Manchester Metropolitan University

e-space

Manchester Metropolitan University's Research Repository

Please cite the Published Version

Irshad, Azeem, Usman, Muhammad, Chaudhry, Shehzad Ashraf, Bashir, Ali Kashif, Jolfaei, Alireza and Srivastava, Gautam (2021) Fuzzy-in-the-Loop-Driven Low-Cost and Secure Biometric User Access to Server. IEEE Transactions on Reliability, 70 (3). pp. 1014-1025. ISSN 0018-9529

DOI: https://doi.org/10.1109/tr.2020.3021794

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Accepted Version

Downloaded from: https://e-space.mmu.ac.uk/626645/

Usage rights: © In Copyright

Additional Information: This is an Author Accepted Manuscript of a paper accepted for publication in IEEE Transactions on Reliability, published by and copyright Institute of Electrical and Electronics Engineers (IEEE).

Enquiries:
If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines)

Fuzzy-in-the-Loop Driven Low Cost and Secure Biometric User Access to Server

Azeem Irshad, Muhammad Usman, Shehzad Ashraf Chaudhry, Ali Kashif Bashir, Alireza Jolfaei, and Gautam Srivastav

***Abstract*—Fuzzy systems can aid in diminishing uncertainty and noise from biometric security applications by providing an intelligent layer to existing physical systems to make them reliable. In the absence of such fuzzy systems, a little random perturbation in captured human biometrics could disrupt the whole security system, which may even decline the authentication requests of legitimate entities during protocol execution. In the literature, few fuzzy logic-based biometric authentication schemes have been presented; however, they lack significant security features including perfect forward secrecy, untraceability, and resistance to known attacks. This study, therefore, proposes a novel two-factor biometric authentication protocol enabling efficient and secure combination of physically unclonable functions, a physical object analogous to human fingerprint, with user biometrics by employing fuzzy extractor-based procedures in the loop. This combination enables the participants in the protocol to achieve Perfect Forward Secrecy (PFS). The security of the proposed scheme is tested using the well-known real-or-random model. The performance analysis signifies the fact that the proposed scheme not only offers PFS, untraceability, and anonymity to the participants, but is also resilient to known attacks using light-weight symmetric operations, which makes it an imperative advancement in the category of intelligent and reliable security solutions.**

***Index Terms*—Fuzzy systems, biometric fuzzy extractor, mutual authentication, physical unclonable function, user access.**

I. Introduction

THE biometric verification is an integral part of human-centric systems, requiring features extraction and matching patterns using the mathematical techniques from the realm of artificial intelligence [1]. The fuzzy logic has effectively been applied in numerous biometric-pattern matching systems, including fingerprint recognition and face recognition, with the objective to reduce the noise and diminish uncertainty so that the system could behave with reliability, robustness and precision [2]. Although the biometrics are particular to an individual and remains static throughout the lifetime of a person, yet the human biometric features tend to vary a little bit over a period of time, or there might be some noise in the captured biometric template that might render the security solutions inapplicable in the absence of fuzzy systems. Due to such fuzzy system-based intelligence, the biometric recognition systems are exceedingly replacing the conventional password or token-based authentication systems. The biometric authentication mechanisms scan physical features to authenticate an individual. The fingerprint-based authentication systems have had more adoption compared to other biometrics, such as face, iris, and voice, which is mainly due to their higher accuracy rate and convenience [2], [3].

In a human-explainable client-server authentication model, a remote user needs to access the data by logging into the smart device and then ultimately putting the authentication request to the server. This is performed by sending commands to the server from the Internet for real time authentication decisions [3]–[5]. The server renders the fog computing services to users over a public communication channel. The communication between servers and users should be protected from intruder. Both participants need to mutually authenticate one another by establishing an agreed session key before communication. The identity of the user must remain anonymous, while the user itself should remain untraceable, that is, the messages of various sessions belonging to a particular user must remain indistinguishable, termed as untraceability [6]. The de-synchronization attack may be initiated by malicious intruder to loose the synchronization between legal participants. A service provider must be able to quickly thwart an attack destined to deplete its energy and other resources. Besides, the scheme must ensure Perfect Forward Secrecy (PFS) along with other security features, such as a feature that ensures the protection of session keys in case the long term secret keys pertaining to the involved participants are revealed. Hence, only the authorized users should be able to access the services of reliable servers. Many current symmetric key protocols suffer from the above-mentioned limitations, which are mostly covered by engaging Public Key Cryptography (PKC) protocols; however, PKC protocols are computationally too expensive for the low-end, resource constrained devices.

Recently, Physically Uncloneable Functions (PUFs) have been adopted for securing the communication and server-based services due to their unique intrinsic behavior [7]. The Fuzzy Extractor (FE) is employed for improving the strength of mutual authentication and key agreement by removing the noise from biometrics and output of PUF [8]. The biometrics can be stored in an encrypted form on a device using FE and it also takes care of minor changes in biometrics and possible noise while capturing the biometric input. The functionality of FE relies on a hamming distance between the stored and

A. Irshad is with Department of Computer science and software engineering, International Islamic University, Islamabad, Pakistan (e-mail: irshadazeem2@gmail.com).

M. Usman is with Faculty of Computing Engineering and Science University of South Wales, Pontypridd, CF37 1DL, UK (email: muhammad.usman@southwales.ac.uk).

S. A. Chaudhry is with Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey (email: ashraf.shahzad.ch@gmail.com).

A. K. Bashir is with Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK (email: dr.alikashif.b@ieee.org).

A. Jolfaei is with Macquarie University, Sydney, NSW 2109, Australia (email: alireza.jolfaei@mq.edu.au).

G. Srivastav is with Department of Math and Computer Science, Brandon University, Canada and Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan (email: srivastavag@brandonu.ca).

Manuscript received June 18, 2020.

captured biometric template. This distance must be adequately small for a viable authentication model [9], [10].

Many biometric authentication schemes, along the above lines, have been presented with a number of security limitations such as lacking PFS, de-synchronization issues, denial of service attacks, and other known attacks [7], [11]–[18]. To enhance the resilience of the client-server-based authentication protocols using lightweight crypto-primitives, we have designed an authenticated key agreement scheme by efficiently unifying PUFs with fingerprint biometrics of users by keeping the fuzzy extractor in the loop. That is, the device, owned by a user, requires being equipped with PUF for enabling an agreed session key between the user and the server. The scheme is designed with two-factor authentication requirements, without the need of password during the registration or the login phase. This approach is not only secure but is also convenient for users. We have used a widely accepted Real-or-Random (ROR) model to verify the security strengths of the established session key [19].

In this scheme our intent is to ensure PFS besides preserving other security features including anonymity, untraceability, and resistance to known attacks, using lightweight symmetric key operations. Considering this, we designed an efficient and secure authenticated key agreement scheme by combining FE-enabling biometrics and PUF. The main contribution of our work is as follows:

- Our main contribution lies with resolving the flaws, as indicated above, by designing a two-factor user authentication protocol through exploiting the mechanics of FE-enabled biometrics and PUF in a novel way. The users as well as the owned PUF device, both have unique physical properties that assist in establishing an agreed session key between participants.
- The designed scheme eliminates the chances of leakage of biometric information from the device. Being a two-factor authentication protocol, the user can rely only on smart card and biometrics of users for registration, logging into the device, and session key establishment during the execution of protocol without the need of password, the third factor.
- The experiments, formal analysis, and informal analysis are carried out. The results show that the designed scheme, despite being a lightweight symmetric-key protocol commits significant security features, i.e. it may thwart known attacks such as Denial of Service (DoS), de-synchronization and man-in-the-middle attacks, and also achieves untraceability and perfect forward secrecy (PFS) by submitting the PUF-based challenge-response pair towards server, a missing security feature in the existing symmetric key protocols [20].

