A secure user anonymity-preserving biometrics and PUFs-based multi-server authentication scheme with key agreement in 5G Networks

Deqin Xu, Weixin Bian, Qingde Li, Dong Xie, Jun Zhao, Yao Hu

Abstract—The 5G networks can provide high data rates, ultra-low latency and huge network capacity. In 5G networks environment, the popularity of the Internet of Things (IoT) has led to a rapid increase in the amount of data. Multi-server distributed cloud computing technology provides an excellent solution to alleviate network pressure caused by the rapid growth of data. However, this technology serves as a two-edged weapon, which not only makes various IoT applications possible, but also brings growing concerns for user privacy and ever pressing security challenges. To ensure the high security of 5G network-based applications, we design a secure user anonymity-preserving biometrics and PUFs-based multi-server authentication scheme with key agreement. In our method, we make full use of the inherent security features of user fingerprint and smart device PUF to design a secure multi-server authentication scheme with key agreement in 5G Networks. The proposed scheme is able to resist recognized attacks and its robustness has been verified by security analysis.

***Index Terms*—Multi-server; Mutual authentication; Biometrics; Physically unclonable functions.**

I. INTRODUCTION

In recent years, the development of fifth-generation (5G) technology has attracted increasing interest from researchers. It has become the main driving force for the growth of IoT-related applications [1]. Today, the dream of fifth-generation (5G) wireless networks has become a reality in some parts of the world and will expand rapidly. Compared with the existing 4G technology, the new generation technology provides a much higher communication speed, bandwidth and lower communication delay. 5G has higher energy efficiency requirements than 4G in terms of telemedicine information system (TMIS) [2], big data processing and analysis [3] and high reliability, low latency, and advanced artificial intelligence [4]. Therefore, many single-server structures are not suitable for 5G networks. Then some scholars proposed to use a multi-server architecture in the 5G network environment [5-7].

Typically, a multi-server architecture in the 5G network environment consists of users, a group of servers, and a trusted registration center (*RC*), which is responsible for registering users and servers. The Fig 1 shows the system model in a multi-server architecture in the 5G network. A multi-server architecture, where users with 5G supported devices acquire online services from multiple servers through a base station. Compared to traditional single server architectures, multi-server architectures can better balance loads, expand geographic coverage, and improve system security performance [5]. Besides, in a multi-server authentication environment, users only need to register with *RC* once to access multiple servers, which provides great convenience for users.

However, the multi-server authentication scheme is fragile. The openness of the wireless channel and the Internet imposes a serious threat to user privacy and confidential information (during online communication). In this environment, online communication is vulnerable to attack by attackers, which leads to information leakage. External attackers and dishonest internal users (internal attackers) can collect useful information (related to registered users and servers) from communication links, and then use this information to impersonate legitimate users/servers illegally. Much of the information transmitted in the multi-server authentication environment is private and sensitive. How we ensure the security of this type of data is critical.

To solve this problem, early authentication schemes often used a two factor authentication scheme based on password and smart card. Because two-factor authentication relies on knowledge and ownership factors, it is considered more secure than commonly used single-factor authentication. However, people soon realize that once the smart card is stolen, the information in the smart card is easily leaked, so the authentication scheme using smart card is not reliable. In addition, the researchers' past experience shows that the smart card itself is a resource-constrained device. Therefore, it is almost difficult to develop a correct authentication scheme and sufficient security features [8]. However, there exist the security issues in these authentication schemes based on password and smart card, such as password guessing and smart card information loss, many researchers have begun to adopt various biometric-based authentication schemes [9, 10].

In recent years, some mutual authentication schemes based physical unclonable function (PUF) have been proposed in

The work is partially supported by the NSFC (Nos. 61976006 and 61801004), NSF_AH (No. 2108085MF206). (*Corresponding author: W.X. Bian*, e-mail: bwx2353@ahnu.edu.cn)

D.Q. Xu, W.X. Bian, D. Xie, J. Zhao and Y. Hu are with the School of Computer and Information, Anhui Normal University, Wuhu, 241002, China and Anhui Province Key Laboratory of Network and Information Security, Wuhu, 241002, China. Q. Li, is with the Department of Computer Science and Technology, University of Hull, Hull, UK.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3486005

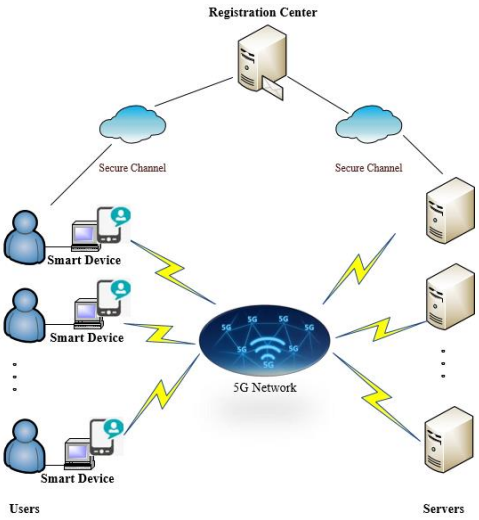

Fig. 1. Multi-server architecture model

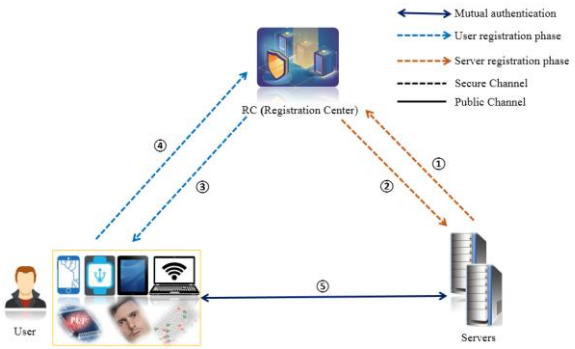

Fig. 2. Network model of proposed scheme

Internet of Things (IoT) systems [11-14]. Just like human biometric technology (such as fingerprints), each *PUF* has its own unique physical characteristics varying from production to production. Therefore, it is almost impossible to predict and produce a clone of *PUF*. It has been shown that by using PUF-enabled devices, it is able to achieve mutual authentication securely in a remote authentication scheme. Since this "biological feature" cannot be copied or cloned, PUF-based authentication schemes are of high degree of robustness and reliability. Obviously, the remote user authentication scheme that combines user biometric features with device PUF will enhance the security of the identity authentication system. Gope et al. [11] directly uses biometric features as a challenge for PUF, incorporating the corresponding response as part of the authentication key. However, due to the influence of device, physiological, behavioral, and environmental factors, there are differences in the biometric features collected each time. This difference may exceed the noise tolerance range of PUFs, and directly using biometric features as challenge for PUFs may result in different responses to the same biometric. Therefore, directly using biometric features as challenge for PUF to generate keys is unreliable. Zhang et al. [12] and Zahednejad et al. [13] use fuzzy extractors to extract biometric keys from biometric features and use them as inputs for PUFs to overcome the aforementioned problems. In their methods, the response of PUF and face feature are used as parameters to generate cancelable biometric. In our scheme, we use PUF to verify the uniqueness of the smart device and use biometric features to verify the user's legitimacy. This bidirectional verification can ensure the security of the system and provide the key security properties. In order to reduce the computational overhead and the mutual authentication without the need for an online registration center, we use Self-certified public key cryptography. Fig 2 shows the network model of our proposed scheme. The main contributions of this work are summarized below:

(1) We design a two factors user authentication and Self-certified public key agreement scheme combining *PUF* and fingerprint biometric, which has high security, convenience and efficiency.

(2) The biometrics and PUFs are used to ensure the uniqueness of the user and smart device respectively, which can ensure the physical security of the proposed scheme.

(3) In our entire authentication protocol process, although we discarded the traditional password, the overall security of our protocol remains very high.

II. Related Work

In 1981, Lamport [15] first proposed an insecure password-based authentication scheme. Lamport's system stores all passwords in a table for authenticating the legitimacy of users. It could be attacked by modifying the password table. Later, many researchers published many improved password-based authentication schemes for solving this problem [16-18]. The most commonly used authentication schemes are based on passwords and tokens. However, these two traditional authentication methods have certain limitations, such as passwords may be forgotten or leaked, and smart cards may be lost or stolen. When the attacker obtains the user's password and smart card, the attacker can perform a series of attacks, such as updating the user's password and logging into the system as a legitimate user. The authentication system relying on passwords and smart cards is not secure enough. Compared with passwords and smart cards, biometric technology-based authentication by using, for example, fingerprints, faces, iris, etc., is not only simple and easy to use, and but also has high recognition rate, high reliability, unique, cannot be copied, lost or forgotten. Besides, it gives people the possibility that using the measurable biological and behavioral characteristics to design the reliable and automatic authentication scheme. In 2002, Lee et al. [19] proposed a fingerprint-based remote user authentication scheme using smart cards.

Although the authentication scheme based on biometrics alleviates the password problem, there is an obvious disadvantage of the single-server authentication scheme concerning registration. If a new user wants to use a large number of network services, they must register on these servers. User registration is very troublesome, which wastes not only

user time, but also server resources. To overcome the shortcomings of single-server authentication schemes, many researchers have proposed multiple multi-server authentication schemes. In 2001, Li et al. [20] first proposed a multi-server authentication scheme using neural network and asymmetric key cryptography. Due to high communication and computation overhead, it is impractical for small mobile devices. After that, a variety of multi-server authenticated key agreement protocols incorporating both symmetric cryptography [21] and asymmetric cryptography [22] have been proposed over the last two decades.

Recently, many researchers have turned their attention to multi-server authentication [23-30]. Haq et al. [23] comprehensively summarizes the proposed authenticated key agreement protocols for multi-server architecture. In 2013, Liao et al. [24] presented a multi-server remote user authentication scheme using Self-certified public keys; however, Hsieh et al. [25] demonstrated that Liao's scheme is vulnerable to trace attack and also require a permanent secure channel to update user information on each server. Biometric recognition is closely related to personal physical characteristics, so it has become the first choice for secure and reliable authentication. Chaturvedi et al. [26] proposed a secure biometric-based authentication scheme for multi-server environment using smart card. The scheme proposed in [27], user can change biometrics and password at any time locally. Chatterjee et al. [28] used the Chebyshev chaotic map and biometric verification along with password verification for authorization and access to various application servers in multi-server environment. Feng et al. [30] proposed a biometrics-based authentication scheme with key distribution for the mobile multi-server environment, and use smart card and elliptic curve cryptosystem to complete user authentication.

Ying et al. [5] proposed an efficient Self-certified public keys-based protocol employing only elliptic curve-based point multiplication operations. But, Ying's scheme remains vulnerable to stolen smart card attack, user impersonation attack and server impersonation attack. To solve this problem, we design a two factors user authentication and key agreement scheme combining PUFs and biometric. Compared with passwords and encryption keys, the use of biometric identification has completive advantages, such as resistance to copying or guessing, not easy to be forged, and users not needing to remember or not having to worry about losing the key. 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. So, it can ensure the uniqueness of the smart device.

The rest of the paper is organized as follows. Section III introduces some preliminary knowledge. In Section IV, we give a brief review of Ying's scheme. Section V describes the cryptanalysis of Ying's scheme. The proposed three-factor remote user authentication and key agreement scheme and corresponding security analysis are presented in Sections VI and VII, respectively. Finally, we draw our conclusion in Section VIII.

III. PRELIMINARY

This section gives a brief introduction of some preliminary knowledge.

