IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 20, 2017, accepted April 5, 2017, date of publication April 26, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2694044

On the Design of Fine Grained Access Control With User Authentication Scheme for Telecare Medicine Information Systems

SANTANU CHATTERJEE[1], SANDIP ROY[2], ASHOK KUMAR DAS[3], (Member, IEEE), SAMIRAN CHATTOPADHYAY[4], NEERAJ KUMAR[5], (Member, IEEE), ALAVALAPATI GOUTHAM REDDY[6], (Student Member, IEEE), KISUNG PARK[7], AND YOUNGHO PARK[7]

[1]Research Center Imarat, Defence Research and Development Organization, Hyderabad 500 069, India
[2]Department of Computer Science and Engineering, Asansol Engineering College, Asansol 713 305, India
[3]Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500032, India
[4]Department of Information Technology, Jadavpur University, Kolkata 700 098, India
[5]Department of Computer Science and Engineering, Thapar University, Patiala 147 004, India
[6]KINDI Laboratory, Department of Computer Science and Engineering, Qatar University, Doha 2713, Qatar
[7]School of Electronics Engineering, Kyungpook National University, Daegu 702-701, South Korea

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported by the Ministry of Education, Korea, through the BK21 Plus Project under Grant 21A20131600011.

ABSTRACT A telecare medicine information system (TMIS) for health-care delivery service requires information exchange among multiple IT systems, where different types of users with different access privileges are involved. In TMIS, users generally communicate via public channels. Hence, authentication is essential to provide access to the genuine users. However, access rights for the correct information and resources for different services to the genuine users can be provided with the help of efficient user access control mechanism. The existing user authentication protocols designed for TMIS only provide authentication, but for this kind of application, it is required that the authorized users should also have unique access privilege to access specific data. This paper puts forwards a new fine grained access control with user authentication scheme for TMIS. We present the formal security analysis using both the widely accepted real-or-random model and Burrows–Abadi–Needham logic. The proposed scheme supports user anonymity, forward secrecy, and efficient password change without contacting the remote server. In addition, the proposed scheme is comparable with respect to communication and computation costs as compared with other related schemes proposed in TMIS. Moreover, better tradeoff among security and functionality features, and communication and computation costs makes the proposed scheme suitable and practical for telecare medicine environments as compared with other existing related schemes.

INDEX TERMS Fine-grained access control, biometric authentication, bilinear maps, telecare medicine information systems, fuzzy extractor, security, BAN logic, ROR model.

I. INTRODUCTION

Information and communication technologies are increasingly used in telemedical and healthcare sector to improve medical services with reduced costs. Telecare medicine information system (TMIS) allows delivering personal health assistance to the patient's homes as patients can access healthcare related information on their electronic devices. Also, use of TMIS makes it possible to set up a connection between patients at home and doctors at a clinical center or home healthcare agency.

TMIS involves interconnected healthcare facilities and automated patient electronic medical records (EMRs) that are transmitted between patients and the telecare server. Therefore, a secure communication among connected patients, doctors and the telecare server is an essential requirement. Several authentication schemes are proposed for TMIS to provide security on the issues like authentication, privacy protection and data confidentiality. Unfortunately, none of these schemes discuss on the need of centralized or distributed data access control strategies of the user to access the medical server data.

The basic objectives of a practical telemedical and healthcare system cannot be fulfilled without a proper access control of the user sensitive records stored in the medical server.

The server data may belong to different security levels and is meant to be accessed only by the selected types of users. The problem of assigning unique access privilege to a particular user is called *fine-grained access control*. Fine-grained data access control can identify and impose different access privileges for different types of users. For example, a medical officer or senior doctor should be able to access all types of medical records and diagnostic information of a patient for the purpose of overall treatment, whereas a nurse might only need to check the current sugar level or blood pressure of a patient.

