties. As can be observed from Table IV, our scheme is the only one that satisfies all security properties whilst other schemes are vulnerable to at least one or more security attacks.

To further analyze the computational performance of our scheme, a comparison on the computational costs in all phases between the proposed scheme and other relevant schemes has been conducted. The comparison results are summarized in Table V. In the Table V, $T_{PUF}$ represents the PUF operation cost, $T_m$ is the modular exponentiations calculation cost and $T_h$ the time complexity for hashing function. It is worth noting that although our scheme has slightly higher computational costs comparing to others [9-15], it provides the most comprehensive security features that can withstand the known various attacks with balanced computational efficiency to mutual authentication and key agreements. Furthermore, there is no use of password involved in our proposed scheme. Therefore, in terms of security, convenience and efficiency, the proposed scheme is the most appropriate and practical scheme amongst the related biometrics-based schemes [6, 9-15, 17, 18].

To more rigorously evaluate the performance of our scheme with respect to [6, 9-15, 17, 18], simulation of the cryptographic operations has been conducted. The user's device is a HTC One smartphone and the Server is an Intel Core i5-4300 machine. The execution time of the cryptographic operations has been estimated by using the JCE library [35] for all related schemes. In addition, the 128-bit arbiter PUF has been used for PUF operation and BCH code [36] has been adopted for FE.Gen (.) and FE.Rec (.) operations. According to our simulation, we found the following outcomes: each hash operation ($T_h$) takes 0.026 ms at the user's device and 0.011 ms at the server side; each PUF operation ($T_{PUF}$) takes 0.13 ms at the user's device; each FE.Gen (.) operation takes 2.67 ms at user's device and FE.Rec (.) operation takes 3.35 ms at the server; each modular operation ( $T_m$ ) takes 21.86 ms at user's device and 14.6 ms at the server. Now, based on the simulation results, the execution of the authentication phase takes 6.35ms in total where 2.93 ms was spent on the user's device ($5T_h$+$1T_{PUF}$+1FE.Gen $(\cdot)$) and

TA BLE III
COMPARISON FOR PROTECTING USER'S BIOMETRICS AND PASSWORD

| | [6] | [9] | [10] | [11] | [12] | [13] | [14] | [15] | [17] | [18] | Our |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | YES | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO |
| F2 | NO | YES | YES | YES | YES | YES | YES | YES | NO | NO | NO |
| F3 | NO | NO | NO | NO | NO | NO | NO | NO | YES | YES | YES |
| F4 | NO | NO | NO | NO | NO | NO | NO | NO | NO | NO | YES |
| F5 | YES | YES | YES | YES | YES | YES | NO | YES | YES | YES | - |

F1: Stores fingerprint template into user devices directly
F2: Stores fingerprint template encrypted by hash function into devices
F3: Stores the data extracted from fingerprint template using FE into devices
F4: No password required
F5: Provides change-password

TABLE IV
COMPARISON OF THE SECURITY PROPERTIES

| Security properties | [6] | [9] | [10] | [11] | [12] | [13] | [14] | [15] | [17] | [18] | Our |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Protects user anonymity | NO | NO | YES | YES | NO | NO | NO | YES | NO | YES | YES |
| Withstands online password guessing attack | YES | NO | NO | - | NO | NO | NO | YES | - | - | YES |
| Withstands offline password guessing attack | - | NO | NO | YES | NO | NO | NO | YES | YES | YES | YES |
| Withstands stolen mobile device/smart card attack | YES | YES | NO | YES | NO | NO | YES | - | YES | - | YES |
| Withstands insider attack | YES | YES | YES | YES | NO | NO | YES | - | YES | YES | YES |
| Withstands replay attack | YES | NO | YES | YES | NO | YES | NO | - | NO | YES | YES |
| Withstands user masquerade attack | YES | NO | NO | YES | NO | NO | NO | YES | YES | YES | YES |
| Withstands server spoofing attack | NO | NO | NO | YES | NO | NO | NO | YES | YES | - | YES |
| Withstands man-in-the-middle attack | YES | YES | YES | YES | NO | YES | NO | - | YES | - | YES |
| Withstands known key secrecy attack | YES | YES | YES | YES | YES | YES | - | - | YES | YES | YES |
| Withstands temporary information attack | NO | - | - | - | YES | - | - | - | NO | YES | YES |
| Achieves perfect forward secrecy | YES | - | - | YES | YES | - | YES | - | YES | YES | YES |
| Supports mutual authentication | NO | YES | YES | YES | NO | NO | NO | YES | YES | YES | YES |
| Provides session key agreement | YES | NO | YES | YES | NO | NO | NO | NO | YES | YES | YES |
| No time synchronization | NO | YES | YES | YES | YES | YES | YES | YES | YES | NO | YES |