A. Encrypted One-way Hash Function

A one-way collision-resistant hash function $h: \{0,1\}^* \rightarrow \{0,1\}^n$ is considered as a deterministic algorithm, which takes a binary string ($x \in \{0,1\}^*$) of any length as input and outputs a binary string ($h(x) \in \{0,1\}^n$) of fixed length n. A one-way encrypted hash function is a special hash function with the following properties:

- The hash function is essentially decisive. Under the action of the hash function, the same input message outputs the same hash value.
- For any input x, it is difficult to find an x' such that $h(x) = h(x')$. This property is referred to as weakly collision resistance.

Finding a pair of inputs (x, y), with $x \neq y$, so that $h(x) = h(y)$ is computationally infeasible. This property is referred to as strong collision resistance.

B. Fuzzy Extractor

The fuzzy extractor can reliably extract nearly uniform random value K from the input biometrics BIO. This extraction method is fault-tolerant, with the help of auxiliary information hd, even if the input biometrics changes slightly, the K' value generated will be the same as the original value K, as long as the distance between K' and K is in the Hamming distance. According to Kumari et al. [29], a fuzzy extractor includes two effective random processes: "generation" and "recovery" (*FE.Gen* and *FE.Rec*). *FE.Gen* is a probabilistic key generation approach, which on biometric input *BIO* outputs an "extracted" string K_u and an auxiliary string hd, such that $(K_u, hd) = FE.Gen(BIO)$. *FE.Rec* is a deterministic reconstruction strategy, the key K_u' from the noisy input BIO' and the auxiliary data hd, is effectively recovered, $K_u' = FE.Rec(BIO', hd)$. When the distance between K_u' and K_u is within the Hamming distance, we consider these two values to be equal.

C. PUF

Physical unclonable function (*PUF*) is a kind of hardware function realization circuit that depends on chip characteristics, which has uniqueness and randomness. It is an integrated circuit which takes a string of bits as an input challenge and produces a string of bits called the response. The distinctive feature of *PUF* is that it relies on a physical basis, which makes it impossible to regenerate *PUF* using cryptographic primitives. In addition, the term "physically unclonable" indicates that it is computationally infeasible or difficult to generate a physically identical *PUF* [31]. Using *PUF* in a smart device can make the device have its own unique "fingerprint" that cannot be cloned or copied. The response R of the *PUF* to the challenge C can be expressed as: $R=PUF(C)$. A *PUF* is said to be secure if it meets the following requirements: If an input C is given the same *PUF* multiple times, it will produce the same response R with a very high probability; If the same input C is given to different *PUFs*, the response obtained by each *PUF* is very different, and there is a high probability. Let us assume $P = \{PUF_1(\cdot), \cdots, PUF_M(\cdot)\}$ as

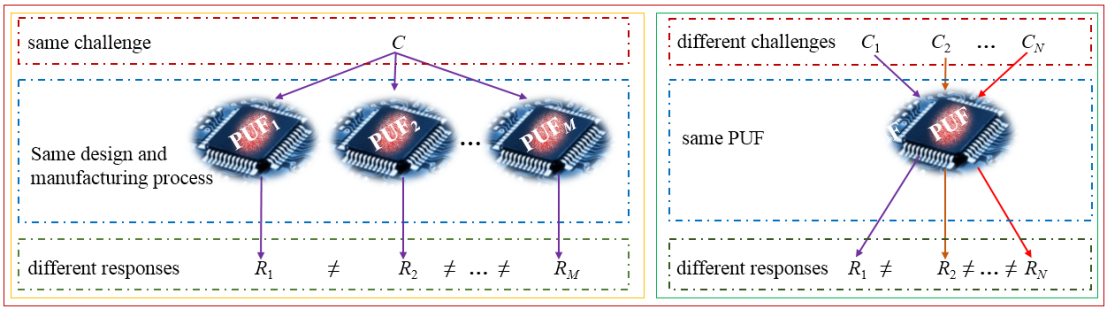

Fig. 3. The physical properties of PUFs

a set of PUFs and $C=\{C_1,\cdots,C_N\}\, s.t. \forall nC_n \in \{0,1\}^k$ as the set of challenges, then a $(d,h,l,\lambda,\varepsilon)$-secure PUF needs to meet the following requirements:

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

$$\mathrm{P_r}\left[d_H(PUF_m(C_p),PUF_m(C_q))>d\right]\geq 1-\varepsilon, \quad 1\leq m\leq M,1\leq p,q\leq N\wedge p\neq q \tag{2}$$

$$\mathrm{P_r}\left[\hat{H}_\infty(PUF_m(C_p),PUF_n(C_q))>\lambda\right]\geq 1-\varepsilon, \quad 1\leq m,n\leq M\wedge m\neq n;1\leq p,q\leq N\wedge p\neq q \tag{3}$$

Where d_H is the Hamming distance and $\hat{H}_\infty$ is the min-entropy of the PUF output. Fig 3 illustrates the principle.

D. Elliptic Curve Cryptography (ECC)

Compared with traditional cryptosystems such as DSA, DH and RSA, elliptic curve cryptosystem provides a more effective cryptosystem [32]. So, *ECC* has been widely used in authentication schemes for the multi-server 5G environment. Let $E_p\ (a, b)$ be a non-singular elliptic curve $y^2=x^3+ax+b\ (mod\ p)$ over a finite field F_p of prime order $p>3$, where, $(a, b)\in F_p$ and $4a^3+27b^2\neq 0(mod\ p)$. Given a random integer $a\in F_p^*$ and a point $P\in E_p\ (a, b)$, the elliptic curve point multiplication $a\cdot P$ over $E_p\ (a, b)$ is defined as $a\cdot P=P+P+...+P$ (a times). The security of *ECC* relies on the following two intractable problems:

- Elliptic Curve Discrete Logarithm Problem (ECDLP): For given Q_1, $Q_2\in E_p$, it is computationally hard to compute the scalar k such that $Q_2=k\,Q_1$.
- Elliptic Curve Diffie-Hellman Problem (ECDHP): For given P, xP, $yP\in E_p$, it is computationally hard to compute $xyP\in E_p$ without the knowledge of x or y.

IV. A Review of Ying's Scheme

In the section, we briefly introduce Ying et al.'s protocol [5]. We provide the definitions of notations appearing in the remaining of this work in Table I.

A. Setup Phase

In this phase, RC selects two large prime numbers p, q and a non-regular elliptic-curve $E: y^2=x^3+ax+b\ mod\ p$, where $a,b\in F_p$, $4a^3+27b^2\neq 0$. Let O be the infinity, P be the generator point of E with order q of the group G (where $P\neq O$). *RC* chooses a random number $S_{RC}\in Z_q^*$ as the system private key and the system public key is then set as $Pub_{RC}=S_{RC}.P$. At the same time, RC generates two one-way hash function: $H_1:\{0,1\}^*\rightarrow Z_q$, $H_2:\{0,1\}^*\times G\rightarrow Z_q^*$. Finally, *RC* publishes parameters $\{P, p, q, Pub_{RC}, H_1, H_2\}$ and keeps S_{RC}.

TABLE I
Definition of Notations

Symbol	Description
p,q	Two large prime numbers
P	Generator of the group G
G	Elliptic curve points group
S_{RC}	Private key of *RC*
Pub_{RC}	Pbulic key of *RC*
S_j	j^{th} server
U_i	i^{th} user
$PUF(.)$	Physical unclonable function
ID_{ui}	Identity of user U_i
ID_{sj}	Identity of server S_j
BIO_{ui}	Biometric data of user U_i
b_i, a_j, n_1, n_2, n_3	Random Numbers
$FE.Gen(.)$	Fuzzy extractor probabilistic key generation function
$FE.Rec(.)$	Fuzzy extractor deterministic reconstruction function
K_{ui}	Biometric key
RC	Trusted registration center
SK_{ij}	Session key between user U_i and sever S_j
$h(.)$	A one-way hash function
$\oplus$	Bitwise xor operator
$\|\|$	Concatenation operator
$\mathcal{A}$	An adversary

B. User Registration Phase

Step 1: U_i selects ID_{ui}, PW_{ui}, c_0 and computes $\beta_{ui}=H_1(ID_{ui}||PW_{ui}||c_0)$. Next, U_i sends $\{IDui, \beta_{ui}\}$ to *RC*.

Step 2: *RC generates* a random number $c_1\in Z_q^*$ and computes $A_{ui}=c_1.P$, $\emptyset_{ui}=H_2(\beta_{ui}||A_{ui})$, $N_{ui}=A_{ui}\oplus H_1(ID_{ui})$ and $M_{ui}=H_2(A_{ui}).S_{RC}mod\ q+c_1$.Then, *RC* sends $\{\emptyset_{ui}, N_{ui}, M_{ui}\}$ to U_i.

Step 3: U_i *stores* $\{\emptyset_{ui}, N_{ui}, M_{ui}, c_0\}$ into its smart card and finishes the registration.

C. Server Registration Phase

Step 1: S_j sends $\{ID_{sj}\}$ to RC.

Step 2: *RC* selects $c_2\in Z_q^*$ and computes $B_{sj}=c_2.P$ and $D_{sj}=H_2(ID_{sj}||B_{sj}).S_{RC}mod\ q+c_2$. Then, S_j sends $\{B_{sj}, D_{sj}\}$to S_j.

Step 3: Subsequently, S_j computes $D_{sj}.P$, and check $D_{sj}.P=?\,B_{sj}+H_2(ID_{sj}||B_{sj}).Pub_{RC}$. If it holds, S_j stores $\{B_{sj}, D_{sj}\}$, and then finishes the registration.

D. Mutual Authentication Phase

Step 1: U_i inputs ID_{ui}^* and PW_{ui}^*. Then U_i computes $\beta_{ui}^* = H_1(ID_{ui}^* || PW_{ui}^* || c_0)$, $A_{ui}^* = N_{ui} \oplus H_1(ID_{ui}^*)$ and validates U_i's identity and password by verifying $\emptyset_{ui} =? H_2(\beta_{ui}^* || A_{ui}^*)$.

Step 2: If it holds, U_i selects c_3, and computes $DID_{ui} = H_1(ID_{ui} || c_3)$. Then U_i chooses $c_4 \in Z_q^*$ and computes $F_{ui} = c_4.P$, $R_{ui} = H_2(DID_{ui} || F_{ui})$ and $\sigma_{ui} = M_{ui} + R_{ui}.c_4 mod\ q$. U_i transmits message $\{\sigma_{ui}, DID_{ui}, A_{ui}^*, F_{ui}\}$ to S_j over a public channel.

Step 3: S_j receives it and verifies $\sigma_{ui}.P =? H_2(A_{ui}^*).Pub_{RC} + A_{ui}^* + H_2(DID_{ui} || F_{ui}).F_{ui}$. If it does not hold, the verifier rejects the message.

Step 4: After U_i passes the authentication, S_j selects $c_5 \in Z_q^*$ and computes $F_{sj} = c_5.P$, $R_{sj} = H_2\left(DID_{ui} \left|\left| ID_{sj} \right|\right| B_{sj}\right)$ and $\sigma_{sj} = D_{sj}.S_{RC}\ mod\ q + R_{sj}.c_5$. Subsequently, S_j sends $\{\sigma_{sj}, B_{sj}, ID_{sj}, F_{sj}\}$ to U_i.

Step 5: U_i verifies whether $\sigma_{sj}.P = H_2(ID_{sj} || B_{sj}).Pub_{RC} + B_{sj} + H_2\left(DID_{ui} \left|\left| ID_{sj} \right|\right| B_{sj}\right).F_{sj}$.