In a hierarchical access control, a user of higher security level class has the ability to access information items (such as message, data, files, etc.) of other users of lower security classes. Several key management techniques in the user hierarchical access control have been proposed in the literature. Achieving fine-grained data access control with an efficient authentication mechanism is an important research area in TMIS. Before allowing access to the sensitive and private data of the patients, an external user (doctor) must be authenticated for a particular access privilege by the medical server. To address this challenge, we propose a new fine-grained access control using smart card and biometric based user authentication scheme, specially tailored for TMIS. To the best of our knowledge, this work is the first one to realize distributed fine-grained data access control with authentication for TMIS.

A. OUR CONTRIBUTIONS

In this paper, we propose a new fine-grained user access control scheme based on attributes. We also use user biometrics and password for authentication purpose. To increase the overall performance, we divide all users into several groups based on the access type, and also introduce key policy attribute based encryption to provide access control with full granularity.

In summary, the following contributions are listed below:

- We introduce the concept of fine-grained data access control of server data with suitable authentication scheme in TMIS.
- The proposed scheme provides password and group-based user authentication depending on the access rights provided for the genuine users in TMIS.
- The proposed scheme provides user anonymity during any message communication that protects patient's privacy. Also, a user never delivers his/her original identity to the the medical server. Hence, the original identity of the user can not be disclosed to an attacker even if the server spoofing attack is executed.
- The proposed scheme provides better security as compared with the other relevant authentication schemes because it resists denial-of-service (DoS), privileged-insider, stolen smart card, replay, man-in-the-middle, password guessing, impersonation and reflection attacks.
- The proposed scheme establishes a secret session key between the user and the medical server so that the established key can be used for future secure communication of the real-time data between them in the telecare system.
- Finally, the proposed scheme provides efficient and flexible way to change a legal user's password locally, which does not require any involvement of the medical server.

B. MOTIVATION

In TMIS, different types of users send different types of data requests to the medical server. The users of this system are of heterogeneous types in nature that include patients, doctors, health staffs, insurance persons, medical researchers, etc. The access privilege of the user, domain and range of data accessibility and the privacy levels of the users are different with respect to a healthcare system. The users having similar features and similar data requirements can constitute a user group with an assigned group identity. Further, based on the user requirements and security levels, information stored in medical server can be classified into several information types, where each type contains a set of data attributes. Hence, having a prior knowledge of intended information type and group identity, a user can achieve attribute-based access control over server data. This allows a user to achieve fine-grained server data access control with full granularity. Till date, several protocols have been developed in TMIS that provide proper user authentication. But none of them delivers a mechanism to provide user authentication with proper access privilege through fine-grained access control in TMIS. This motivates us to develop a fine-grained access control with full granularity with the help of user authentication scheme in TMIS.

C. ADVERSARY MODEL

We apply the Dolev-Yao threat model (DY model) [1]. According to DY model, any two communicating parties communicate over an insecure channel. An attacker has the ability to eavesdrop the transmitted messages over the public insecure channel. In addition, he/she has the ability to alter, modify or delete the contents of the transmitted messages. Furthermore, if the smart card or mobile device of a user is lost or stolen, an attacker will be able to extract all the sensitive information stored in its memory by the power analysis attack on the smart card or the mobile device [2], [3].

D. ORGANIZATION OF THE PAPER

The rest of this paper is organized as follows. In Section II, we briefly review the existing authentication schemes in TMIS. Also, we review the relevant attribute based and group based access control schemes in this section. The related mathematical preliminaries are provided in Section III. In Section IV, we propose a new fine-grained data access control scheme with authentication for TMIS. In Section V, through a detailed security analysis, we analyze how our scheme is resistant to different types of possible well-known attacks. Security, functionality analysis and performance comparison of our scheme with related existing schemes are given in Section VI and Section VII. Finally, we conclude the paper in Section VIII.