The rest of the paper is organized as follows: Section II provides a summary of the related work. Section III explains the preliminary concepts. Section IV elucidates our proposed model. Section V illustrates the formal and informal security analysis. Section VI compares the performance efficiency of the proposed model with other schemes. The last section concludes the paper.

II. Related Work

This section illustrates the related work with respect to the evolution of single factor to biometric three-factor authentication techniques. In conventional security systems, the identity and password were used for remote logins into the system. The Lamport scheme was the first such authentication method utilizing insecure channels with a password table for storing users' verifiers [28]. Due to modification attacks in password tables, Lee et al. [21] introduced a smart cards-based scheme using the principle of ElGamal's cryptosystem, which removed the use of password tables. However, the forged identity problems [22] in smart card-based schemes lead the researchers to move towards a three-factor biometric-based ElGamal's cryptosystem [11]. However, such schemes were exposed to password modifications and masquerading attacks [12], which were later addressed by other researchers to provide sufficient mutual authentication. Later, Poh et al. introduced an improved biometric scheme with an effective session key agreement [13]. However, Poh et al.'s scheme is not able to maintain the privacy of three-factor authentication parameters. To address these shortcomings, Zhou and Ren presented a secure, mutual authentication protocol by employing costly exponentiation operations [14].

The biometric features may offer significant authentication capabilities over the above discussed schemes based on mere passwords or cryptographic keys [13], as far as the resistance from copying, guessing, losing the key type issues are concerned. Due to such properties, it becomes crucial to maintain its privacy as well as security of those biometric factors during authentication phase. However, the security of biometrics may become a concern if it becomes compromised due to the storage of corresponding biometric templates in smart cards or servers [13]. To cover such risks many schemes [14] employed keyed hash functions and fuzzy extractors for encrypting the biometrics before storing on devices. Nevertheless, the auxiliary data recovered from FE is not encrypted and is stored directly on smart card or device, which makes susceptible the privacy of biometric templates. Zhou and Ren presented a threshold predicate encryption protocol for only revealing the matched result without exposing the biometric data [14]. Wang et al. [24] also examined privacy-malicious threats related to the exposure of biometrics databases by using adversarial machine learning techniques. However, due to the storage of encrypted biometrics in repository [14], [24], the leakage concerns for the biometric data still exist with those approaches.

The Han et al. [26] and Reddy et al. [27] employed public-key based operations to ensure perfect forward secrecy, but were susceptible to de-synchronization attacks. In recent years, a few smart device and internet of things (IoT)-based security solutions have adopted PUFs to assist in mutual authentication. The peculiarity of Physically Uncloneable Functions (PUFs) has made it attractive for adoption in building the secure Industrial Internet of Things (IIoT)-based applications. PUFs are designed with random disparities in Integrated Circuits (ICs) while manufactured. Each PUF is designed with unique physical signature which can neither be cloned nor rebuilt,

TABLE I
RELATED LITERATURE

Scheme	Features	Drawbacks	Year
[21]	One of the pioneer schemes supporting fingerprint-based authentication with the help of smart card and Elgamal cryptosystem	Forged identity problems	2002
[22]	Improved three-factor biometric authentication with the Elgamal cryptosystem	Password modification and masquerading attacks	2004
[12]	A three-factor biometric authentication scheme based on discrete logarithm problem	Exposed to impersonation, replay, and temporary information attacks	2014
[23]	A three-factor biometric authentication scheme based on modular exponentiation cryptographic operations	Stolen device and Man-in-the-middle attack	2017
[24]	Examined privacy threats related to the exposure of biometrics by using adversarial machine learning techniques	The privacy leakage concerns still persist due to storage of encrypted biometrics.	2017
[25]	PUF-based two-factor authentication scheme with the combination of biometrics	De-synchronization attack, and unable to provide equivalent three-factor security	2018
[14]	A secure Threshold Predicate Encryption protocol for encrypting the biometrics	Costly exponentiation operations	2018
[13]	Privacy preserving scheme for smart home user	Unable to provide equivalent three-factor privacy to the user	2019
[26]	Client-server three-factor authentication scheme employing elliptic curve cryptography operations	De-synchronization attack, Anonymity, and server impersonation attack	2018
[27]	Improved client-server three-factor authentication scheme employing public-key cryptography	De-synchronization attack, costly crypto-primitives	2019
[7]	PUF-based Wireless Sensor Network oriented scheme	Denial of Service (DoS) threat due to direct use of PUF without removing its noise	2019
[20]	Fuzzy extractor-based biometric symmetric authentication protocol for client server environment	Man in the middle attack, DoS attack	2020

similar to human biometric features. In this connection, Bian et al. [20] presented a PUF as well as fuzzy extractor-based biometric symmetric authentication protocol for client server environment. However, the scheme is vulnerable to denial of service attack (DoS) on the server's end. The server needs to consult its repository for n number of times to verify the identity of user, leading to DoS attack, where n is total number of users. Besides, [20] suffers man-in-the-middle attack in case the malicious intruder alters the message on its way from server to user. Also Gope and Sikdar [25] employed PUF with the combination of biometrics to authenticate the user. However, on the down side, it suffered de-synchronization attack. Later, Gope et al. [7] introduced another PUF-based scheme for industrial wireless sensor networks; however the biometrics were utilized directly without removing the noise which could result in denial of service threat [20].

The above literature manifests that most of the three-factor symmetric key-based schemes do not comply with PFS, untraceability, and synchronization properties of security. To serve the purpose, many of those schemes employed symmetric key-based as well as public key-based costly crypto-primitives [26], [27]. Although, the schemes employing costly primitives fulfill PFS, but could not preserve the other security features including untraceability, or resistance to known attacks such as DoS attack, de-synchronization attack, replay attack, stolen-verifier attack etc. Therefore, an efficient security solution countering the above-mentioned flaws in the schemes is inevitable. The related work is also summarized in Table I.

III. PRELIMINARIES

This section presents preliminary details to assist readers in understanding this article.

A. Physically Uncloneable Function

A PUF uniquely maps an input to a specific output [10]. The challenge and response-based input-output pair remains unique for each PUF-based chip. A PUF-based circuit entails the following features:

- The response of a PUF circuit is strictly a function of unique manufacturing structure of the chip.
- The response of a PUF is always unpredictable.
- It is highly improbable to clone the PUF functionality.
- PUF's characteristics remain stable over time, and they are easy to be evaluated and implemented.

Since, PUF's output depends on environment variables, any alterations in the system can modify the output of the circuit. Also, it is assumed that the interactions between PUF and other devices cannot be tampered [22].

In challenge-response pair (CRP)-oriented PUF, the one-way response R for a challenge C may be identified as $R \leftarrow \mathcal{G}(C)$, where $\mathcal{G}$ represents PUF function. The PUFs are integrated chips with unique internal structure that maps one-way function from C to R. The hard prediction and easy construction renders it a good candidate to be used as security primitive for low-end devices. This is because the output of $\mathcal{G}$ function relies on physical features, and any kind of tampering attempt may change the behavior of device and hence makes it ineffective.

B. Fuzzy Extractor

Fuzzy extractor procedures could be efficiently employed to remove the noise from the output of PUF circuits, which further enhances the reliability of PUF circuits. Zhang et al. [18] analyzed the problems of dissipative screening and missing measurements regarding discrete switching of fuzzy-based systems. They proposed the solution by employing random variables with the combination of "Bernoulli binary distribution". The fuzzy extractor consists of two procedures, that is, 1) Generation function $Gen(\cdot)$ and 2) Reproduction function $Rep(\cdot)$ which are described below:

$Gen(\cdot)$: Considering a challenge-response pair, that is, R_{s_i}=$PUF(Ch_i)$, wherein the $Gen(\cdot)$ function in response to Ch_i challenge, outputs a tuple bearing secret key R_i and auxiliary data ad_i , that is, $Gen(R_{s_i})$=(R_i, ad_i).