V. Cryptanalysis of Ying's Scheme

In this section, we discuss the security drawbacks in Ying's remote user authentication protocol for multi-server 5G networks. In the login phase and authentication phase, the information exchange between the user and the server is carried out under an insecure channel. We assume that the attacker can completely control the insecure channel, in other words, the attacker can eavesdrop, intercept, delete and tamper with any transmitted messages in the insecure channel. Besides, as having been addressed in the work from Kocher et al. [33] and Messerges et al. [34], the confidential information stored in the smart cards can be extracted by side channel attacks by, for instance, physically monitoring its power consumption. Therefore, once the smart card is stolen, the secret information registered by the user during the registration phase will be stolen by the attacker, and the security of the entire protocol will no longer exist. In the section below, we analyze in detail the security flaws of the Ying's scheme in different scenarios.

A. Incorrect Mutual Authentication Phase

Mutual authentication phase in Ying's scheme, the authors use the private key S_{RC} of RC to compute the signature of server $\sigma_{sj} = D_{sj}.S_{RC}\ mod\ q + R_{sj}.c_5$. But in fact, one server registered in RC never knows the private key of RC. So, the server will not be able to compute the signature σ_{sj} of server in Ying's scheme. The signature of server σ_{sj} should be modified to meet the authentication requirement, and it should be rewritten as $\sigma_{sj} = D_{sj}\ mod\ q + R_{sj}.c_5$ based on the original Ying's scheme.

B. Without Session Key Agreement

Ying's scheme does not support the session key agreement between the user U_i and the server S_j. In some cases, after a successful authentication phase, some secret messages may be communicated between the user and the server. These messages should be encrypted to provide confidentiality of the transmitted data, such as online transaction payment. In order to provide this confidential communication, an agreed session key is required. However, in Ying's scheme, the establishment of the session key is not shown. If the solution does not have a session key, then the user U_i and the server S_j will need to perform some additional methods to generate and share the session key, which will inevitably increase the total cost of calculation and communication.

C. Smart Card Stolen or Lost Attack

Smart card stolen or lost attack is a type of attack when the user's smart card is stolen or lost. In this case, an attacker $\mathcal{A}$ can extract the information stored in the smart card and can easily change the password of the smart card, or can guess the password of the user by using password guessing attacks. Then $\mathcal{A}$ can impersonate the user to login to the system.

In Ying's scheme, if an attacker $\mathcal{A}$ steals the user U_i's smart card, then the information $(\phi_{ui}, N_{ui}, M_{ui}, c_0)$ stored in the smart card can be derived by $\mathcal{A}$. What is more, $\mathcal{A}$ can also get login request information $Msg1 = \{\sigma_{ui}, DID_{ui}, A_{ui}^*, F_{ui}\}$ through interception. $\mathcal{A}$ now can carry out the off-line identity guessing attack to guess a valid identity ID_{ui} as following:

Step 1: $\mathcal{A}$ can get $H_1(ID_{ui}^*)$ by computing $N_{ui} \oplus A_{ui}^* = N_{ui} \oplus N_{ui} \oplus H_1(ID_{ui}^*) = H_1(ID_{ui}^*)$.

Step 2: $\mathcal{A}$ guesses ID_{ui}' and then compare $H_1(ID_{ui}') =? H_1(ID_{ui}^*)$. Repeat the step until correct match is found, then $\mathcal{A}$ succeeded to obtain the identity ID_{ui} of U_i.

Once the identity of U_i is obtained to $\mathcal{A}$ using the above steps, $\mathcal{A}$ now can launch offline password guessing attack as following:

Step 1: $\mathcal{A}$ can get β_{ui} from $\phi_{ui} = H_2(\beta_{ui} || A_{ui})$.

Step 2: $\mathcal{A}$ guesses PW_{ui}' and compute $\beta_{ui}' = H_1(ID_{ui} || PW_{ui}' || c_0)$, then compare $\beta_{ui}' =? \beta_{ui}$. Repeat the step until correct match is found, then $\mathcal{A}$ succeeded to extract the password PW_{ui} of U_i.

Therefore, when the user's smart card is stolen by $\mathcal{A}$, $\mathcal{A}$ not only obtain the user's identity ID_{ui} but also get password PW_{ui} through an offline guessing attack. If $\mathcal{A}$ successfully guesses the identity and password of user, the harmfulness is apparent to the user. In Ying's scheme, the password of user for different severs is the same. Therefore, once $\mathcal{A}$ get the identity and password of user, $\mathcal{A}$ can access all services registered in RC.

D. User Impersonation Attack

In the case of user impersonation attack, the attacker can impersonate the user U_i to communicates with server S_j. In Ying's scheme [5], when $\mathcal{A}$ successfully guesses the user's identity and password, he possesses ID_{ui} and PW_{ui}, and then he can impersonate U_i at any time in the following manner:

Step 1: $\mathcal{A}$ select a nonce c_3' and computes $DID_{ui}' = H_1(ID_{ui} || c_3')$.

Step 2: $\mathcal{A}$ selects $c_4' \in Z_q^*$ and then computes $F_{ui}' = c_4'.P$, $R_{ui}' = H_2(DID_{ui}' || F_{ui}')$, $\sigma_{ui}' = M_{ui} + R_{ui}'.c_4'\ mod\ q$ and $A_{ui}' = N_{ui} \oplus H_1(ID_{ui})$.

Step 3: $\mathcal{A}$ sends login request $(\sigma_{ui}', OID_{ui}', A_{ui}', F_{ui}')$ to S_j. It

is obvious that the login request can be authenticated successful by S_j. Consequently, $\mathcal{A}$ can impersonate a legal user to login one server registered in RC.

E. Server Impersonation Attack

In this attack, $\mathcal{A}$ needs to convince the user U_i that the information is coming from a legal server S_j. $\mathcal{A}$ can get information $Msg1$={ σ_{ui} , DID_{ui} , A^*_{ui} , F_{ui} } and $Msg2$={ σ_{sj} , ID_{sj} , B_{sj}, F_{sj} } between U_i and S_j and impersonation server as follows:

Step 1: $\mathcal{A}$ intercepts the login request of U_i $Msg1$ and the response message of S_j $Msg2$ get the identity DID_{ui}, ID_{sj} and B_{sj}.

Step 2: $\mathcal{A}$ extract c_5 by computing $F_{sj}.P^{-1} = c_5$, and then computes $R'_{sj} = H_2\left(DID_{ui} \left|\left| ID_{sj} \right|\right| B_{sj}\right)$, $M_{s1} = \sigma_{sj} - R'_{sj}.c_5 = D_{sj} \bmod q$.

Step 3: $\mathcal{A}$ disguises information $\sigma'_{sj} = M_{s1} + R'_{sj}.c_5$, and sends the forged message $Msg2' = \{\sigma'_{sj}, ID_{sj}, B_{sj}, F_{sj}\}$ to the remote user U_i.

Step 4: After receiving the message $Msg2'$, user U_i computes $Check = H_2(ID_{sj}||B_{sj})Pub_{RC} + B_{sj} + R_{sj}.F_{sj}$ and verifies $\sigma'_{sj}.p = M_{s1}.p + R'_{sj}.c_5.p =? Check$, where $M_{s1}.p = H_2(ID_{sj}\left|\right|B_{sj}).Pub_{RC} + B_{sj}$, $R'_{sj}.c_5.p = R'_{sj}.F_{sj} = R_{sj}.F_{sj}$. It is obvious that they are equal, so U_i. accepts the mutual message, and $\mathcal{A}$ can impersonate the server S_j to communicates with U_i.

In this way, the user U_i believes that the response message $Msg2'$ is from the legal sever S_j.

VI. Proposed Scheme

In this section, we describe the details of our novel remote user authentication and key agreement scheme for multi-server 5G networks using biometrics of user and *PUFs* of device. The important notations used to describe the proposed scheme are listed in Table I. In many biometric authentication systems the risk of biometric data leakage is extremely high as biometric data is directly stored in user devices or smart cards. In our scheme, the biometric data of users will not be stored on any devices, which can completely eliminate the risk of biometric data leakage. The proposed scheme integrating device PUF and user biometrics to realize two-way authentication between device and user. The biometric key will be computed by using fuzzy extractor. And then, using it as a challenge, the corresponding response of PUF, i.e. the 'biometric' of the device can be obtained. Our authentication approach involves three entities, namely: user (U_i), server (S_j) and registration center (RC), where RC performs the responsibilities of registration. The proposed scheme requires three stages: setup phase, registration phase, login, mutual authentication and key agreement phases. It supports remote user authentication without password and session key agreement between the users and the servers.

A. Setup Phase

In this phase, the RC selects a non-singular elliptic curve E_p over a finite field F_p, a generating element $P \in G$, where p is a large prime and G is an additive group of order q consisting of points on E_p. The RC selects its private key S_{RC} and computes the system public key $Pub_{RC} = S_{RC} \cdot P$. A secure one-way hash function $h(\cdot)$ is generated. The parameters P, p, q, Pub_{RC}, $h(\cdot)$ are made public by the RC.

B. Registration Phase

The registration of users and servers is conducted in the secure channel.

(1) Users registration phase

Fig 4 shows users registration phase of proposed scheme. The following steps are performed during this phase.

Step UR1: U_i selects an identity ID_{ui} and obtains its biometrics BIO_{ui} by using the biometrics extractor.

Step UR2: U_i obtains the user secret key K_{ui} and the auxiliary data hd_{ui} from biometrics using the function $FE.Gen(\cdot)$ i.e., $(K_{ui}, hd_{ui}) = FE.Gen(BIO_{ui})$. After that, the U_i computes the PUF outputs $R_{ui} = PUF(K_{ui})$ to obtain the device secret key R_{ui}. And then the U_i computes $PW_{ui} = h(K_{ui}||R_{ui})$, $AID_{ui} = h(ID_{ui}||R_{ui})$ and $hd_{ui}^* = hd_{ui} \oplus h(ID_{ui})$. Finally, U_i sends $\{PW_{ui}, AID_{ui}\}$ along with a request for registration via a secure channel to the RC. It is noteworthy that K_{ui} is not disclosed to any others in subsequent communications because it can only be generated by biometrics of user. Further, the uniqueness of the user's biometrics is perfectly combined with the physical uniqueness of the device PUF.

Step UR3: After receiving the registration information sent by the user U_i, the RC generates a random number b_i and computes $V_i = b_i \cdot P$, $Z_i = h(h(AID_{ui}||Pw_{ui})||V_i) \cdot S_{RC} \bmod q + b_i$, $X_i = h(Z_i||V_i) \oplus h(AID_{ui}||PW_{ui})$, $T_{ui} = h(h(Z_i||V_i)||PW_{ui})$, $DV_i = V_i \oplus h(PW_{ui})$ and $DZ_i = Z_i \oplus h(PW_{ui}||V_i)$. Finally, RC sends $\{X_i, T_{ui}, DV_i, DZ_i\}$ to U_i.

Step UR4: U_i stores $\{X_i, T_{ui}, DV_i, DZ_i, hd_{ui}^*\}$ into smart device.

Obviously, our scheme needs both the legitimate user and the legitimate smart device to know the user secret key K_u and the device secret key R_u, otherwise the attacker will be helpless. At the same time, we don't need to worry about password guessing attacks in our scheme because we don't need passwords in our scheme.

(2) Servers registration phase

Fig 5 shows servers registration phase of proposed scheme. The following steps are performed during this phase.

Step SR1: S_j selects an identity ID_{sj} and submits $\{ID_{sj}\}$ via a secure channel to RC for registration.

Step SR2: RC generates a random number a_j and computes $Pub_{sj} = P \cdot a_j$ and $K_{sj} = a_j^{-1} \cdot S_{RC} - h(ID_{sj})$. Then, RC sends $\{Pub_{sj}, K_{sj}\}$ to S_j.