II. RELATED WORK

Over last few years, researchers have developed numerous password based authentication schemes using smart card in the field of TMIS [4]–[6]. Along with this, to ensure security and authorized communication, some biometrics or chaotic map based schemes are also developed that provide user anonymity, uniqueness and privacy. Biometric based remote user authentication schemes are introduced in TMIS to provide enhanced security [7]. These schemes can resist stolen smart card attack, off-line password guessing, impersonation attack, etc.

User anonymity preserving scheme with dynamic ID based authentication was proposed for TMIS by Chen *et al.* [8]. A series of enhanced anonymity preserving authentication schemes have been proposed in order to provide better security to the system and to withstand security drawbacks of the earlier schemes [5], [9], [10].

Chaotic map and chaotic hash function based user authentication scheme with key agreement scheme using smart card was proposed by Guo and Chang in TMIS environment [11]. To enhance its security, functionality and performance on the computation and efficiency, several chaotic map based user authentication schemes with smart card have been proposed [10]–[13].

Session key agreement with mutual authentication between a user and the medical server is essential for future secret communication of data in a telecare system. Very recently, researchers have developed authentication schemes with secret shared session key security [11], [12], [14], [15].

Fine-grained access control systems assign unique access privilege to a particular user and allow flexibility in specifying the access rights of individual users. Though several techniques are known for implementing fine-grained access control in different fields, little attention has been received so far to implement it in the field of medical telecare and health sector with proper authentication.

Shamir [16] and Blackley [17] introduced a tree access structure based cryptographic technique known as secret-sharing schemes (SSS). Sahai and Waters proposed Fuzzy Identity-Based Encryption (FIBE) [18] that introduced another cryptographic primitive, called attribute based encryption (ABE). The root idea of FIBE comes from the seminal work of Identity Based Encryption (IBE) proposed by Shamir [19], and it is also based on several primitive works of IBE [20], [21].

A much enriched form of ABE, called Key-Policy Attribute-Based Encryption (KP-ABE) was developed by Goyal *et al.* [22] to achieve fine-grained access control of encrypted data. Their scheme uses the concept of bilinear pairing based cryptographic primitives. Yu *et al.* proposed a scheme to implement the idea of KP-ABE into the field of wireless sensor network (WSN) [23]. Yu *et al.*'s scheme exploits the fundamental cryptographic concepts of KP-ABE technique [22]. Chatterjee and Roy then proposed fine-grained user access control scheme with attribute based encryption using elliptic curve cryptography for hierarchical WSN [24]. KP-ABE techniques are also used in various applications like cloud security [1], [25], enterprise class applications [25] and WSN security [23]–[25].

Chatterjee and Das [26] proposed a novel ECC-based user access control scheme with attribute-based encryption. Recently, Chatterjee *et al.* also designed two fine grained access control schemes for secure data access in cloud networks [27] and enterprise class applications [28]. In addition, Odelu *et al.* proposed a privacy-preserving three-party authentication suitable for battery-limited mobile devices [29]. Generally speaking, these schemes aim to achieve fine grained data access control over user data, but they do not provide proper user authentication as well, which is extremely required for TMIS based applications.

III. MATHEMATICAL PRELIMINARIES

We apply one-way cryptographic hash function, Chebyshev polynomial and chaotic maps, and fuzzy extractor for the proposed authentication scheme and also for analyzing other existing schemes in TMIS. For this purpose, we describe the fundamental concepts on one-way hash function [30], fuzzy extractor on biometrics input, Chebyshev polynomial and chaotic maps [31], [32].

A. COLLISION-RESISTANT ONE-WAY HASH FUNCTION

A one-way cryptographic hash function $H \colon \{0, 1\}^* \rightarrow \{0, 1\}^n$ takes a binary string $q \in \{0, 1\}^*$ of any arbitrary length as an input and produces a binary string $H(q) \in \{0, 1\}^n$ as an output. The collision-resistant property of $H(\cdot)$ is given as follows [33].