$Rep(\cdot)$: In view of PUF-based output R_{s_i}, the $Rep(\cdot)$ extracts the same original key R_i by employing auxiliary data ad_i, with a provision that the hamming distance between existing PUF output R'_{s_i} and the original PUF output R_{s_i} is not over a pre-established error absorbing threshold ψ. Therefore, it ensures $Rep(R'_{s_i}, ad_i)$=R_i.

In this regard, an assessment for error absorbing threshold is provided by Cheon et al. [29]. That is, in case the hamming distance between existing PUF output R'_{s_i} and original PUF output R_{s_i} is η, and the number of bits in the input string is b_{in}, then ψ=η/b_{in}.

C. Malicious Intruder Model

We assess the security features of contributed protocol under Canetti-Krawczyk Model (CK) attack model [15]. The capabilities of the malicious attacker under this model are narrated below.

- A deceitful attacker $\mathcal{I}$ could intercept, erase, append, alter, hold or replay the eavesdropped messages exchanged among the legal entities.
- $\mathcal{I}$ may illegaly access the mobile gadget of user and extract all of its stored contents employing power differential analysis.

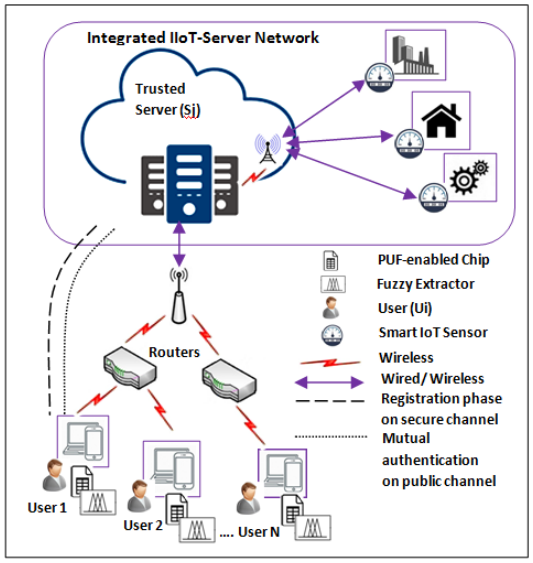

Fig. 1. System Model

- $\mathcal{I}$ could also launch recognized threats such as impersonation attack, replay attack, forgery and man-in-the-middle attacks.
- $\mathcal{I}$ may access short term ephemeral secrets from the user's storage, and long term secrets under the assumption of CK model.
- The server is treated as a reliable trusted authority for the purpose of registration.

D. System Setup

The system model represents the proposed remote mobile-user authentication protocol. The mobile gadgets employed in the system could be any kind of smart devices including mobile phones, laptops, and tablets which are normally utilized in a remote authentication system. The biometric authentication provides a way to authenticate any entity on the basis of unique physical characteristics of an individual. The contributed model is supposed to accomplish a fuzzy-extractor and PUF induced biometric two-way authenticated key agreement between server and mobile devices of users. In the current system model, the mobile gadgets are outfitted with fingerprint capturing sensor as well as PUF with strict adherence to $\mathcal{G}_i$ function, while the server serves as a trusted party. The mobile user registers and gets mutually authenticated with the same trusted party as depicted in Fig. 1. After successful authentication, the mobile users may access IoT data which is readily updated on those servers.

IV. Proposed Scheme

The PFS, anonymity, and untraceability features are mostly addressed by PKC-based primitives used in the existing protocols. Since PKC-based primitives are computationally expensive, an alternative solution could be in the use of a lightweight authenticated encryption using symmetric key operations, which are deployable by low-end devices.Although, few symmetric key-based security solutions have employed biometrics and PUF to enhance the security [8], [20], the security problems still persist in some way or other. Hence, to attain the mentioned security objectives with low cost primitives, we propose a two-factor biometric authenticated key agreement scheme between an end user and a server integrated with cloud and IoT environment. The proposed scheme aims at constructing an agreed session key without taking the password from the user, through employing PUF and fuzzy extractor functions in the loop. During the initialization setup, the i-th end user U_i and the j-th trusted authority or server S_j choose their private keys, K_u and K_s, respectively. Notations used in the scheme are shown in Table II. The proposed model covers two important phases, that is, 1) Registration phase and 2) Login and authentication phase. These phases are described below.

TABLE II
Symbols with Definitions

Symbols	Definition
U_i, S_j	Mobile User, Trusted Server
ID_i, ID_s	Identities of U_i and S_j
K_u, K_s	Private secret keys of U_i and S_j
PUF_i	Physical Uncloneable Function for U_i
B_f	Fingerprint biometric impression
$\mathcal{F}_g/\mathcal{F}_r$	Fuzzy Extractor Generation and Reproduction
N_u, N_s	Random nonces
$E_k(\cdot)/D_k(\cdot)$	Symmetric encryption / decryption
$\mathcal{I}$	Malicious Intruder
SK	Sesseion key between U_i and S_j
$\oplus$, $\|$	XOR , Concatenation
A_i, v	Pseudo-identity, Encrypted parameter to compute A_i
$h(\cdot)$	Secure one-way hash function

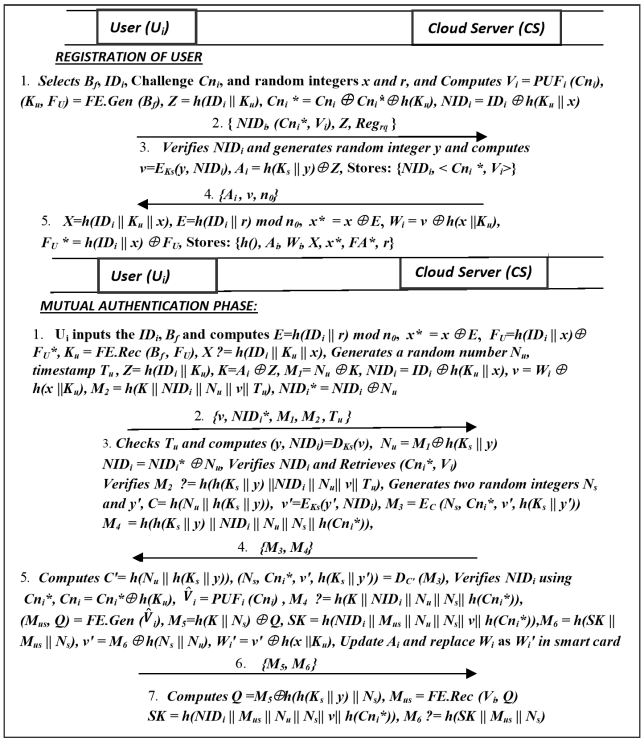

Fig. 2. Proposed Model

A. Registration phase

In this phase, the user U_i executes the registration procedure with S_j in a secure channel. The detailed steps are elaborated below:

1) Initially, the user U_i chooses an identity ID_i and scans his/her fingerprint biometrics B_f on a device. Then, the user defines two random numbers x and r, as well as a challenge Cn_i.
2) Next, the user U_i computes $V_i = \mathcal{G}_i(Cn_i)$ on PUF, and employs $\mathcal{F}_g$ on B_f to generate the private key K_u

along with auxiliary data F_U, that is, $(K_u, F_U) = \mathcal{F}_g$ (B_f). It then computes $Z = h(ID_i||K_u)$, $Cn_i^* = Cn_i \oplus h(K_u)$, and pseudo-identity $NID_i = ID_i \oplus Cn_i^* \oplus h(K_u||x)$. Then, the user U_i sends $\{NID_i, < Cn_i^*, V_i >, Z, Reg_{rq}\}$ to the server over a secure channel, bearing Reg_{rq} the registration request.