Step SR2: S_j stores K_{sj} and publishes $\{ID_{sj}, Pub_{sj}\}$.

C. Login, Authentication and Key Agreement Phases

The purpose of this phase is the mutual authentication between U_i and S_j, and also establish a secret session key so that they can use this session key for their future secure communication. The login and mutual authentication phases are

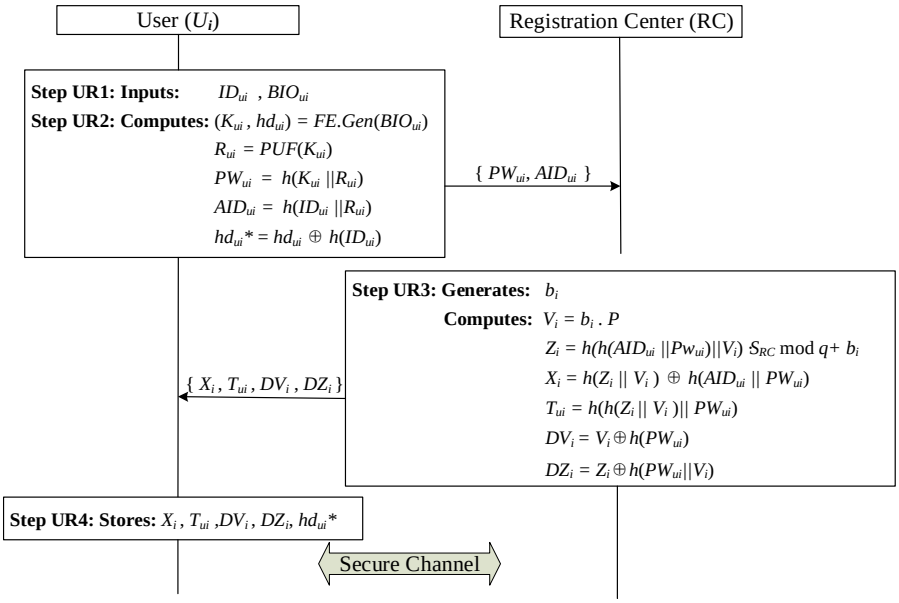

Fig. 4 Users registration phase.

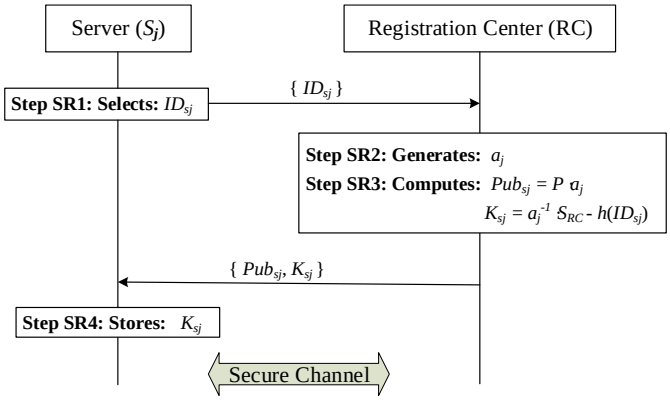

Fig. 5. Servers registration phase

briefly described in Fig 6. Specific steps are shown as follows:

Step L1: Firstly, U_i inputs its identity ID_{ui} and BIO_{ui} and computes $hd_{ui}{}' = hd_{ui}* \oplus h(ID_{ui})$.

Step L2: U_i extracts $K_{ui}{}' = FE.Rec(BIO_{ui}, hd_{ui}{}')$ and computes $R_{ui}{}' = PUF(K_{ui}{}')$, $PW_{ui}{}' = h(K_{ui}{}' || R_{ui}{}')$, $AID_{ui}{}' = h(ID_{ui}{}' || R_{ui}{}')$, $h(Z_i || V_i) = X_i \oplus h(AID_{ui}{}' || PW_{ui}{}')$.

Step L3: The smart device has to validate U_i's authentication. The smart device computes $h(h(Z_i || V_i) || PW_{ui}{}')$ and compares it with the stored T_{ui}. If they are not equal, the session is terminated.

Step L4: Smart device generates n_1 and n_2. Then, smart device computes $V_i = DV_i \oplus h(PW_{ui}{}')$, $K_1 = n_1 \cdot (Pub_{RC} - h(ID_{sj}) \cdot Pub_{sj})$, $M_{ij} = n_1 \cdot Pub_{sj}$, $EK_i = n_2 \cdot P$, $Z_i = DZ_i \oplus h(PW_{ui}{}' || V_i)$, $A = Z_i + n_2$, $V_i{}' = h(K_1) \oplus V_i$, $AID_{ui}{}^* = h(K_1 || Z_i) \oplus h(AID_{ui}{}' || PW_{ui}{}')$. Finally, U_i sends $\{M_{ij}, EK_i, A, V_i{}', AID_{ui}{}^*\}$ to S_j.

Step A1: S_j receives the message $\{M_{ij}, EK_i , A, V_i{}', AID_{ui}{}^*\}$ and completes user verification. S_j obtains $K_1{}'$ by computing $K_1{}' = K_{sj} \cdot M_{ij}$ and decrypts messages $V_i = V_i{}' \oplus h(K_1{}')$ and computes $h(AID_{ui}{}' || PW_{ui}{}') = AID_{ui}{}^* \oplus h(K_1{}' || V_i)$.

Step A2: S_j computes $h(h(AID_{ui}{}' || PW_{ui}{}') || V_i) \cdot Pub_{RC} + EK_i + V_i$, and compares it with $A \cdot p$. If they are not equal, the session is terminated.

Step A3: Subsequently, S_j generates n_3 and computes $EK_j = n_3 \cdot P$, $EK_j{}' = EK_j \oplus V_i$, $EK_{ij} = n_3 \cdot EK_i$, $TS_u = h(h(AID_{ui}{}' || PW_{ui}{}') || EK_j || K_1{}')$. S_j sends $\{EK_j{}', TS_u\}$ to user U_i.

Step A4: After receiving the message sent by the S_j, U_i computes $EK_j = EK_j{}' \oplus V_i$, $TS_u{}' = h(h(AID_{ui}{}' || PW_{ui}{}') || EK_j || K_1)$.

Step A5: S_j compares TS_u with $TS_u{}'$, and when they are equal, U_i believes that S_j is credible. Finally, U_i computes $EK_{ij}{}' = EK_j \cdot n_2$, $SK_{ij} = h(h(AID_{ui}{}' || PW_{ui}{}') || ID_{sj} || EK_{ij}{}' || K_1)$, $M = h(SK_{ij} || EK_{ij}{}')$ and submits M to S_j via common channel.

Step A6: After receiving the message sent by the U_i, S_j computes $SK_{ij} = h(h(AID_{ui}{}' || PW_{ui}{}') || ID_{sj} || EK_{ij} || K_1{}')$. S_j calculates $h(SK_{ij} || EK_{ij})$ and compares it with M, and when they are equal, S_j believes that U_i is legal. U_i and S_j successfully established session key SK_{ij}. U_i and S_j use the established session key SK_{ij} for future communication.

VII. Security Analysis

In this section, we will conduct a security analysis of the proposed scheme. Firstly, we are using the Real-Or-Random model proposed by Abdalla et al. [35] which proves that the session key security is preserved by the proposed protocol. Secondly, we'll employ the widely accepted BAN logic [36] to

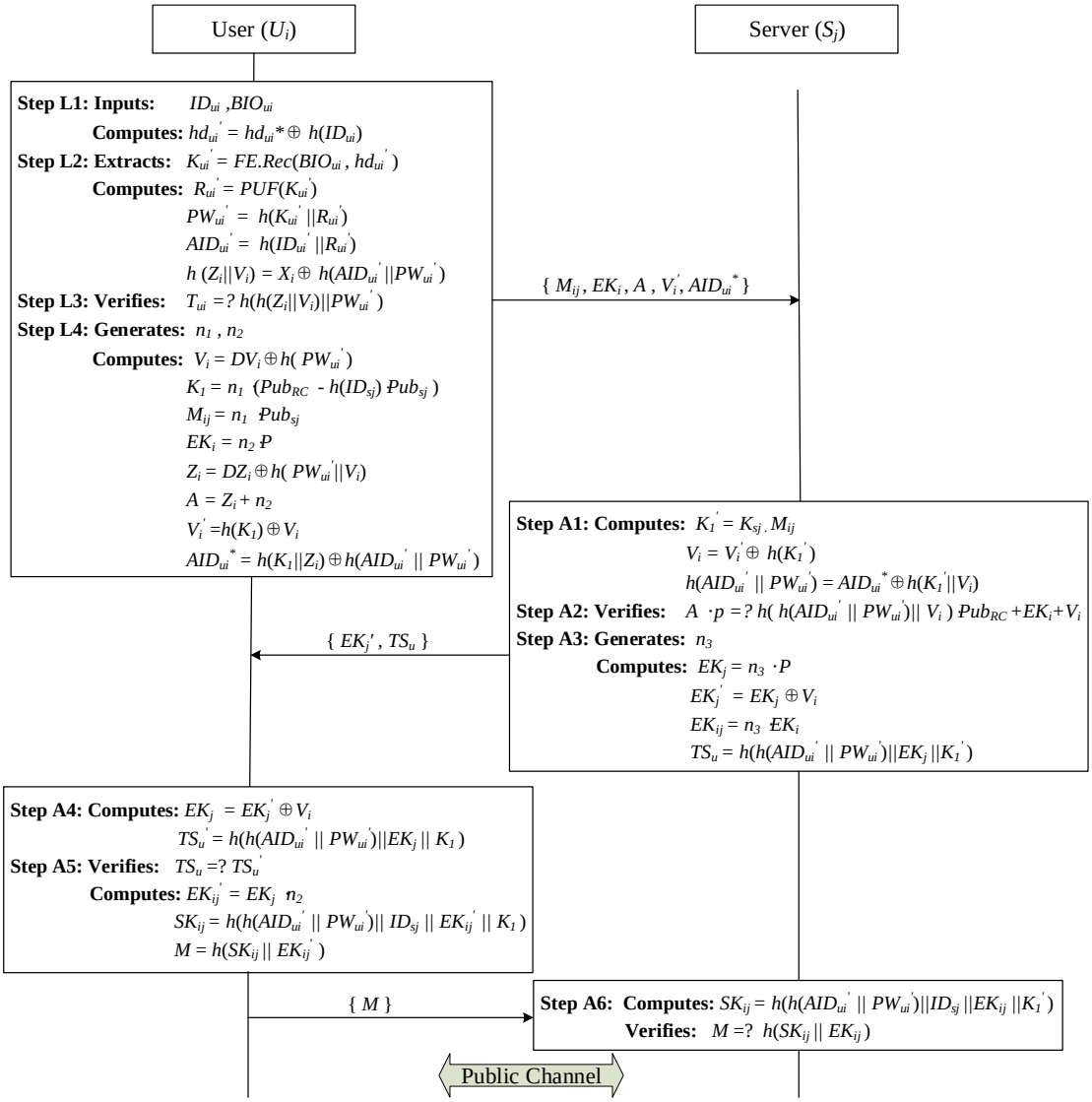

Fig. 6. Login, authentication and key agreement phases

demonstrate the security of the proposed scheme. Furthermore, we will use informal security analysis to demonstrate that the proposed scheme can perfectly achieve the security and privacy of biometrics.

A. Formal Security Analysis using Real-Or-Random (ROR) Model

Recently, formal security analysis based on the ROR model has been popularized, and the analysis method is applied to various authentication key exchange protocols. In the case of passive/active attacks, the ROR model can still provide session key *SK* security.