Definition 1: The advantage probability of an adversary $\mathcal{A}$ in finding collision with the execution time t is defined as $Adv_{\mathcal{A}}^{HASH}(t) = Pr[(a, b) \in_R \mathcal{A} : a \neq b, H(a) = H(b)]$, where $P_r[E]$ refers to the probability of occurring an event E and $(a, b) \in_R \mathcal{A}$ means the pair (a, b) is randomly selected by $\mathcal{A}$. By an (ϵ, t)-adversary $\mathcal{A}$ attacking the collision resistance of $H(\cdot)$, it is meant that the runtime of $\mathcal{A}$ is at most t and that $Adv_{(A)}^{HASH}(t) \leq \epsilon$.

B. CHEBYSHEV POLYNOMIAL AND ITS PROPERTIES

The Chebyshev polynomial $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ of degree n is defined as

$$T_n(x) = \begin{cases} cos(n \cdot arccos(x)) & \text{if } x \in [-1, 1] \\ cos(n\theta) & \text{if } x = cos\theta, \theta \in [0, \pi]. \end{cases}$$

The recurrence relation of Chebyshev polynomial is as follows

$$T_n(x) = \begin{cases} 1 & \text{if } n = 0 \\ x & \text{if } n = 1 \\ 2xT_{n-1}(x) - T_{n-2}(x) & \text{if } n \geq 2. \end{cases}$$

One can refer to [31] and [32] for the semi-group property of the enhanced Chebyshev polynomial and Chaotic map-based discrete logarithm problem (CMDLP). The CMDLP is stated as follows. For any given x and y, it is computationally infeasible to find an integer s such that $T_s(x) = y$.

Bergamo *et al.* [34] described an attack that allows one to compute an integer solution s from the equation $T_{s'}(x) = T_s(x)$ if both $T_s(x)$ and x ($x \in [-1, +1]$) are known by computing $s' = \frac{arccos(T_s(x))+2k\pi}{arccos(x)}$, $k \in \mathcal{Z}$, where $\mathcal{Z}$ is the set of all integers.

C. BIOMETRICS AND FUZZY EXTRACTOR

Biometric keys, such as iris, fingerprint and palmprint, are now increasingly used in several authentication protocols due to their uniqueness property [35]–[38]. The major advantages of using the biometric keys are (i) they are extremely hard to forge or distribute, (ii) they are extremely difficult to copy or share, and (iii) they can not be lost or forgotten as they can not be guessed easily [39], [40].

Recently, the fuzzy extractor method has been used effectively in extracting biometric key from a given user biometric input [4], [41]. The fuzzy extractor takes a biometric feature input, say $\mathcal{B}$ from user and exploits a probabilistic generation function in a permissible error tolerant manner to generate the unique random string, say α and the auxiliary string, say β. Further, using a deterministic reproduction procedure, it generates the same original string α, with auxiliary string β and a noisy user biometric $\mathcal{B}'$ that differs from the original biometric $\mathcal{B}$ up to a threshold value [29], [42].

The fuzzy extractor is defined by five tuples $(\mathcal{M}, \lambda, \tau, m, \delta)$ along with two algorithms $Gen(\cdot)$ and $Rep(\cdot)$.

- $\mathcal{M} = \{0, 1\}^v$ represents a metric space of biometric data points with finite dimension. The distance function $\Delta : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{Z}^+$ calculates the similarity between two different biometric inputs $\mathcal{B}_1$ and $\mathcal{B}_2$.
- λ is the length (in bits) of unique string α.
- τ is the permissible error tolerance.
- m is the min-entropy of a probability distribution W on metric space $\mathcal{M}$.
- δ is the allowable maximum statistical distance between two probability distributions $\langle \alpha_1, \beta \rangle$ and $\langle \alpha_2, \beta \rangle$.