3) The server S_j verifies NID_i after receiving the request, and chooses a random number y. Then, it computes $v = E_{K_s}(y, NID_i)$, $A_i = h(K_s||y) \oplus Z$, and stores $NID_i, < Cn_i^*, V_i >$ in its database. In addition, it sends $\{A_i, v\}$ to U_i on a confidential channel.
4) The user U_i, after getting the message, computes $X = h(ID_i||K_u||x)$, $E = h(ID_i||r) \text{mod } n_0$, $x^* = x \oplus E$, $W_i = v \oplus h(x||K_u)$, $F_U^* = h(ID_i||x) \oplus F_U$, and stores $\{h(\cdot), A_i, W_i, X, x^*, F_U^*, r\}$ safely in its device.

B. Login Procedure

In the login phase, the user U_i initially inputs its identity ID_i, and imprints fingerprint B_f into the device. Next, it performs the following steps:

1) By employing the U_i's identity ID_i, the user device calculates $E = h(ID_i \| r) mod\ n_0, x^* = x \oplus E$, and extracts $F_U = h(ID_i \| x) \oplus F_U^*$. Then, it recovers the private key K_u by computing $K_u = \mathcal{F}_r$ (B_f, F_U). Next, it calculates and checks the equation $X \stackrel{?}{=} h(ID_i \| K_u \| x)$. If the match is unsuccessful, it aborts; otherwise, the user engenders a random integer N_u as well as a timestamp T_u, and calculates $Z = h(ID_i \| K_u)$ and $K = A_i \oplus Z$.
2) Next, it computes $M_1 = N_u \oplus K$, $NID_i = ID_i \oplus h(K_u||x)$, $v = W_i \oplus h(x \| K_u), M_2 = h(K \| NID_i \| N_u \| v \| T_u)$ and $NID_i^* = NID_i \oplus N_u$. In M_1, N_u is masked with the use of K_u. Next, U_i sends the authentication request $\{v, NID_i^*, M_1, M_2\}$ to server.

C. Authentication and key agreement procedure

In mutual authentication phase, S_j upon receiving $\{v, NID_i^*, M_1, M_2, T_u\}$ executes the under-mentioned steps.

1) In the beginning, S_j checks T_u, and calculates $(y, NID_i) = D_{K_s}(v)$. Then, it extracts N_u by calculating N_u=$M_1 \oplus h(K_s \| y)$, and computes $NID_i = NID_i^* \oplus N_u$. After validating NID_i, it fetches the respective challenge-response pair (Cn_i^*, V_i) from its database. Then, S_j calculates M_2 and checks $M_2 \stackrel{?}{=} h(h(K_s||y)||NID_i||N_u||v||T_u)$. If this validity does not hold true, S_j terminates the session. Otherwise, S_j computes engenders two random integers N_s and y', and calculates $C = h(N_u||h(K_s||y))$. In addition, it computes v'=$E_{K_s}(y', NID_i), M_3$=$E_C(N_s, Cn_i^*, v', h(K_s||y'))$ and $M_4 = h(h(K_s||y)||NID_i||N_u||N_s||h(Cn_i^*))$. Finally, it sends the message $\{M_3, M_4\}$ to U_i.
2) U_i upon receiving the message calculates C'=$h(N_u||h(K_s||y))$ and extracts $(N_s, Cn_i^*, v', h(K_s||y'))$ after decrypting M_3 using C' and validates NID_i against Cn_i^*. Then, it further calculates Cn_i=$Cn_i^* \oplus h(K_u), \widehat{V}_i$=$PUF_i(Cn_i)$ and validates $M_4 \stackrel{?}{=} h(K||NID_i||N_u||N_s||h(Cn_i^*))$. Next, U_i computes Mus and auxiliary data Q by using fuzzy extractor $Gen(\cdot)$, i.e. (M_{us}, Q)=$\mathcal{F}_g$ ($\widehat{V}_i$). Further, it computes M_5=$h(K||N_s) \oplus Q, SK$=$h(NID_i||M_{us}||N_u||N_s||v||h(Cn_i^*))$, and M_6=$h(SK||M_{us}||N_s)$. Next, it calculates v'=$M_6 \oplus h(N_s||N_u), W_i'$=$v' \oplus h(x||K_u)$, and replaces W_i as W_i' in the user device. Finally, it sends the message $\{M_5, M_6\}$ to S_j.
3) After receiving $\{M_5, M_6\}$, S_j computes Q=$M_5 \oplus h(h(K_s||y)||N_s)$, and M_{us}=$\mathcal{F}_r$ (V_i, Q) by using $\mathcal{F}_r$. Ultimately, it calculates the session key as SK=$h(NID_i||M_{us}||N_u||N_s||v||h(Cn_i^*))$ and validates it by checking $M_6 \stackrel{?}{=} h(SK||M_{us}||Ns)$.

V. Security Analysis

In formal security analysis, we perform the analysis of the proposed scheme using a widely accepted RoR model [29], for proving the mutual authenticity of a shared session key among legitimate entities. In accordance with ROR model, a malicious intruder $\mathcal{I}$ must distinguish a factual session key of instance from a random key. In this model, two entities, i.e. U_i and S_j are supposed to participate in authentication phase. The informal analysis is also followed by the formal analysis in this section. We elaborate security model, semantic security of SK and related proofs below:

A. Security model

Participants: We suppose that $\prod_{S_j}^t$ is t^{th} instance of server S_j, and $\prod_{U_i}^r$ is the r^{th} instance of user U_i, labeled as oracles. **Partnering**: The partner of instance $\prod_{U_i}^r$ regarding U_i is said to be the instance $\prod_{S_j}^t$ of S_j and vice-versa. The partnering-based identity of $\prod_{S_j}^t$ is $pid_{U_i}^r$ for $\prod_{U_i}^r$. The partial transcript of exchanged messages between user and server remains distinct, and forms a session identity as $sid_{U_i}^r$ for the same members.

Freshness: The instances $\prod_{S_j}^t$ or $\prod_{U_i}^r$ are called as fresh with the provision that related SK is not exposed to $\mathcal{I}$. $Adversary$: Considering the ROR model, the $\mathcal{I}$ can modify, block, or remove the communication messages in transit. Alternatively, $\mathcal{I}$ enjoys full authority over an insecure channel considering the following queries:

- $Execute(\prod^t, \prod^r)$: To model an eavesdropping attack, $\mathcal{I}$ can employ this query between U_i and S_j.
- $Send(\prod^t, m)$: The $Send$ query, being modeled as an active threat, permits a participating instance in sending or receiving m.
- $Corrupt\_Device(\prod_{U_i}^t)$: This query models stolen device attack, and using this, $\mathcal{I}$ may access the contents of device.
- $Reveal(\prod^t)$: This query reveals existing session key SK to $\mathcal{I}$ as agreed between $\prod^t$ and respective partner.
- $Test(\prod^t)$: The semantic security of an established SK between U_i and S_j relating to the indistinguishability of ROR model [29], is modeled by $Test$ query. Before the initiation of game, a fair coin c is flipped, while $\mathcal{I}$ stores its outcome to decide afterwards about the consistency of Test query's output. In case, with the use of Test query, the SK is examined to be fresh, then $\prod^t$ outputs SK

if c = 1, or else if c = 0, it returns a random number. Otherwise, it gives null $\perp$.

B. Semantic security of SK

Following ROR model, $\mathcal{I}$ needs to differentiate between the valid SK of instance, and a random secret. $\mathcal{I}$ may issue many test queries to either of the instances, that is, $\prod^t$ or $\prod^r$. The output of Test query must be consistent with random bit c. When the game ends, $\mathcal{I}$ makes a guess of bit c' with a purpose to win. $\mathcal{I}$ wins the game if both bits are equal, that is, c'=c. The $\mathcal{I}$'s advantage for breaking the semantic security of suggested model $\prod$ in time t is symbolized as $Adv_{\prod}^{Akf}(t)$=$|2.Pr[Succs]-1|$, where $Succs$ denotes the event that $\mathcal{I}$ may win the game. The protocol $\prod$ shall be secure in ROR model if the advantage $Adv_{\prod}^{Akf} \leq \vartheta$ for any sufficiently small $\vartheta > 0$.