(1) ROR model

In our proposed solution, there are three participants, remote user U_i, server S_j and registration center *RC*. Each instance of the authentication process is taken as an oracle.

Participants: Let $\Pi_{U_i}^u$ be the uth instance of U_i, $\Pi_{S_j}^s$ the sth instance of S_j, and Π_{RC}^r are the instances r of the registration center *RC*.

Accepted State: If an instance Π_z switches to the receiving state after receiving the last authorization protocol message, then the instance enters the accepted state.

Partnering: We denote that the two instances are partnered if they satisfy the following three conditions: The instances $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$ are both in the accepted state. $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$ have the same session id *sid* and they will authenticate with each other. $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$ are reciprocal partners. And SK_{ij} is token to be the partial transcript of authentication process between $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$.

Freshness: If the session key SK_{ij} established between $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$ is not leaked via the reveal oracle *Reveal*( ) query defined below, $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$ are considered fresh.

Adversary: We believe that adversary $\mathcal{A}$ completely controls all communication in the network. $\mathcal{A}$ can query, intercept, modify, delete, forge, and even inject some or all of the exchange information between $\Pi_{U_i}^u$ and $\Pi_{S_j}^s$. $\mathcal{A}$ can access the following queries:

Execute(Π^t, Π^u): This query is executed by $\mathcal{A}$ to inquire about any two instances exchanging messages. It is modeled as an eavesdropping attack.

Reveal(Π^t): Using this query, $\mathcal{A}$ can know the session key SK_{ij} which is generated by Π^t and its partner in the current session.

Send(Π^t,*m*): This query implements an active attack wherein $\mathcal{A}$ can send a message *m* to a participate instance Π^t, and in reply, it receives a response from Π^t.

CorruptSmartDevice($\Pi^u_{U_i}$): It corresponds to a user smart device loss/stolen attack wherein $\mathcal{A}$ can get all the sensitive secret information stored in the user smart device.

Test(Π^s): Based on the indistinguishability of the model, the semantic security model of SK_{ij} is established between $\Pi^u_{U_i}$ and $\Pi^s_{S_j}$. In this query, an unbiased coin *c* is flipped in the beginning of the game, and its output is used as a decider. The outcome is kept secret to $\mathcal{A}$ to check the output from the *Test*( ) query. Let $\mathcal{A}$ execute this query. If the session key SK_{ij} shared between $\Pi^u_{U_i}$ and $\Pi^s_{S_j}$ is fresh, Π^s returns SK_{ij} when $c = 1$ or a random number when $c = 0$. Otherwise, it returns null.

Semantic security of the session key: In the ROR model, attacker $\mathcal{A}$ was tested in the experiment to distinguish between the real session key SK_{ij} and the instance's random key. Therefore, $\mathcal{A}$ is allowed to query a large number of *Test*( ) operations to the instance of $\Pi^u_{U_i}$ or $\Pi^s_{S_j}$. The output of the *Test*( ) query should match the random bit *c*. Finally, $\mathcal{A}$ returns a guess bit c', if $c = c'$, then $\mathcal{A}$ successfully obtains the correct information in the experiment. Let *E* denote the event in which the $\mathcal{A}$ wins the game. Adv_P^{AKE}, *P* and *Pr*[*X*] respectively represent the advantage of the attacker to break the semantic security of the authentication key agreement (AKE) protocol proposed by us, our protocol and the probability of event *X*. The advantage achieved by $\mathcal{A}$ in the semantic security of AKE is represented $Adv_P^{AKE} = |2 \cdot Pr[E] - 1|$. We say *P* is a secure multi-server scheme in the ROR sense when $Adv_P^{AKE} \le \varepsilon$, for ε is negligible.

Random oracle: Both attacker $\mathcal{A}$ and each participant are provided with a one-way hash function $h(\cdot)$, which is modeled as a random oracle, say *Hash* [37]. The *Hash* oracle is simulated by a two-tuple (*a, b*) table of binary strings. In case, if a hash query *h(a)* is made, the *Hash* oracle returns *b* when *a* is present in the table; else, it returns a uniform random string *b* and the pair (*a, b*) is kept safe in the corresponding table [38].

(2) Security proof

Under the ROR model, the formal proof of the session key security of the system is as follows:

Definition: If $Adv_A^{Hash}(t)$ denotes the advantage of an adversary $\mathcal{A}$ in finding a hash collision in polynomial time *t*, then

$$Adv_{\mathcal{A}}^{Hash}(t)= Pr[(ins_1,ins_2 \leftarrow \mathcal{A} : ins_1 \neq ins_2 \text{ and } h(ins_1)=h(ins_2)] \le \varepsilon_0 \quad (4)$$

Where (ins_1,ins_2) indicates that the inputs ins_1 and ins_2 are randomly selected by $\mathcal{A}$. An (ε_0, t)- an adversary cannot break the collision resistance of $h( )$ with advantage $\le \varepsilon_0$ in run time *t*, where $\varepsilon_0>0$ and is negligible.

Theorem: Let's assume that $\mathcal{A}$ be an adversary running in polynomial time *t* against the proposed scheme *P* in the ROR model, *l* denotes the number of bits present in the biometric key. And then, the advantage of $\mathcal{A}$ in breaking the session key security of *P* can be estimated as:

$$Adv_P^{AKE}(t) \le \frac{q_h^2}{|\text{Hash}|} + \frac{q_p^2}{|PUF|} + 2 \cdot \max(C' \cdot q_s^{s'}, \frac{q^s}{2^l}) \quad (5)$$

Where q_h, q_p, q_s, $|Hash|$ and $|PUF|$ denote the number of *Hash* queries, the number of *PUF* queries, the number of *Send* queries, the range space of $h( )$, the range space of PUP function, respectively, and *C'* and *S'* are the Zipf's parameters [39].

Proof: Proof of the formal security key is as follows, very similar to what has appeared in the literature [13]. We need the next game stages Gm_j(*j*=0,1,2,3, 4). We use $Succ_j^{\mathcal{A}}$ to indication that the attacker can win Gm_j.

***Game Gm₀*:** In the initial game Gm_0, the bit *c* is chosen by a polynomial-time *t* adversary $\mathcal{A}$. According to the semantic security of session key, it can be obtained as follows:

$$Adv_P^{AKE}(t)= |2 \cdot Pr[Succ_0^{\mathcal{A}}]-1| \quad (6)$$

***Game Gm₁*:** $\mathcal{A}$ invokes the *Execute*( ) query in the game to implement the eavesdropping attacks. Then, $\mathcal{A}$ calls the *Test*( ) query at the end of Gm_1. The output of the *Test*( ) operation is used as a deciding factor for distinguishing the actual session key SK_{ij} between U_i and S_j. Next, the session key formation is as follows: U_i computes the session key $SK_{ij}=h(h(AID_{ui}^{'}||PW_{ui}^{'})||ID_{sj}||EK_{ij}^{'}||K_1)$ shared with S_j, and the same session key computed by S_j, is shared with U_i as $SK_{ij}=h(h(AID_{ui}^{'}||PW_{ui}^{'})||ID_{sj}||EK_{ij}||K_1^{'})$. Suppose $\mathcal{A}$ intercepts message $Msg1=\{M_{ij}, EK_i, A, V_i^{'}, AID_{ui}^{*}\}$, $Msg2=\{EKj', Tsu\}$ and $Msg3=\{M\}$.The session key computation by $\mathcal{A}$ needs the long-term secrets EK_{ij}, User's master key K_{ui} and the *RC*'s private key S_{RC}. $\mathcal{A}$ also need the short-term secrets n_1, n_2 and n_3. In addition, it is difficult for an attacker to know the user U_i's pseudo-identity identifier AID_{ui}. Without these secret credentials, the chance of winning game Gm_1 by intercepting messages *Msg*1, *Msg*2 and *Msg*3 is not increased and it is equivalent to the game G_0. It then follows that:

$$Pr[Succ_1^{\mathcal{A}}] = Pr[Succ_0^{\mathcal{A}}] \quad (7)$$

***Game Gm₂*:** *Send*( ) operations and *Hash*( ) queries are used in this partial game. The simulation of this part of the game is similar to the active attack, by intercepting *Msg1*, *Msg2* and *Msg3*, then $\mathcal{A}$ tries to crack the session key between U_i and S_j. *Msg1*, *Msg2* and *Msg3* relate to random numbers n_1, n_2 and n_3. Hence, there is no collision in hash outputs when $\mathcal{A}$ makes *Hash* queries on these intercepted messages. Therefore, due to the collision resistance of the one-way cryptographic hash function $h( )$, the calculation of AID_{ui}, *RC*'s master key S_{RC}, the response R_{ui} of user device PUF, biometrics key K_{ui}, and short-term keys n_1, n_2 and n_3 is computationally infeasible. Since game Gm_2 is identical to game Gm_1 when the simulation of *Send*( ) and *Hash*( ) queries is not involved, according to the results of birthday paradox, the following conclusions are obtained:

$$\left|Pr[Succ_1^{\mathcal{A}}] - Pr[Succ_2^{\mathcal{A}}]\right| \le \frac{q_h^2}{2|Hash|} \quad (8)$$

***Game Gm₃*:** This round of the game differs from Gm_2 only in replacing the hash queries with PUF queries. Therefore, the conclusion for this round is similar to that of Game2:

$$\left|Pr[Succ_3^{\mathcal{A}}] - Pr[Succ_2^{\mathcal{A}}]\right| \le \frac{q_p^2}{2|PUF|} \quad (9)$$

***Game Gm4*:** In the game *Gm4*, the *CorruptSmartDevice* ( ) operation is used. In this game, $\mathcal{A}$ has the secret credentials $\{X_i, T_{ui}, DV_i, DZ_i, hd_{ui}^*\}$ from U_i's smart device, where $hd_{ui}^*=hd_{ui} \oplus h(ID_{ui})$, $X_i=h(Z_i||V_i) \oplus h(AID_{ui}||PW_{ui})$. Without the secret credentials ID_{ui} and biometric secret key K_{ui}, it is computationally infeasible to derive the AID_{ui} and PW_{ui}. Assuming K_{ui} is l bits, the guessing probability of $K_{ui} \in \{0,1\}^l$ by $\mathcal{A}$ is approximately $1/2^l$. Note that games *Gm3* and *Gm4* are identical when biometrics guessing attacks are not involved. If the system has a limit on the number of input errors, we can know:

$$\left|Pr[Succ_3^{\mathcal{A}}] - Pr[Succ_4^{\mathcal{A}}]\right| \le \max(C' \cdot q_s^{s'}, \frac{q^s}{2^l}) \quad (10)$$

Since all games are executed, attacker $\mathcal{A}$ can only guess the correct bit c. Then come to the following conclusion:

$$Pr[Succ_4^{\mathcal{A}}] = \frac{1}{2} \quad (11)$$

According to formula (6) and formula (7), we can get the following conclusions:

$$\frac{1}{2} Adv_P^{AKE}(t) = \left|Pr[Succ_0^{\mathcal{A}}] - \frac{1}{2}\right| = \left|Pr[Succ_1^{\mathcal{A}}] - \frac{1}{2}\right| \quad (12)$$

According to the triangle inequality |a-b|+|b-c|≥|a-c|, and (8), (9), (10), (11) and (12) get the following corollary:

$$\begin{aligned} \frac{1}{2} Adv_P^{AKE}(t) &= \left|Pr[Succ_1^{\mathcal{A}}] - \frac{1}{2}\right| = \left|Pr[Succ_1^{\mathcal{A}}] - Pr[Succ_4^{\mathcal{A}}]\right| \\ &\le \left|Pr[Succ_1^{\mathcal{A}}] - Pr[Succ_3^{\mathcal{A}}]\right| \\ &\quad + \left|Pr[Succ_3^{\mathcal{A}}] - Pr[Succ_4^{\mathcal{A}}]\right| \\ &\le \left|Pr[Succ_1^{\mathcal{A}}] - Pr[Succ_2^{\mathcal{A}}]\right| \\ &\quad + \left|Pr[Succ_2^{\mathcal{A}}] - Pr[Succ_3^{\mathcal{A}}]\right| \\ &\quad + \left|Pr[Succ_3^{\mathcal{A}}] - Pr[Succ_4^{\mathcal{A}}]\right| \\ &\le \frac{q_h^2}{2|\text{Hash}|} + \frac{q_p^2}{2|PUF|} + \max(C' \cdot q_s^{s'}, \frac{q^s}{2^l}) \end{aligned} \quad (13)$$

Finally, multiply both sides of equation (13) by 2 and simplify to get the following result, which proves that our scheme can achieve the session key security.

$$Adv_P^{AKE}(t) \le \frac{q_h^2}{|\text{Hash}|} + \frac{q_p^2}{|PUF|} + 2 \cdot \max(C' \cdot q_s^{s'}, \frac{q^s}{2^l}) \quad (14)$$

B. Formal Security Analysis using Burrows-Abadi-Needham (BAN) Logic

Table II has summarized the various basic notations in our logic analysis used by the BAN logic. Detailed proof is as follows.