TABLE V
COMPUTATIONAL COST COMPARISONS

| Scheme | Registration phase | Login phase | Authentication phase | | Total |
|---|---|---|---|---|---|
| | | | User | Server | |
| [6] | $1T_h$+$1T_m$ | $2T_h$+$2T_m$ | - | $1T_h$+$2T_m$ | $4T_h$+$5T_m$ |
| [9] | $3T_h$ | $3T_h$ | $2T_h$ | $3T_h$ | $11T_h$ |
| [10] | $4T_h$ | $3T_h$ | $3T_h$ | $5T_h$ | $15T_h$ |
| [11] | $5T_h$ | $4T_h$ | $3T_h$ | $6T_h$ | $18T_h$ |
| [12] | $3T_h$ | $2T_h$ | $2T_h$ | $3T_h$ | $10T_h$ |
| [13] | $3T_h$ | $2T_h$ | $3T_h$ | $5T_h$ | $13T_h$ |
| [14] | $3T_h$ | $3T_h$ | $2T_h$ | $4T_h$ | $12T_h$ |
| [15] | $4T_h$ | $3T_h$ | $2T_h$ | $5T_h$ | $14T_h$ |
| [17] | $3T_h$+1FE.Gen(·) | $3T_h$+$1T_m$+1FE.Rec(·) | $3T_h$+$1T_m$ | $5T_h$+$2T_m$ | $14T_h$+$4T_m$+1FE.Gen(·)+1FE.Rec(·) |
| [18] | $3T_h$+$2T_m$+1FE.Gen(·) | $3T_h$+$2T_m$+1FE.Rec(·) | $2T_h$+$1T_m$ | $4T_h$+$3T_m$ | $12T_h$+$8T_m$+1FE.Gen(·)+1FE.Rec(·) |
| Our | $7T_h$+$1T_{PUF}$+1FE.Gen(·) | $6T_h$+1FE.Rec(·) | $5T_h$+$1T_{PUF}$+1FE.Gen(·) | $7T_h$+1FE.Rec(·) | $25T_h$+$2T_{PUF}$+2FE.Gen(·)+2FE.Rec(·) |

3.42 ms was spent on the server (7Th+1FE.Rec(·)). This is significantly less than the time reported in [6], [17], and [18] as these schemes are based on the computationally expensive modular operation. On the other hand, even though, the

schemes presented in [9-15] takes less time than the proposed scheme, but according to Table III and Table IV they cannot ensure many imperative security properties.

## VI. Conclusion

In the paper, following the analysis of the pitfalls of the existing biometrics-based remote user authentication schemes, we proposed a new Bio-AKA scheme which combines PUF and fingerprint biometric to provide a secure two factors user authentication and key agreement scheme. We explored the inherent security properties of PUFs and biometrics and how our scheme is capable of achieving the desired security characteristics. Without storing biometric information on the device, we have managed to develop a scheme that can completely eliminate the risk of leaking user's biometric information. Fuzzy extractor is employed to improve the robustness of the proposed scheme. Performance analyses have been performed to prove our scheme can withstand various known security attacks including online and offline password guessing attach, stolen mobile device and smart card attack, replay attack, user/server masquerade attack, man in the middle attack, known key secrecy attack, temporary information attack, and insider attack. Meanwhile, our scheme protects user anonymity, provides session key agreement, supports mutual authentication and achieves perfect forward secrecy. The scheme is robust and remarkably convenient without the need for password. Furthermore, the scheme can still provide security even for some extreme situations such as the mobile device being stolen by adversary attacker or the messages being intercepted over insecure channels. All the above features have shown the validity of our scheme for practical real life applications for remote user authentication.

## References


[1] L. Lamport, "Password authentication with insecure communication," Communications of the ACM, vol. 24, no. 11, pp. 770-772, 1981.

[2] M.S. Hwang, L.H. Li, "A new remote user authentication scheme using smart cards," IEEE Transactions on consumer Electronics, vol. 46, pp. 28-30, 2000.

[3] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE transactions on information theory, vol. 31, no. 4, pp. 469-472, 1985.

[4] C.K. Chan, L.M Cheng, "Cryptanalysis of a remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 46, no. 4, pp. 992-993, 2000.

[5] J.K. Lee, S.R. Ryu, K.Y. Yoo, "Fingerprint-based remote user authentication scheme using smart cards," Electronics Letters, vol. 38, no. 12, pp. 554-555, 2002.

[6] C.H. Lin, Y.Y. Lai, "A flexible biometrics remote user authentication scheme," Computer Standards & Interfaces, vol. 27, no. 1, pp. 19-23, 2004.

[7] M.K. Khan, J. Zhang, "Improving the security of a flexible biometrics remote user authentication scheme," Computer Standards & Interfaces, vol. 29, no. 1, pp. 82-85, 2007.

[8] M.K. Khan, J. Zhang, X. Wang, "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," Chaos, Solitons & Fractals, vol. 35, no. 3, pp. 519-524, 2008.

[9] C.L. Chen, C.C Lee, C.Y. Hsu, "Mobile device integration of a fingerprint biometric remote authentication scheme," International Journal of Communication Systems, vol. 25, no. 5, pp. 585-597, 2012.