$Random\ Oracle$: In the scheme, the participating members as well as $\mathcal{I}$ may access one-way hash function in addition to PUF, as simulated by the defined random oracles.

$Definition$ 1: Cryptographic Hashing function. The one-way hashing function $h : \{0,1\}^* \rightarrow \{0,1\}^n$ represents a deterministic function that inputs a string of dynamic length and results in an output string of a predetermined length, say n-bits. If $Adv_{\mathcal{I}}^{Cr\text{-}h}(\tau)$ function denotes $\mathcal{I}$'s advantage in searching for hash collision,

$$Adv_{\mathcal{I}}^{Cr\text{-}h}(\tau) = Pr[(\mu_1, \mu_2) \Leftarrow_R \mathcal{I} : \mu_1 \neq \mu_2 \wedge h(\mu_1) = h(\mu_2)]. \quad (1)$$

An (ζ, τ)-intruder breaking the collision resistance of $h(\cdot)$ suggests that $Adv_{\mathcal{I}}^{Cr\text{-}h}(\tau) \leq \zeta$ with at most run time τ.

$Definition$ 2: Protected PUF. Any PUF, say PUF_a, will be secure iff for two random input challenges $Ch_1, Ch_2 \in \{0,1\}^L$ it generates corresponding output responses $Rs_1, Rs_2 \in \{0,1\}^L$ with minimum variation ds_1 , while alternatively, for any two distinct PUF$_s$ (PUF$_a$, PUF_b), the input challenge Ch_1 must produce distinctive form of responses $Rs_1, Rs_2 \in \{0,1\}^L$ having at least ds_2 amount of variation. Alternatively,

$$Pr[HD(PUF_a(Ch_1), PUF_a(Ch_2)) > ds_1] = 1 - \varepsilon, \quad (2)$$

$$Pr[HD(PUF_a(Ch_1), PUF_b(Ch_1)) > ds_2] = 1 - \varepsilon, \quad (3)$$

where the parameter ε is negligibly small, Ch_1 and Ch_2 are the two challenges chosen randomly by the intruder, HD be the hamming distance, while ds_1 and ds_1 symbolize for thresholds of error tolerance in PUF.

C. Security Proof

The following theorem 1 adequately proves that our protocol supports SK-based security.

Theorem 1: If we assume $\mathcal{I}$ to be a probabilistic polynomial time (PPT) intruder executing $\prod$ in time t where ℓ is the number of bits in B_f. The advantage of intruder for breaking semantic security of contributed model $\prod$ for extracting the session key SK by employing the given number of hash and PUF queries is derived as

$$Adv_{\prod}^{Akf} \leq \frac{q_h^2}{|hash|} + \frac{q_{P_f}^2}{|PUF|} + 2max(C'.q_{se'}^{s'}, \frac{q_{se}}{2^\ell}), \quad (4)$$

where q_h, q_{P_f}, q_{se} represent the number of queries for $hash$, PUF, and $send$ oracles, the $|hash|$, $|PUF|$ correspond to range space in hash and $P_f(\cdot)$ functions, respectively, while the factors C' and s' are Zipf's parameters [19].

Proof: As per the proofs [19], we formulate a sequence of five games denoted as G_e, where $\{0 \leq e \leq 4\}$ for verifying the security of SK in our protocol. The $Succs_j$ characterize an event wherein $\mathcal{I}$ may guess the bit c in G_e. The elaborated discussion about these games is shown below.

Game G_0: This game is supposed to be a factual attack by $\mathcal{I}$ against our biometric authentication scheme in ROR-based model. Since, the bit c should be chosen in the start of G_0, it is obvious that

$$Adv_{\prod}^{Akf}(t) = |2.Pr[Succ_0] - 1| \quad (5)$$

Game G_1: The game G_1 is translated from G_0 by simulating as an eavesdropping attack by $\mathcal{I}$ through initiating Execute $(\prod^t, \prod^r)$. Then, $\mathcal{I}$ needs to initiate Test query for evaluating the variation in legitimate SK and random integer. In our protocol, the session key SK is calculated as SK=$h(UID_i||L_{us}||N_u||N_s||w)$ between S_j and U_i with the use of UID_i, L_{us}, N_u, N_s, and w parameters. Nonetheless, the capturing of $\{w, UID_i^*, M_1, M_2, T_u, M_3, M_4, M_5,$ and $M_6\}$ factors in communication messages would not help $\mathcal{I}$ in evaluating the factors UID_i, L_{us}, N_u, N_s in SK. Since, their computation also requires access to long term private keys (K_s and K_u) and the seizure of PUF_i circuit held with the users. Thus, the probability for wining G_1 by interception of communication factors is not increased. Hence,

$$Pr[Succs_0] = Pr[Succs_1]. \quad (6)$$

Game G_2: The game G_1 is converted into G_2 by adding $Send$ and $hash$-based oracle queries. Due to those queries it may be termed as an active attack while $\mathcal{I}$ may attempt for deceiving a legitimate member by using fabricated and modified messages. $\mathcal{I}$ may issue many queries of Hash oracle to monitor the collisions. It is mention worthy that all openly exchanged parameters in authentication procedure carry the entity's identity, arbitrarily defined variables, and long term private keys. Therefore, no collision is traced if $\mathcal{I}$ initiates multiple $Send$ queries. According to the birthday paradox,

$$|Pr[Succs_2] - Pr[Succs_1]| \leq \frac{q_h^2}{2|hash|}. \quad (7)$$

Game G_3: The G_3 is developed from G_2 with the addition of modeling for $Send$ and PUF oracle-based queries. Thus, pursuing the analogous argument as in G_2 in consideration of secure PUF function, we have

$$|Pr[Succs_3] - Pr[Succs_2]| \leq \frac{q_{P_f}^2}{2|PUF|}. \quad (8)$$

Game G_4: In game G_4, the simulation for $Corrupt\_Device$ is incorporated, such that $\mathcal{I}$ could get the stored information $\{h(\cdot), G_i, H_i, W, r^*, FA^*\}$ from device of the user. Nevertheless, $\mathcal{I}$ cannot extort the identity or private key K_u of user which is safe due to fingerprint-based fuzzy extractor function $B_f \in \{0,1\}^l$. Due to incorporating PUF, the probability for guessing biometrics is B_f is $\frac{1}{2^l}$ [19]. The use of PUF and fuzzy extractor obviates

the need password in registration or authentication phase. The identity ID_i of user is concatenated with the private secret K_u, and is not guessable due to the collision- resistant feature of hash. Therefore, it yields

$$|Pr[Succs_4] - Pr[Succs_3]| \leq max(C'.q_{se'}^{s'}, \frac{q_{se}}{2^{\ell}}) \quad (9)$$

The intruder $\mathcal{I}$ employs all of these queries, while the last possibility of winning the game is only random guessing of the bit c through implementing $Test$ query. Thus, we have

$$|Pr[Succs_4] = \frac{1}{2}. \quad (10)$$

Referring to (5), (6) and (10), we have

$$\frac{1}{2}Adv_{\prod}^{Akf}(t) = |Pr[Succs_0] - \frac{1}{2}| \quad (11)$$

$$= |Pr[Succs_1] - \frac{1}{2}|$$

$$= |Pr[Succs_1] - Pr[Succs_4]| \quad (12)$$

With the application of triangular inequality as well as Eqs. (7), (8), and (9), we have the following outcome:

$$|Pr[Succs_1] - Pr[Succs_4]| \leq |Pr[Succs_1] - Pr[Succs_3]|$$

$$+|Pr[Succs_3] - Pr[Succs_4]| \leq |Pr[Succs_1] - Pr[Succs_2]|$$

$$+|Pr[Succs_2] - Pr[Succs_3]| + |Pr[Succs_3] - Pr[Succs_4]| \leq$$

$$\frac{q_h^2}{|hash|} + \frac{q_{P_f}^2}{|PUF|} + 2max(C'.q_{se'}^{s'}, \frac{q_{se}}{2^{\ell}}) \quad (13)$$

Using Eqs. (11), (12) and (13), it yields

$$Adv_{\prod}^{Akf}(t) \leq \frac{q_h^2}{2|hash|} + \frac{q_{P_f}^2}{2|PUF|} + max(C'.q_{se'}^{s'}, \frac{q_{se}}{2^{\ell}}). \quad (14)$$

■

D. Informal Analysis

1) Supports mutual authentication: Our scheme supports mutual authentication, since the server monitors absolute authenticity of user on the basis of calculation and verification of M_2?=$h(h(K_s||y)||NID_i||N_u||v||T_u)$. The server knows that both parameters, i.e. NID_i as well as K equivalent to $h(K_s||y)$, are protected under the biometric factors of user. Similarly, the user verifies the server on account of the same credentials, which are accessible to server from the use of private secret key K_s. However, the availability of these factors to malicious intruder on simultaneous basis would be hard assumption.