TABLE II
BASIC NOTATIONS OF THE BAN LOGIC

Basic notation	Meaning
$P \models X$	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
$P \Rightarrow X$	Principal P has jurisdiction over the statement X
$\#(X)$	Formula X is fresh
$P \xleftrightarrow{K} Q$	P and Q use the shared key K to communicate.
$\overset{K}{\mapsto} P$	K is a public key of P
$P \overset{X}{\rightleftharpoons} Q$	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 basic logical inference rules involved in BAN logic are listed below, which will help us to analyze our own schemes with theoretical basis later.

Message-meaning rule (R1): $\dfrac{P \models Q \xleftrightarrow{K} P,\ P \lhd \{X\}_K}{P \models Q \mid\sim X}$,

$\dfrac{P \models P \overset{K}{\rightleftharpoons} Q,\ P \lhd \{X\}_K}{P \models Q \mid\sim X}$

Nonce-verification rule(R2): $\dfrac{P \models \#(X), P \models Q \mid\sim (X)}{P \models Q \models X}$

Jurisdiction rule(R3): $\dfrac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$

Fresh rule(R4): $\dfrac{P \models \#(X)}{P \models \#(X,Y)}$

Belief rule(R5): $\dfrac{P \models Q \models (X,Y)}{P \models Q \models (X)}$, $\dfrac{P \models (X), P \models (Y)}{P \models (X,Y)}$

For proposed scheme, we use BAN logic language to make relevant assumptions, i.e. A1-A8, and preset the goals realized between U_i and S_j, i.e. G1-G4.

A1: $U_i \models \#(n_1, n_2, n_3)$ A2: $S_j \models \#(n_1, n_2, n_3)$

A3: $U_i \models U_i \xleftrightarrow{Pub_{sj}} S_j$ A4: $S_j \models U_i \xleftrightarrow{Pub_{sj}} S_j$

A5: $U_i \models U_i \overset{EK_{ij}}{\rightleftharpoons} S_j$ A6: $S_j \models U_i \overset{EK_{ij}}{\rightleftharpoons} S_j$

A7: $U_i \models S_j \Rightarrow U_i \xleftrightarrow{SK_{ij}} S_j$ A8: $S_j \models U_i \Rightarrow U_i \xleftrightarrow{SK_{ij}} S_j$

G1. $U_i \models S_j \models U_i \xleftrightarrow{SK_{ij}} S_j$ **G2.** $S_j \models U_i \models U_i \xleftrightarrow{SK_{ij}} S_j$

G3. $U_i \models U_i \xleftrightarrow{SK} S_j$ **G4.** $S_j \models U_i \xleftrightarrow{SK_{ij}} S_j$

Mutual authentication between U_i and S_j is realized by using the above assumptions and rules to prove the proposed protocol. The main steps are as follows:

Msg1: $U_i \rightarrow S_j$: $<M_{ij}, EK_i, A, V_i', AID_{ui}^*, >$

Msg2: $S_j \rightarrow U_i$: $<EK_j', TS_u>$

Msg3: $U_i \rightarrow S_j$: $<M>$

Two pieces of information emerge from this combination:

Msg1: $U_i \rightarrow S_j$: $<M_{ij}, EK_i, A, V_i', AID_{ui}^*, M>$

Msg2: $S_j \rightarrow U_i$: $<EK_j', TS_u>$

The basic symbols of BAN logic are used to formalize Msg1 and Msg2 as:

M1: $S_j \lhd \langle M_{ij}, EK_i, A, V_i', AID_{ui}^*, M \rangle$

That is: $S_j \lhd \{K_1, V_i', AID_{ui}, n_1, n_2, n_3\}_{Pub_{sj}}, \{SK_{ij}\}_{EK_{ij}}$

M2: $U_i \lhd < EK_j', TS_u >$

That is: $U_i \lhd \{< EK_j, K_1, n_1, n_2, n_3 >_{h(AID_{ui}, PW_{ui})}, V_i\}_{Pub_{sj}}$

Based on the previous work, the specific logical analysis process is as follows:

According to information M1, Message-meaning rule R1 and hypothesis A4, A6, we can get:

S1. $S_j \models U_i \mid\sim U_i \xleftrightarrow{SK_{ij}} S_j$

According to information M1 and hypothesis A2 and inference rule R4, we can get:

S2. $S_j \models \#(U_i \xleftrightarrow{SK_{ij}} S_j)$

According to S2 and inference rule R2, we can get:

S3. $S_j \models U_i \models U_i \xleftrightarrow{SK_{ij}} S_j$ **(G2)**

Based on S3 and jurisdiction rule R3 and hypothesis A8, we

get:

S4. $S_j \mid\equiv U_i \xleftrightarrow{SK_{ij}} S_j$ (**G4**)

According to information M2, Under the premise of assumption A3, we can use the message meaning rule R1 to obtain:

S5. $U_i \mid\equiv S_j \mid\sim (< EK_j, K_1, n_1, n_2, n_3 >_{h(AID_{ui}, PW_{ui})}, V_i)$

Next, using the assumption A1, and the freshness rule R4, we can get:

S6. $U_i \mid\equiv \#(< EK_j, K_1, n_1, n_2, n_3 >_{h(AID_{ui}, PW_{ui})}, V_i)$

Moreover, according to S5 and S6, the nonce-verification rule R2 applies and yields:

S7. $U_i \mid\equiv S_j \mid\equiv (< EK_j, K_1, n_1, n_2, n_3 >_{h(AID_{ui}, PW_{ui})}, V_i)$

According to S7, a series of assumptions A1, A3, A5 and $SK_{ij} = h(h(AID_{ui}^{'}||PW_{ui}^{'})||ID_{sj}||EK_{ij}^{'}||K_1)$, the freshness rule R5 and nonce-verification rule R2 apply and yield:

S8. $U_i \mid\equiv S_j \mid\equiv U_i \xleftrightarrow{SK_{ij}} S_j$ (**G1**)

Finally, according to S8, A7 and jurisdiction rule R3, we can infer:

S9. $U_i \mid\equiv U_i \xleftrightarrow{SK} S_j$ (**G3**)

The above goals 1-4 clearly indicate that our scheme achieves the mutual authentication between U_i and S_j. Therefore, our scheme maintains a secure interactive authentication between U_i and S_j.

C. Informal Security Analysis

Based on some common security attributes, and assuming that the public communication channel can be controlled by an attacker. Furthermore, we have respected the facts that the smart devices can be lost or stolen by a hacker, as a result, the information stored in the devices are insecure. We conducted a detailed informal analysis of the proposed scheme, which more scientifically proved that our scheme can perfectly realize the biometrics security and privacy. We claim our scheme is secure and can satisfy the following propositions.

Proposition 1. *User anonymity and untraceability protection*

Proof. In our solution, ID_{ui} is encrypted by the response of PUF equipped in smart device, which has strong user anonymity. Even the RC knows nothing about ID_{ui} in our scheme. The user real identity ID_{ui} is hidden in message $AID_{ui}=h(ID_{ui}||R_{ui})$. To obtain ID_{ui} using this message, an adversary must konw ID_{ui} and R_{ui}. To get the response of PUR R_{ui} an adversary has to extract the biometrics key K_{ui} of user. As the biometric is uniquely secret to the user only, an attacker cannot obtain the real ID_u value, even the RC and servers also cannot retrieve it. In the login phase, U_i sends $\{M_{ij}, EK_i, A, V_i^{'}, AID_{ui}^{*}\}$ to S_j, and the parameters in this message are dynamic because n_1 and n_2 are generated randomly during each session. So, the same U_i sends different messages to the same server each time. In conclusion the proposed protocol can resist user tracing attacks. Therefore, our proposed solution can provide user anonymity and untraceability, and further protect user privacy.

Proposition 2. *Withstand both online and offline password guessing attack*

Proof. The proposed scheme does not involve the use of passwords, which is not only convenient for users, but also saves the storage space of the server. It only uses the secret key K_{ui} extracted from the user's biometrics and the R_{ui} generated by the *PUF* to encrypt the user-side data. Therefore, the attacker will never have the opportunity to perform a password guessing attack.

Proposition 3. *Resist stolen smart device attack*

Proof. Assume that the attacker stole the user device and extracted the information $\{X_i, T_{ui}, DV_i, DZ_i, hd_{ui}^{*}\}$ stored in it. All the obtained values are safeguarded using the biometrics of user. Among them, $X_i=h(Z_i||V_i) \oplus h(AID_{ui}||PW_{ui})$, $T_{ui}= h(h(Z_i||V_i)||PW_{ui})$, $DV_i=V_i \oplus h(PW_{ui})$, $DZ_i=Z_i \oplus h(PW_{ui}||V_i)$ and $hd_{ui}^{*}=hd_{ui} \oplus h(ID_{ui})$. Both user identity and biometric information are not directly stored in device, so the attacker cannot obtain any valuable data from the extracted information. In order to obtain ID_{ui} from $hd_{ui}^{*}=hd_{ui} \oplus h(ID_{ui})$, the attacker has to get hd_{ui} from biometrics by executing *FE.Gen*( ). It is impossible to get hd_{ui} without obtaining user biometric. Therefore, our scheme can resist smart card attacks.

Proposition 4. *Resist known key secrecy attack*

Proof. In the proposed scheme, the session key $SK_{ij} = h(h(AID_{ui}^{'}||PW_{ui}^{'})||ID_{sj}||EK_{ij}^{'}||K_1)$ does not reveal any information. Suppose the attacker obtains a session key to exchange in the previous communication, it is still impossible for him to construct the current session key. Firstly, each key is hashed with one way hash function, no information can be drawn from the session key. Secondly, each session key involves random numbers, which guarantee unique key for each session. Finally, each session key involves the secret key K_{ui} extracted from the user's biometrics and the R_{ui} generated by the *PUF*. Therefore, our scheme can withstand the known key secrecy attack.

Proposition 5. *Resist user impersonation attack*

Proof. Suppose an attacker U_a masquerades a legitimate user U_i and obtains the information $\{X_i, T_{ui}, DV_i, DZ_i, hd_{ui}^{*}\}$ stored in the smart device of U_i. To access S_j, the attacker needs to generate and send a valid login request message $\{M_{ij}, EK_i, A, V_i^{'}, AID_{ui}^{*}\}$ to S_j. To compute these values, U_a needs to know ID_{ui}, the biometric secret key K_{ui} and the R_{ui} generated by the *PUF*. However, U_a is not able to obtain these pieces of secret information. Therefore, our scheme withstands the user impersonation attack.