[10] T.T. Truong, M. Tran, A.D. Duong, "Robust mobile device integration of a fingerprint biometric remote authentication scheme," In 2012 IEEE 26th International Conference on Advanced Information Networking and Applications. IEEE. Fukuoka, Japan, pp. 678-685, 2012.

[11] M.K. Khan, S. Kumari, M.K. Gupta, "More efficient key-hash based fingerprint remote authentication scheme using mobile device," Computing, vol. 96, no. 9, pp. 793-816, 2014.

[12] C.T. Li, M.S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," Journal of Network and computer applications, vol. 33, no. 1, pp. 1-5, 2010.

[13] A.K. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," IET Information Security, vol. 5, no. 3, pp. 145-151, 2011

[14] Y.H. An, "Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards", Journal of Biomedicine and Biotechnology, vol. 2012, pp. 1-6, 2012.

[15] M.K. Khan, S. Kumari, "An improved biometrics-based remote user authentication scheme with user anonymity," Journal of Biomedicine and Biotechnology, vol. 2013, pp. 1-9, 2013.

[16] S. Ibjaoun, A.A. El Kalam, V. Poirriez, Ouahman, et al., "Analysis and enhancements of an efficient biometric-based remote user authentication scheme using smart cards," In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). IEEE. Agadir, Morocco, pp. 1-8, 2016.

[17] X. Li, J. Niu, Z. Wang, et al., "Applying biometrics to design three‐factor remote user authentication scheme with key agreement," Security and Communication Networks, vol. 7, no. 10, pp. 1488-1497, 2014.

[18] A. Chaturvedi, D. Mishra, S. Jangirala, et al., "A privacy preserving biometric-based three-factor remote user authenticated key agreement scheme," Journal of Information Security and Applications, vol. 32, pp. 15-26, 2017.

[19] 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, vol. 38, no. 1, pp. 97-139, 2008.

[20] J. Delvaux, D. Gu, I. Verbauwhede, et al., "Efficient fuzzy extraction of PUF-induced secrets: Theory and applications," In International Conference on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg. pp. 412-431, 2016.

[21] A.K. Jain, K. Nandakumar, A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," Pattern Recognit. Lett., vol. 79, pp. 80-105, 2016.

[22] S. Barra, A. Castiglione, M. De Marsico, et al., "Cloud-Based Biometrics (Biometrics as a Service) for Smart Cities, Nations, and Beyond," IEEE Cloud Computing, vol. 5, no. 5, pp. 92-100, 2018.

[23] K. Zhou, J. Ren, "PassBio: Privacy-Preserving User-Centric Biometric Authentication," IEEE Transactions on Information Forensics and Security, vol. 13, no. 12, pp. 3050-3063, 2018.

[24] Y. Wang, J. Wan, J. Guo, Y. Cheung and P. C. Yuen, "Inference-Based Similarity Search in Randomized Montgomery Domains for Privacy-Preserving Biometric Identification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 7, pp. 1611-1624, 2018.

[25] M.N. Aman, K.C. Chua, B. Sikdar, "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1327-1340, 2017.

[26] P. Gope, A. K. Das, N. Kumar and Y. Cheng, "Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks," in IEEE Transactions on Industrial Informatics. doi: 10.1109/TII.2019.2895030, 2019.

[27] G.E. Suh, S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in Proc. IEEE/ACM DAC, San Diego, CA, USA, Jun. 2007, pp. 9-14.

[28] M. Burrows, M. Abadi, R.M. Needham, "A logic of authentication," Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, vol. 426, no. 1871, pp. 233-271, 1989.

[29] H.S. Rhee, J.O Kwon, D.H Lee, "A remote user authentication scheme without using smart cards," Comput Stand Interfaces, vol. 31, no. 1, pp. 6-13, 2009.

[30] P. Kocher, J. Jaffe, B. Jun, "Differential power analysis," In: Proceedings of Advances in Cryptology, Santa Barbara, pp. 388-397, 1999.

[31] T.S. Messerges, E.A Dabbish, R.H Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Trans Comput, vol. 51, no. 5, pp. 541-552, 2002.

[32] M. La Polla, F. Martinelli and D. Sgandurra, "A survey on security for mobile devices," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 446-471, 2013.

[33] H.H. Kilinc, T. Yanik, "A survey of SIP authentication and key agreement schemes," IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1005-1023, 2014.

[34] S.F. Ding, W.X. Bian, T. Sun, et al., “Fingerprint enhancement rooted in the spectra diffusion by the aid of the 2D adaptive Chebyshev band-pass filter with orientation-selective”, Information Sciences, vol. 415, pp. 233-246, 2017.

[35] Java Cryptography Architecture (JCA), Apr. 2017, [online] Available: http://docs.oracle.com/javase/6/docs/technotes/guides/crypto/CrypoSpec.html.

[36] Y. Dodis et al., "Fuzzy extractors: How to generate strong keys from biometrics and other noise data", SIAM J. Compt, vol. 38, no. 1, pp. 97-139, 2008.