2) Resists impersonation and replay attacks: Our scheme could resist user and server impersonation threats. If an attacker tries to replay or manipulate seized messages for reissuing the authentication request towards server, then the latter may confirm its genuineness by verifying the computed parameter M_2?=$h(h(K_s||y)||NID_i||N_u||v||T_u)$. The involved timestamp T_u may thwart the replay attack on instant basis. However, if we eliminate the timestamp from the protocol, yet the server may nullify the possibility of replay attack during the verification of M_6?=$h(SK||M_{us}||N_s)$. Besides, an intruder may not compute M_2 or M_6 parameters according to the randomly selected nonces N_{uI} and N_{sI} due to lack of $h(K_s||y)$ and NID_i credentials.

```
Completing equations...
Completing equations...
-- Query not attaeker(sk[])
Completing...
Starting query not attacker(sk[])
RESULT not attacker(sk[]) is true.
-- Query inj-event(endCS(idi)) ==> inj-event(beginCS(idi))
Completing...
Starting inj-event(endCS(idi))==> inj-event(beginCS(idi))
RESULT inj-event(endCS(idi)) ==> inj-event(beginCS(idi)) is true.
-- Query inj-event (endUi (idi_1549)) ==> inj-event(beginUi (idi_1549))
Completing...
Starting query inj-event(endUi(idi_1549)) ==> inj-event(beginUi (idi_1549))
RESULT inj-event(endUi(idi_1549)) ==> inj-event(beginUi(idi_1549)) is true.
```

Fig. 3. Simulation Outcome

3) Supports perfect forward secrecy: Our scheme supports forward secrecy if long term secret keys of server or user are compromised by the intruder, it may not compute current or previous session keys SK=$h(NID_i||M_{us}||N_u||N_s||v||h(Cn_i^*))$. In order to compute SK, besides the possession of private secret key K_s, $\mathcal{I}$ also needs access to PUF_i, Cn_i and V_i for computing the session key SK [30], [31].

4) Resists de-synchronization attack: To resist the replay attacks and untraceability problems, many authentication protocols employ synchronization parameters [27] which are updated on both sides with the completion of session. Any A may hold the communication messages by blocking the channel from getting through towards legal entity, to desynchronize the databases on both sides. However, our scheme is resistant to this drawback since the server needs not storing any synchronization parameter on its end [32].

5) Resists DoS attack: In many schemes the authentication information of user is not properly included in its submitted message towards server; as a result the latter will have to frequently consult its secondary storage for verifying the user's identity. For instance, in [20] $\mathcal{I}$ may replay the intercepted messages towards server and initiate a denial of service attack towards server by exploiting the mentioned limitation. However, our scheme is resistant to DoS attack.

6) Resists temporary information threat: In this protocol, if an intruder $\mathcal{I}$ recovers short term temporary secrets from the user's database such as N_u, N_s, still the former would not be able to evaluate the session key SK=$h(NID_i||M_{us}||N_u||N_s||v||h(Cn_i^*))$ because it requires access to U_i's private key K_u as well to compute M_{us}, which is a strong assumption. At the same time, it requires access to $h(K_s||y)$ to compute $h(Cn_i^*)$ for the session key, which makes the scheme resistant to temporary information threat [33].

7) Resists key compromise impersonation (KCI) threat: If long term secret key of user K_u is exposed to $\mathcal{I}$, the later cannot construct a legitimate $\{M_3, M_4\}$ message in response to user's request message for impersonating as a server. Consequently, our scheme is resistant of KCI threat [34].

E. Security Proof using ProVerif

The Proverif provides an automated tool [35] environment to simulate the key agreement scheme and verify its properties on the benchmark of session key's confidentiality and mutual authenticity of involved legal entities. In this connection, we designed the respective modules for verifying the security features of contributed model employing the

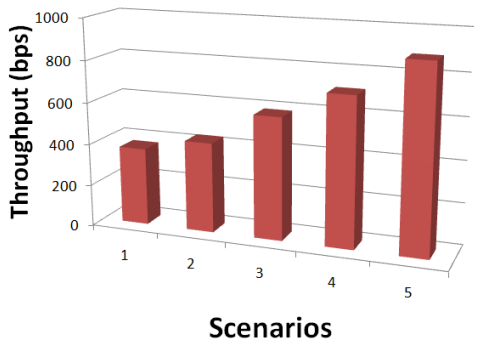

Fig. 4. Throughput

Fig. 5. End-to-End Delay

ProVerif tool. This tool utilizes commonly applied principles of π calculus that could support a number of state-of-the-art crypto-primitives together with hash, symmetric key and public key operations, encryption, and digital signatures etc. For implementing the simulation, we engage two events Ui and CS to execute the respective authentication codes. The entity Ui utilizes two events, that is, beginUi(bitstring) and endUi(bitstring) for authenticating server (CS). In the same manner, the beginCS(bitstring) and endCS(bitstring) events are used by server for authenticating user. The results of queries are computed, which exhibit the stability of the order both pairs of events. The simulation findings are depicted in Fig. 3. which show that the contributed model ensures mutual authenticity for the involved legal participants against the attacker.

F. NS2 Simulation

The experimental demonstration of the proposed scheme using network simulation tool (NS2 2.35) [31] on hardware HP-E8300-Core i5), 2.93 GHz processor with 6GB RAM using Ubuntu 16.12 OS is presetned in this subsection. The network performance of our scheme is measured with the benchmark parameters, i.e., end-to-end delay and throughput. The simulation parameters are given below. The total time of our simulation is 2400 seconds (40 minutes). The mobile users were positioned in rectangular area of 180 $\times$130m^2 with eight rows, while there were 8 users in a single row having inter-distance of 22 meters. We placed 4 cloud servers with 50m of user-based neighbors. A single controlling authority for managing servers was positioned in the network simulation. We employed Adhoc On-Demand Distance Vector (AODV) routing protocol and MAC based wireless channel between user and cloud server, and peer to peer channel between cloud server and controlling authority. We considered all types of static as well as moving users for five different scenarios. The number of users increases from scenario 1 to scenario 5. The speeds for the mobile users range from 2-15 meters per second.

1) End-to-End delay (EED): EED is the average time that takes a message to reach its destination from its source, and can be expressed as $\sum_{i=1}^{N_m}(T_r - T_s)/N_m$, where T_s and T_r represent the sending and receiving time of packet i, respectively. The notation N_m is the aggregate number of messages. The EED includes the sending and receiving time for establishing the mutually agreed session key between the two participants, which is of paramount importance for an authentication protocol, since it must be less for few involved users in the protocol. According to simulation, the EED values range from 0.00382, 0.00453, 0.00535, 0.00621, 0.00753 seconds for five scenarios, respectively. This keeps on increasing with more number of users (8-68) and congestion. Fig. 5 shows the EED of our scheme in milliseconds.