Proposition 6. *Resist server impersonation attack*

Proof. Even if U_a is a legitimate user and can intercept all messages from a public channel, it still fails to masquerade as S_j in our proposed scheme. To masquerade as a legitimate server, attacker needs to compute a valid login response message $\{EK_j^{'}, TS_u\}$, where $EK_j^{'}=EK_j \oplus V_i$ and $TS_u = h(h(AID_{ui}^{'}||PW_{ui}^{'}) ||EK_j||K_1^{'})$. However, attacker cannot compute a valid TS_u because U_a has no chance of getting the biometrics key K_{ui} and the identity ID_{ui} of user. So, the attacker cannot generate a valid response message and forge any legitimate server. Therefore, our scheme withstands the server impersonation attack.

Proposition 7. *Resist man-in-the-middle attack*

Proof. Suppose the user login message $\{M_{ij}, EK_i, A, V_i^{'}, AID_{ui}^{*}\}$ has been intercepted by an attacker U_a during the login phase,

and attacker intends to perform a man-in-the-middle attack by modifying the $\{M_{ij}^*, EK_i^*, A^*, V_i'^*, AID_{ui}^*\}$ message. Note that $V_i'=h(K_1) \oplus V_i=h(K_1) \oplus b_i$ ⊕ P, $A=Z_i+n_2=h(h(AID_{ui}||Pw_{ui})||V_i) S_{RC} \bmod q+b_i$ and $AID_{ui}^*=h(K_1||Z_i) \oplus h(AID_{ui}'||PW_{ui}')$. However, it is impossible to get both b_i generated randomly for U_i by RC and the secret key S_{RC} of RC. U_a is much less likely to obtain the biometrics key K_u and user identity ID_{ui}. Furthermore, U_a cannot compute some values because of the difficulty of ECDLP and the *PUF*. As a result, U_a has no way to modify properly all the transmitted messages between U_i and S_j during either the login or authentication phases. This shows that our scheme is immune to man-in-middle attacks.

Proposition 8. *Resist replay attack*

Proof. An attacker cannot replay the message in our proposed scheme as each transmitted message includes the values of nonce n_1, n_2 and n_3. Because of the difficulty of ECDLP, it is hard for the attacker to obtain these nonce values. As a result, the attacker cannot construct a new message. Thus, the proposed scheme can withstand replay attacks.

Proposition 9. *Resist session-specific temporary information attacks*

Proof. Even if an attacker U_a gets the temporary information n_1, n_2 and n_3 it cannot compute the session key $SK_{ij}=h(h(AID_{ui}'||PW_{ui}')||ID_{sj}||EK_{ij}||K_1)$ without knowing the value ID_{ui}, R_{ui} and K_{ui} and RC's private key. But, U_a is impossible to know those pieces of information. This shows that the proposed scheme is secure from the session-specific temporary information attack.

Proposition 10. *Ensure session key freshness property*

Proof. In our scheme, each session key SK_{ij} involves nonce, and user's biometrics secret key for each session. The existence of nonce makes the session key generated every time a connection is established completely different. This property for different sessions guaranties the unique key for each session.

Proposition 11. *Provision of session key agreement*

Proof. In our scheme, after a successful authentication both user U_i and the server S_j need to independently compute the session $SK_{ij}=h(h(AID_{ui}'||PW_{ui}')||ID_{sj}||EK_{ij}||K_1)$ for subsequent communications. It was only when SK_{ij} computed by U_i and S_j are identical that they can communicate securely using SK_{ij}. Therefore, the proposed scheme can provide session key agreement.

Proposition 12. *Provide mutual authentication without support of RC*

Proof. In our proposed scheme, user U_i and the server S_j will successfully establish a session after mutual authentication without support of RC. S_j authenticates U_i by checking $h(h(AID_{ui}'||PW_{ui}')||V_i)$ ⊕ $Pub_{RC}+EK_i+V_i=?A$ p. If an attacker wants to launch an attack, the RC's private key is needed, but this is impossible. Moreover, it cannot achieve the biometrics secret key. Therefore, it is a secure authentication. On the other hand, the server S_j challenges the user U_i by sending the message TS_u to U_i. The attacker wants to compute TS_u, which requires user's information ID_{ui}, R_{ui} and K_{ui} and RC's private key. But, it is hard to get those pieces of information. Therefore, our scheme provides a secure mutual authentication without support of RC.

Proposition 13. *Resist insider attack*

Proof. In the user authentication scheme with the participation of the Registration Center (RC), the RC is considered as a trusted third party. We always try to keep the useful information of users/servers in the RC as little as possible. In our proposed scheme, RC does not store the information of users/servers. Therefore, the internal users is impossible to obtain the legitimate user's/server's registration information from the RC, and they cannot impersonation the legitimate user/server for legal authentication.

VIII. PERFORMANCE COMPARISON

To evaluate the performance of the proposed scheme, in this section, we compare the performance of the proposed scheme with the relevant schemes proposed by Ying et al. [5], Xu et al. [22], Chaturvedi et al. [26], He et al. [27], Kumari et al. [29] and Feng et al. [30] in terms of security, computation complexity and communication overhead. Because each user and server only need to register in the registry once, we only compare the login and authentication phases.

A. Security Comparisons

The list of desired security properties is shown in Table III, and the symbol '√' represents that the scheme prevents attack or satisfies the attribute, and the symbol '×' represents that the scheme does not resist attack or failed to satisfy the attribute. It can be seen from Table III that our solution is the only solution that satisfies all security attributes, while other solutions are subject to at least one or more security attacks.

B. Comparison of computational complexity

To further evaluate the performance of proposed scheme, we have carried out a detailed analysis of the computational costs in all phases between the proposed scheme and other relevant schemes.

Table IV shows the comparison of the proposed solution with existing similar schemes concerning computational efficiency with two party authentication protocol in multi–server environment. Let T_{ecc}, T_h, T_H, T_f, T_r, T_C, T_e , T_b and T_{pm} denote the time to execute an elliptic curve point multiplication operation, one-way hash function $h(\ )$, bio-hashing, a fuzzy extraction operation, random number generation, chaotic map operation, modular exponentiation, bilinear map computation, and pairing-base scalar multiplication, respectively. According to the results reported in [23], T_{ecc}≈0.6ms, T_h≈0.008ms, T_H≈0.029ms, T_f ≈0.029ms, T_r≈0.029ms, T_C≈0.23ms, T_e≈2.13ms and T_b≈6.55ms. The computation costs of various related schemes are shown in Table IV and Fig 7. It is worth noting that although our scheme has a slightly higher computational cost than some schemes, it provides strong security features, can withstand various known attacks, and balances the computational efficiency of mutual authentication and key agreement. In addition, the use of passwords is not involved in our solution and we also use *PUF* features to ensure the physical security of smart devices. Therefore, from the perspective of safety, convenience and efficiency, the solution proposed in this article is the most suitable and practical

TABLE III
COMPARISON OF THE SECURITY PROPERTIES

Security properties	[5]	[22]	[26]	[27]	[29]	[30]	[Ours]
User anonymity protection	√	√	×	√	√	√	√
untraceability protection	√	√	-	√	√	√	√
Resist password guessing attack	×	-	√	√	√	-	√
Resist stolen smart device/card attack	×	√	√	√	√	×	√
Resist known key secrecy attack	√	√	√	√	×	√	√
Resist user impersonation attack	×	√	√	√	√	√	√
Resist server impersonation attack	×	√	√	√	√	√	√
Resist man-in-the-middle attack	√	√	√	√	×	×	√
Resist replay attack	√	×	√	√	√	√	√
Resist session-specific temporary information attacks	×	×	×	×	-	×	√
Support mutual authentication	√	√	√	√	√	√	√
Ensure key freshness property	√	√	√	√	√	√	√
Provision of session key agreement	×	√	√	√	√	√	√
Provide mutual authentication without support of RC without support of RC	√	√	√	√	√	×	√
Resist insider attack	√	√	√	_	√	√	√

TABLE IV
COMPARISON OF COMPUTATIONAL COMPLEXITY

Schemes	Operations			computation cost(ms)		
	User	Server	Total	User	Server	Total
[5]	$4T_{ecc}+7T_h+2T_r+T_f$	$4T_{ecc}+3T_h+T_r$	$8T_{ecc}+10T_h+3T_r+T_f$	2.543	2.453	4.996
[22]	$3T_{ecc}+9T_h+2T_r+T_f$	$3T_{ecc}+6T_h+2T_r$	$6T_{ecc}+15T_h+4T_r+T_f$	1.959	1.906	3.865
[26]	$2T_e+5T_h+T_r+2T_H$	$2T_e+4T_h+T_r$	$4T_e+9T_h+2T_r+2T_H$	4.387	4.321	8.708
[27]	$2T_{pm}+8T_h+T_r+2T_e$	$5T_h+T_b+T_r+5T_e$	$2T_{pm}+13T_h+2T_r+T_b+7T_e$	9.993	17.269	27.262
[29]	$2T_c+5T_h+T_r+T_f$	$2T_c+3T_h+2T_r$	$4T_c+8T_h+3T_r+T_f$	0.498	0.482	0.98
[30]	$3T_{ecc}+7T_h+T_r+2T_H$	$5T_{ecc}+17T_h+T_r$	$8T_{ecc}+24T_h+2T_r+2T_H$	1.943	3.165	5.108
Ours	$4T_{ecc}+11T_h+2T_r+T_f$	$4T_{ecc}+6T_h+T_r$	$8T_{ecc}+17T_h+T_f+3T_r$	2.575	2.477	5.052

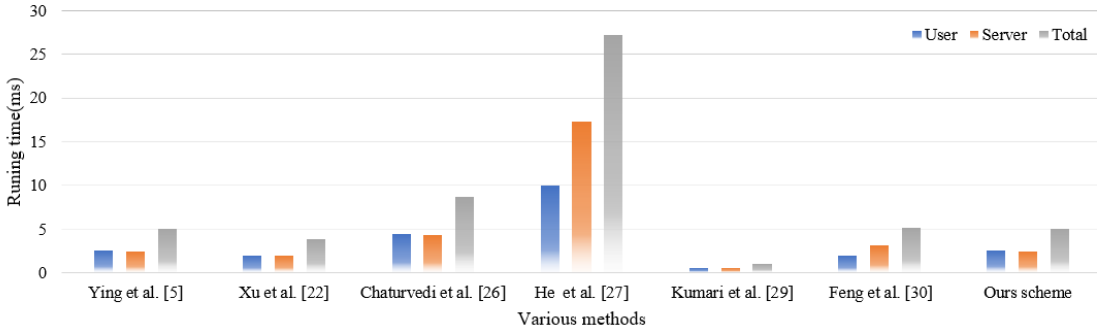

Fig. 7. Comparison of computational complexity

solution among the related biometric-based solutions.

C. Comparison of Communication Overhead

For the communication cost analysis, we assume that the hash digest (output) is 160 bits, if we use SHA-1 hash function (Secure Hash Standard, 1995), timestamp is 32 bits, user identity ID_u is 160 bits, and random nonce/number is also 160 bits. We consider ECC public 1024 bits [5, 22, 30]. Similarly, public/private keys, prime numbers in Chebyshev chaotic map cryptography [29], Finite field cryptography [26] and pairing-based cryptography [27] are assumed to be 1024 bits.