The functions $Gen(\cdot)$ and $Rep(\cdot)$ are defined as follows:

- *Gen*: It is defined as $\langle \alpha, \beta \rangle \leftarrow Gen(\mathcal{B})$, where $\alpha \in \{0, 1\}^\lambda$ and $\mathcal{B} \in \mathcal{M}$ such that statistical distance between the probability distributions $\langle \alpha, \beta \rangle$ and $\langle \alpha_1, \beta \rangle$, $SD(\langle \alpha, \beta \rangle, \langle \alpha_1, \beta \rangle) \leq \delta$. Here, α_1 refers a uniform binary string of length λ, where $\lambda = m - 2\log(\frac{1}{\delta}) + O(1)$ [4], [41].
- *Rep*: It is defined as follows: $\forall \mathcal{B} \in \mathcal{M}$, $\forall \mathcal{B}' \in \mathcal{M}$ and $\Delta(\mathcal{B}, \mathcal{B}') \leq \tau$ such that if $\langle \alpha, \beta \rangle \leftarrow Gen(\mathcal{B})$, then $Rep(\mathcal{B}', \beta) = \alpha$.

Suppose $\mathcal{I}$ is a string of 2^k elements, with $k < n$. Further, assume that (i) $\mathcal{I}_e$: $\mathcal{M} \rightarrow \mathcal{I}$ is an encoding function (one-to-one), and (ii) $\mathcal{I}_d$: $\{0, 1\}^n \rightarrow \mathcal{I}$ is a decoding function (error tolerant up to τ bits). Then $Gen(\mathcal{B})$ outputs $\alpha = H(\mathcal{B})$ and public parameter $\beta = \mathcal{B} \oplus \mathcal{I}_e(\alpha)$. Taking noisy biometric $\mathcal{B}'$ and public parameter β, $Rep(\mathcal{B}', \beta)$ generates $\alpha' = \mathcal{I}_d(\mathcal{B}' \oplus \beta) = \mathcal{I}_d(\mathcal{B}' \oplus \mathcal{B} \oplus \mathcal{I}_e(\alpha)) = \mathcal{I}_d(\mathcal{I}_e(\alpha)) = \alpha$, if the condition $\Delta(\mathcal{B}, \mathcal{B}') \leq \tau$ is satisfied.

TABLE 1. Notations used in this paper.

Symbol	Description
MAS	Medical application server
U_j	j^{th} user
ID_{U_j}	Unique identifier of U_j
SC_j	Smart card of U_j
$\mathcal{I}$	Universe of all server attributes
$\vert G \vert$	Order of group G
$H(\cdot)$	Secure one-way hash function
MK_s	Master key of server MAS
$A \vert\vert B$	Data A concatenates with data B
$E_K()$	Symmetric key encryption using the key K
$D_K()$	Symmetric key decryption using the key K
T_{U_j}, T_s	Current timestamps of U_j and MAS, respectively
$Gen(\cdot)$	Fuzzy extractor generation function
$Rep(\cdot)$	Fuzzy extractor reproduction function
$\triangle T$	Maximum transmission delay

D. BILINEAR PAIRING AND BILINEAR MAP

The fundamentals of bilinear map are described briefly [22].

Bilinear Map: Let G_1, G_2 and G_T be multiplicative cyclic groups of prime order p. Let g_1 and g_2 be generators of G_1 and G_2, respectively. A bilinear map is an injective function $e : G_1 \times G_2 \rightarrow G_T$ with the following three properties:

1) *Bilinearity:* For all $u \in G_1$, $v \in G_2$, a, $b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
2) *Non-degeneracy:* $e(g_1, g_2) \neq 1$, 1 is the identity in G_T.
3) *Computability:* There is an efficient algorithm to compute $e(u, v)$ for each $u \in G_1$ and $v \in G_2$.

IV. THE PROPOSED FINE GRAINED ACCESS CONTROL SCHEME

In this section, we first tabulate the important notations that are useful to explain and analyze our scheme. We then explain in detail the various phases related to our scheme.