2) Throughput: The throughput characterizes the number of bits transmitted in a unit time. Fig. 4 depicts the throughput of the proposed scheme in bits per second (bps) for three scenarios. The throughput can be expressed as $\frac{N_{pr}\times|PKT_S|}{T_d}$, where N_{pr} is the number of received packets, T_d be the aggregate time in sec, and $|PKT_S|$ is the size of packet. The recorded values of throughput in simulation are 372.41, 432.31, 586.28, 712.73, 885.92 bps for scenarios 1 to 5. The throughput is increased with the increase in the number of exchanged messages due to the more number of users. Thus, the scheme shows high throughput with more number of users.

VI. Efficiency Analysis and Discussion

We demonstrate the comparative analysis of security properties in Table II. Our protocol satisfies all of the mentioned security properties, while all other schemes are prone to at least one or more security limitations. To conduct the analysis based on the computational overheads in different phases among proposed and several related schemes. The analysis of the computational costs for compared schemes is shown in Table IV. In Table IV, the cryptographic operation T_{PUF} represents the PUF-based cost of operation, T_{EPM} the elliptic curve point multiplication, T_m the cost of modular exponentiation operation, T_h the cost of hashing function, T_s the symmetric encryption or decryption cost, T_{FEG} the cost of fuzzy extractor-based generation function, and T_{FEC} the cost of fuzzy extractor-based reproduction function. Although, it is evident that the computational cost of proposed scheme is a bit higher than [7], [21], however it is proved to be resilient against well known attacks. Besides, there is no need of password-based login verification, it requires only identity and fingerprint-based biometric operation captured using fuzzy extractor-in-the-loop.

Our scheme has comparable security features and efficiency than schemes [7], [12], [21]–[23]. Being a symmetric cryptographic protocol, the contributed authentication key agreement is equally practical for sensor networks, internet of things (IoT), and smart devices which are deficient in power resources. For rigorous evaluation of the performance of proposed protocol, the experimental simulation has been

TABLE III
COMPARISON OF SECURITY FUNCTIONALITIES

Schemes	[21]	[22]	[12]	[23]	[20]	[7]	[25]	[27]	[26]	Ours
Supports Anonymity and untraceability	×	×	×	✓	✓	✓	✓	✓	×	✓
Login viability without password	×	×	✓	✓	✓	✓	✓	✓	✓	✓
Resist stolen device attack	×	✓	✓	×	✓	✓	✓	✓	✓	✓
Resist insider attack	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
Resist replay attack	×	✓	×	✓	✓	✓	✓	✓	✓	✓
Resist user impersonation attack	×	×	✓	✓	✓	✓	✓	✓	✓	✓
Resist server impersonation attack	×	×	✓	×	✓	✓	✓	✓	×	✓
Resist Man-in-the-middle attack	×	✓	✓	×	×	✓	✓	✓	✓	✓
Resist known key secrecy attack	✓	✓	×	×	✓	✓	✓	✓	✓	✓
Resist temporary information attack	✓	×	×	✓	✓	✓	✓	✓	✓	✓
Supports perfect forward secrecy	✓	✓	✓	✓	×	×	✓	✓	✓	✓
Supports mutual authentication	×	×	✓	✓	✓	✓	✓	✓	✓	✓
Supports session key agreement	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
Resist denial of service attack	✓	✓	✓	✓	×	✓	✓	✓	✓	✓
Resist de-synchronization attack	×	✓	✓	✓	✓	×	×	×	×	✓
Two factor authentication without password	×	×	×	×	✓	×	×	×	×	✓
Supports efficient symmetric key operations	✓	×	×	×	✓	✓	✓	×	×	✓
Resists Key compromise impersonation attacks	✓	✓	✓	✓	×	✓	✓	✓	✓	✓

× *Feature not supported*, ✓*Feature supported*

TABLE IV
COMPUTATIONAL COSTS

Scheme	Registration phase (Reg)	U_i	S_j	Total	Latency
[21]	$3T_h$	$4T_h$	$10T_h$	$17T_h$	0.143 ms
[22]	$1T_h + 1T_m$	$2T_h + 2T_m$	$4T_h + 5T_m$	$7T_h + 8T_m$	36.46 ms
[12]	$3T_h + 1T_{FEG}$	$6T_h + 2T_m + 1T_{FEC}$	$14T_h + 4T_m + 1T_{FEG} + 1T_{FEC}$	$23T_h + 6T_m + 2T_{FEG} + 2T_{FEC}$	40.28 ms
[23]	$3T_h + 2T_m + 1T_{FEG}$	$5T_h + 5T_m + 1T_{FEC}$	$2T_h + 1T_m$	$10T_h + 6T_m + 1T_{FEG} + 1T_{FEC}$	58.45 ms
[20]	$7T_h + 1T_{PUF} + 1T_{FEG}$	$12T_h + 1T_{PUF} + 1T_{FEG} + 1T_{FEC}$	$7T_h + 1T_{FEC}$	$26T_h + 2T_{PUF} + 2T_{FEG} + 2T_{FEC}$	9.95 ms
[7]	$2T_h + 1T_{PUF} + 1T_{FEG}$	$10T_h + 5T_{PUF} + 1T_{FEC}$	$7T_h$	$21T_h + 6T_{PUF} + 1T_{FEG} + 1T_{FEC}$	4.76 ms
[25]	$3T_h + 1T_{PUF}$	$5T_h + 2T_{PUF} + 1T_{FEG}$	$5T_h + 1t_{FEG}$	$13T_h + 3T_{PUF} + 1T_{FEG} + 1T_{FEC}$	6.21 ms
[27]	$5T_h + 1T_S + 1T_{FEG}$	$7T_h + 2T_{EPM} + 1T_{FEC}$	$6T_h + 2T_{EPM} + 2T_S$	$18T_h + 4T_{EPM} + 2T_S + 1T_{FEC} + 1T_{FEG}$	36.12 ms
[26]	$5T_h + 1T_S$	$7T_h + 2T_{EPM} + 1T_S$	$5T_h + 2T_{EPM} + 2T_S$	$17T_h + 4T_{EPM} + 4T_S$	32.60 ms
Ours	$8T_h + 1T_{PUF}$+1T_{FEG}	$16T_h + 1T_{PUF} + 1T_S + 1T_{FEG} + 1T_{FEC}$	$7T_h + 1T_{FEC} + 3T_S$	$31T_h + 2T_{PUF} + 4T_S + 2T_{FEG} + 2T_{FEC}$	10.19 ms

performed by employing a Smartphone (Lenovo Zuk Z1) bearing Quad-core 2.6 GHz-processor and 6GB RAM having Android OS V5.1.2, and a server PC (HP-E8300-Core i5), 2.93 GHz processor with 6GB RAM using Ubuntu 16.12 OS). The execution timings of all cryptographic operations used in proposed as well as compared schemes are evaluated by using JCE library [16]. Moreover, the 128-bit arbiter PUF is utilized in PUF-based operation, while BCH code has been used for fuzzy extractor (FE)-based generation $Gen(\cdot)$ and reproduction $Rep(\cdot)$ operations.