The user and the server in Ying et al.'s scheme [5] send messages $\{\sigma_{ui}, DID_{ui}, A_{ui}^*, F_{ui}\}$ and $\{\sigma_{sj}, B_{sj}, ID_{sj}, F_{sj}\}$ respectively to the other party. Thus, it needs (1024+160+1024 +1024) + (1024+160+1024+1024) = 6464 bits. In the scheme proposed in [22], the user and the application server communicate with the other party by the message $\{DID_i, Auth_i,$

A_i, T_1}, {A_j, $Auth_j$, T_2}, and {$Auth$}. Hence, the total communication overhead of this scheme is (160+160+1024 +32) + (1024+160+32) + (160) = 2752 bits. In proposed scheme in [26], the user submits M_1={UID_i, $D_i^{'}$, R_i, T_i} to server as a login request. Then the server sends M_2={ $D_j^{'}$, R_j, T_j } to user. So, this scheme bears total communication overhead of (160+1024 +160+32) + (1024+160+32) = 2592 bits. In [27], {R_{Ui}, y, a_{Sj}, C_{Ui}} are transmitted between the user and the server, where $C_{Ui} = h_6(x) \oplus (ID_{Ui} \| g_{Ui} \| a_{Ui})$. From this protocol, we can see that the total communication overhead is 1024+1024+160 + (160 +1024+160) = 3552 bits. In [29], transmitted messages between the user and the server are {D_i, M_{i1}, M_{i2}, SID_j, d_i, M_{j1}, M_{j2}, M_{i3}}, where $D_i = Ksr \oplus (u_i \| n_i \| ID_i)$. From this protocol, we can see that the total communication overhead is (160+160+160) + 1024+160+160+160+1024+160+160 = 3328 bits. The scheme of Feng et al. [30] exchanges five messages, i.e., M_1 = {RID_i, E_i, a}, M_2 = {RID_i, E_i, a, RID_j, F_j, β}, M_3 = {CID_i, γ, UID_j, δ}, M_4 = {UID_j, δ, F_j, η}, and M_5 = {ρ}. So, the total communication overhead of [30] is (160+1024+160) + (160+1024+160+160 +1024+160) + (160+160+160+160) + (160 +160+1024+160) + (160) = 6336 bits. In our protocol, the transmitted messages between the user and the server include three messages: {M_{ij}, EK_i, A, $V_i^{'}$, AID_{ui}^{*}}, {$EK_j^{'}$, TS_u}, and {M}. So, the communication overhead in our scheme is (1024+1024+160+ 160+160) + (1024+160) + (160) = 3872 bits.

The communication overhead are shown in Fig. 8. It can be seen that the scheme of Ying et al. [5] and Feng et al. [30] require the larger cost. Although the cost consumption of He et al. [27] and Kumari et al. [29] and Xu et al. [22] are slightly lower than our scheme, their scheme cannot resist some common security attacks. Therefore, the proposed scheme has a better balance between performance and security.

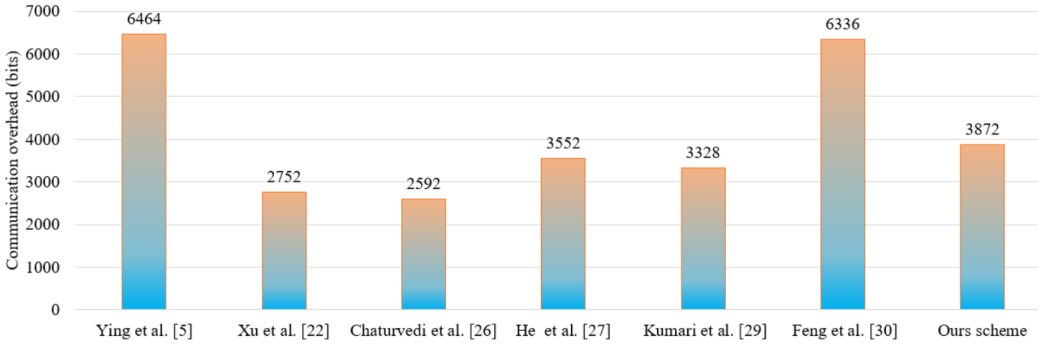

Fig. 8. Computation overhead comparison

IX. Conclusion

In this article, we showed Ying's solution is insure and explained why their solution cannot resist smart card theft and counterfeiting attacks. To overcome the security flaws in Ying's scheme, we proposed a secure user anonymity-preserving biometrics and *PUFs*-based multi-server authentication scheme with key agreement in 5G Networks. We explored the inherent security features of *PUF* and biometrics, and showed how our solutions can achieve the required security features. Without storing user biometric information, we managed to develop a solution that can completely eliminate the risk of leaking user biometric information. The fuzzy extraction algorithm and *PUF* operation were used to improve the robustness of our scheme. Performance analysis shows that the scheme can resist various known security attacks. The scheme is robust and convenient, and does not require any secret words. Compared with the state-of-the-art schemes on multi-server environment authentication, our scheme is more feasible and more promising.

References

[1] Li S, Da Xu L, Zhao S. 5G Internet of Things: A survey. Journal of Industrial Information Integration, 2018, 10: 1-9.
[2] Kondakov A, Kulik S. Intelligent Information System for Telemedicine. Procedia Computer Science, 2020, 169: 240-243.
[3] M.S.Obaidat, I.Traore, and I.Woungang, Biometric-Based Physical and Cybersecurity Systems. Cham, Switzerland: Springer, 2019.
[4] Meng Z, Pan J S, Tseng K K. PaDE: An enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowledge-Based Systems, 2019, 168: 80-99.
[5] Ying B, Nayak A. Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. Journal of Network and Computer Applications, 2019, 131: 66-74.
[6] Wu T Y, Lee Z, Obaidat M S, et al. An authenticated key exchange protocol for multi-server architecture in 5G networks. IEEE Access, 2020, 8: 28096-28108.
[7] Wong A M K, Hsu C L, Le T V, et al. Three-Factor Fast Authentication Scheme with Time Bound and User Anonymity for Multi-Server E-Health Systems in 5G-Based Wireless Sensor Networks. Sensors, 2020, 20(9): 2511.
[8] Wang D, Wang P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. Computer Networks, 2014, 73: 41-57.
[9] rshad A, Ch S A, Shafiq M, et al. An Improved Biometric Multi-Server Authentication Scheme for Chang et al.'s Protocol. Information Technology and Control, 2019, 48(2): 211-224.
[10] Bian W, Gope P, Cheng Y, et al. Bio-AKA: An efficient fingerprint based two factor user authentication and key agreement scheme. Future Generation Computer Systems, 2020, 109: 45-55.
[11] Gope P, Das A K, Kumar N, et al. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. IEEE transactions on industrial informatics, 2019, 15(9): 4957-4968.
[12] Zhang H, Bian W, Jie B, et al. A complete user authentication and key agreement scheme using cancelable biometrics and PUF in multi-server environment. IEEE Transactions on Information Forensics & Security, 2021, 16: 5413-5428.
[13] Zahednejad, B., & Gao, C. Z. A secure and efficient AKE scheme for IoT devices using PUF and cancellable biometrics. Internet of Things, 2023, 24, 100937.
[14] Zhao J, Bian W, Xu D, et al. A Secure Biometrics and PUFs-Based Authentication Scheme With Key Agreement For Multi-Server Environments. IEEE Access, 2020, 8: 45292-45303.
[15] Lamport L. Password authentication with insecure communication. Communications of the ACM, 1981, 24(11): 770-772.

[16] Hwang T, Chen Y, Laih C J. Non-interactive password authentications without password tables, in Proc. IEEE region Conf. Comput. Commun. Syst., Sep. 1990, pp. 429-431.
[17] Lee N Y, Chiu Y C. Improved remote authentication scheme with smart card. Computer Standards & Interfaces, 2005, 27(2): 177-180.
[18] Juang W S, Chen S T, Liaw H T. Robust and efficient password-authenticated key agreement using smart cards. IEEE Transactions on Industrial Electronics, 2008, 55(6): 2551-2556.
[19] Lee J.K, Ryu SR, Yoo KY. Fingerprint-based remote user authentication scheme using smart cards. Electronics Letters, 2002, 38(12): 554-555.
[20] Li L H, Lin L C, Hwang M S. A remote password authentication scheme for multiserver architecture using neural networks. IEEE Transactions on Neural Networks, 2001, 12(6): 1498-1504.
[21] Kumar A, Om H. An improved and secure multiserver authentication scheme based on biometrics and smartcard. Digital Communications and Networks, 2018, 4(1): 27-38.
[22] Xu D, Chen J, Liu Q, Provably secure anonymous three-factor authentication scheme for multi-server environments. J. Amb. Intell. Human. Comput., 2019, 10: 611-627.
[23] Haq I, Wang J, Zhou Y, et al. A survey of authenticated key agreement protocols for multi-server architecture. Journal of Information Security and Applications, 2020, 55:102639.
[24] Liao, Y.P, Hsiao, C.M. A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. Future Generat. Comput. Syst., 2013, 29: 886-900.
[25] Hsieh, W.B, Leu, J.S. An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures. J. Supercomput, 2014, 70, 133–148.
[26] Chaturvedi A, Das AK, Mishra D, Mukhopadhyay S. Design of a secure smart card-based multi-server authentication scheme. J Inform Secur Applic, 2016, 30: 64-80.
[27] He D, Zeadally S, Kumar N, Wu W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multiserver architectures. IEEE Transactions on Information Forensics & Security, 2016, 11(9): 2052-64.
[28] Chatterjee S, Roy S, Das AK, Chattopadhyay S, Kumar N, Vasilakos AV. Secure biometric-based authentication scheme using chebyshev chaotic map for multiserver environment. IEEE Trans Dependable Secure Comput., 2018, 15(5):824-39.
[29] Kumari S, Das A K, Li X, Wu F, Khan MK, Jiang Q, et al. A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. Multimed Tools Appl, 2018, 77(2): 2359-2389.
[30] Feng Q, He D, Zeadally S, Wang H. Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment. Futur Gener Comput Syst, 2018, 84: 239-251.
[31] Bansal G, Naren N, Chamola V, et al. Lightweight mutual authentication protocol for V2G using physical unclonable function. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7234-7246.
[32] Koblitz N. Elliptic curve cryptosystems. Mathematics of computation 1987, 48(177): 203-9.
[33] Kocher P, Jaffe J, Jun B. Differential power analysis. Proceeeding of Advances in Cryptology (CRYPTO'99), LNCS 1666 1999; pp: 388-397.
[34] Messergers T S, Dabbish EA, Sloan RH. Examining smart card security under the threat of power analysis attacks. IEEE Transactions on Computers, 2002, 5(5): 541-552.
[35] Abdalla M, Fouque P-A, Pointcheval D. Password-based authenticated key exchange in the three-party setting, in Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC), in Lecture Notes in Computer Science, vol. 3386. Berlin, Germany: Springer, 2005, pp. 65-84.
[36] Burrows M, Abadi M, Needham R. A logic of authentication. ACM Trans, Comput, Syst., 1990, 8(1): 18-36.
[37] Srinivas J, Das A K, Kumar N, et al. Cloud centric authentication for wearable healthcare monitoring system. IEEE Transactions on Dependable and Secure Computing, 2018, 17(5): 942-956.
[38] Odelu V, Das A K, Goswami A. A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Transactions on Information Forensics and Security, 2015, 10(9): 1953-1966.
[39] Wang D, Cheng H, Wang P, Huang X, et al. Zipf's law in passwords. IEEE Transactions on Information Forensics and Security, 2017, 12(11), 2776-2791.