In accordance with the experimental setting, we get the following outcomes: The hash operation T_h takes 0.029ms on Ui's mobile device, and 0.009ms at the server's end; the PUF operation T_{PUF} takes 0.145ms on Ui's device and 0.68ms at server; the fuzzy extractor-based $Gen(\cdot)$ operation T_{FEG} and $Rep(\cdot)$ operation T_{FEC} takes 3.67 on user's end, while 2.06ms at server; a modular exponentiation operation T_m takes 12.42 ms and 5.78 ms at server, the symmetric encryption or decryption T_S takes 0.062ms on Ui's device and 0.019 ms at server; the elliptic curve based point multiplication operation takes 10.92 ms and 5.16 ms at server. The computational network latency of proposed scheme is computed as 10.19 ms, i.e $23T_h$ + $1T_{PUF}$ +$1T_{FEG}$+ $2T_{FEC}$ +$4T_S$, which is less than schemes [12], [22], [23], [26], [27], but higher than [7], [20], [22], [25] as shown in Table III. The schemes [12], [22], [23], [26], [27] bear significantly higher computational cost due to ECC based point multiplication or modular exponentiation operations. Although, the computational cost of our schemes is higher than schemes [7], [20], [22], [25], yet it is secure and free of security limitations instilled in those protocols, as elaborated in Table III. The Table III shows that only two schemes, that is, [12] and our scheme provide two-factor authentication without the need of password. The [20] suffer from key compromise impersonation attacks. The schemes [12], [22], [23], [26], [27] employ heavy computational operations, that is, non-symmetric key operations. The schemes [7], [25]–[27] suffer de-synchronization attacks. The scheme [27] does not support session key agreement, while [13] is prone to server impersonation threat, offline-password guessing attacks, and also lack mutual authentication, anonymity and untraceability. Similarly, [22] also suffer many security issues as depicted in Table III. It is also evident from the graph in Fig. 6, the proposed scheme supports most of the security features in spite of employing light-weight symmetric key operations. The Table IV represents the number of crypto-primitives used in different schemes, which indicates that the proposed scheme bears high number of low cost hash operations, and low number of high cost operations. In this scheme our objective is to ensure PFS using lightweight symmetric key operations, besides maintaining the other security properties as indicated in Table III.

VII. CONCLUSION

The rise of the fuzzy logic-based biometric applications endows with sufficient intelligence in uncertain biometric-based security solutions to make them reliable. This work, therefore, proposed a two-factor biometric authentication protocol between user and server which, in a novel way, employs the characteristics of PUFs and fuzzy extractor-enabled user biometrics. The intelligent, secure, and dependable protocol could ensure untraceability and perfect forward secrecy, and

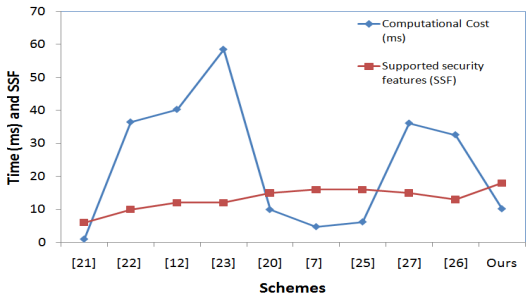

Fig. 6. Comparative findings

is resilient to the DoS and de-synchronization attacks, the main issues posed to the symmetric key-based schemes. Moreover, the presented scheme purges the likelihood of the leakage of the biometric information from the device. The scheme does not need user password in registration and login phases, unlike previous schemes. The security features of the contributed scheme are proved under formal security analysis using the ROR model. In future, we intend to further optimize the mobile user and cloud server interaction with the help of PUF-based fuzzy-in-the-loop solutions in the federated environment.

REFERENCES

[1] G. S. Walia, K. Aggarwal, K. Singh, and K. Singh, "Design and analysis of adaptive graph based cancelable multi-biometrics approach," *IEEE Transactions on Dependable and Secure Computing*, May 2020.

[2] Z. Qin, G. Huang, H. Xiong, Z. Qin, and K.-K. R. Choo, "A fuzzy authentication system based on neural network learning and extreme value statistics," *IEEE Transactions on Fuzzy Systems*, 2019.

[3] S. Garg, K. Kaur, G. Kaddoum, and K.-K. R. Choo, "Towards secure and provable authentication for internet of things: Realizing industry 4.0," *IEEE Internet of Things Journal*, 2019.

[4] A. Irshad, M. Usman, S. A. Chaudhry, H. Naqvi, and M. Shafiq, "A provably secure and efficient authenticated key agreement scheme for energy internet based vehicle-to-grid technology framework," *IEEE Transactions on Industry Applications*, 2020.

[5] A. Rajakumar, G. Raja, A. K. Bashir, J. Chaudry, and A. Ali, "A console grid leveraged authentication and key agreement mechanism for lte/sae," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2677–2689, June 2018.

[6] T. Wang, Z. Zhigao, A. K. Bashir, A. Jolfaei, and Y. Xy, "Finprivacy: A privacy-preserving mechanismfor fingerprint identification," *ACM Transactions on Internet Technology*, 2020.

[7] 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," *IEEE transactions on industrial informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[8] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2884–2895, 2017.

[9] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424–437, 2018.

[10] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "Pmsec: Physical unclonable function-based robust and lightweight authentication in the internet of medical things," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 388–397, 2019.

[11] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, 2020.

[12] X. Li, J. Niu, Z. Wang, and C. Chen, "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.

[13] G. S. Poh, P. Gope, and J. Ning, "Privhome: Privacy-preserving authenticated communication in smart home environment," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[14] K. Zhou and J. Ren, "Passbio: Privacy-preserving user-centric biometric authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 3050–3063, 2018.

[15] M. Wazid, P. Bagga, A. K. Das, S. Shetty, J. J. Rodrigues, and Y. H. Park, "Akm-iov: authenticated key management protocol in fog computing-based internet of vehicles deployment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8804–8817, 2019.

[16] P. Gope, J. Lee, and T. Q. Quek, "Resilience of dos attacks in designing anonymous user authentication protocol for wireless sensor networks," *IEEE Sensors journal*, vol. 17, no. 2, pp. 498–503, 2016.

[17] Y. Zheng, Y. Cao, and C.-H. Chang, "Udhashing: Physical unclonable function-based user-device hash for endpoint authentication," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9559–9570, 2019.

[18] M. Zhang, C. Shen, Z.-G. Wu, and D. Zhang, "Dissipative filtering for switched fuzzy systems with missing measurements," *IEEE transactions on cybernetics*, 2019.

[19] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for iot environment," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3184–3197, 2020.

[20] W. Bian, P. Gope, Y. Cheng, and Q. Li, "Bio-aka: An efficient fingerprint based two factor user authentication and key agreement scheme," *Future Generation Computer Systems*, 2020.

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

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

[23] A. Chaturvedi, D. Mishra, S. Jangirala, and S. Mukhopadhyay, "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.

[24] Y. Wang, J. Wan, J. Guo, Y.-M. 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, 2017.

[25] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2018.

[26] L. Han, X. Tan, S. Wang, and X. Liang, "An efficient and secure three-factor based authenticated key exchange scheme using elliptic curve cryptosystems," *Peer-to-peer Networking and Applications*, vol. 11, no. 1, pp. 63–73, 2018.

[27] A. G. Reddy, A. K. Das, V. Odelu, A. Ahmad, and J. S. Shin, "A privacy preserving three-factor authenticated key agreement protocol for client–server environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 2, pp. 661–680, 2019.

[28] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[29] J. H. Cheon, J. Jeong, D. Kim, and J. Lee, "A reusable fuzzy extractor with practical storage size: Modifying canetti et al.'s construction," in *Australasian Conference on Information Security and Privacy*, 2018, pp. 28–44.

[30] M. Aman, M. H. Basheer, and B. Sikdar, "Data provenance for iot with light weight authentication and privacy preservation," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 441–10 457, 2019.

[31] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in internet of vehicles with blockchain," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[32] M. Aman, M. H. Basheer, S. Dash, J. W. Wong, J. Xu, H. W. Lim, and B. Sikdar, "Hatt: Hybrid remote attestation for the internet of things with high availability," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[33] M. N. Aman, M. H. Basheer, and B. Sikdar, "A lightweight protocol for secure data provenance in the internet of things using wireless fingerprints," *IEEE Systems Journal*, pp. 1–11, 2020.

[34] M. Aman, M. Basheer, and B. Sikdar, "Two-factor authentication for iot with location information," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3335–3351, 2019.
[35] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ecc-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.