A. NOTATIONS

We use the notations listed in Table 1 to describe and analyze the proposed scheme. We use the secure hash standard (SHA-1) [43] as one-way cryptographic hash function. For symmetric key encryption/decryption, we apply the Advanced Encryption Standard (AES-128) [44] in our proposed scheme. Note that for better security, one can also consider SHA-256 as one-way cryptographic hash function [43].

B. DESCRIPTION OF THE SCHEME

In this section, we describe how our proposed fine grained user access control scheme works using attribute based access control. The proposed scheme consists of five phases: 1) setup, 2) registration, 3) login, 4) authorization, and 5) password change. These phases are discussed in detail in the following subsections. We make use of the current timestamps in order to prevent the replay attack. For this reason, we assume that all the entities in TMIS are synchronized with their clocks.

1) SETUP PHASE

This phase is used to pre-load keying materials to the medical application server and user smart card prior to start working. The server *MAS* chooses a set of *network parameters* using the following steps:

- **Step 1:** *MAS* chooses two multiplicative cyclic groups G_1 and G_T of prime order p as well as a bilinear map e: $G_1 \times G_1 \rightarrow G_T$. Let g be the generator of G_1.
- **Step 2:** *MAS* chooses a number t_a uniformly at random from Z_p for each attribute $a \in \mathcal{I}$, and selects a random number $y \in Z_p$, where $Z_p = \{0, 1, \ldots, p-1\}$. *MAS* then computes $Y = e(g, g)^y \pmod p$, and $T_1 = g^{t_1} \pmod p$, $T_2 = g^{t_2} \pmod p$, $\ldots$, $T_{|\mathcal{I}|} = g^{t_\mathcal{I}} \pmod p$.
- **Step 3:** *MAS* establishes a universe of all information types IT. It further creates n smaller disjoints sets of information types $IT_1, IT_2, IT_3, \ldots, IT_n$, which are subsets of IT. Hence, $IT = \bigcup_{i=1}^{n} IT_i$. Each user U_j of the healthcare system requests server information through its assigned group identity GID_j. U_j, using its own group identity GID_j, can access information from one or more suitable information types IT_i, where $IT_i \subset IT$. Further, an information type IT_i might belong to more than one user group identities. Every information type IT_i contains a number of relevant server attributes that provides the necessary server information to a user U_j.
- **Step 4:** Finally, *MAS* assigns a unique randomly generated master key, say MK_S for its own. In addition, *MAS* selects a one-way cryptographic hash function $H(\cdot)$ (for example, SHA-1 [43]).

2) REGISTRATION PHASE

In the registration phase, a user U_j needs to register with the *MAS* for accessing medical data. This phase consists of the following steps:

- Step 1: U_j first chooses his/her identity ID_{U_j} and password PW_j, and then imprints personal biometrics B_j on the sensor of a specific device.
- Step 2: U_j selects a 160-bit random number $r_j \in Z_p$. U_j generates $(\alpha_j, \beta_j) = Gen(B_j)$, where $Gen(\cdot)$ is a fuzzy extractor generation procedure. U_j further computes the masked password $W_j = H(\alpha_j \,||PW_j)$ and calculates $A_{ID_j} = H(\alpha_j \,||ID_{U_j} \,||r_j)$. Next, it chooses its access group id GID_j and then sends the registration request message $\langle A_{ID_j}, W_j, \alpha_j, GID_j \rangle$ to *MAS* via a secure channel.
- Step 3: *MAS* selects a unique server id S_{ID}, and keeps the information A_{ID_j} and GID_j. Further, for each user U_j, it generates $A_j = H(A_{ID_j} \,||TS_{U_j})$, where TS_{U_j} denotes the registration time stamp of U_j. It then calculates the secret parameter $R_{U_j} = H(W_j \,||A_j \,||GID_j)$ for each user U_j.
- Step 4: Finally, the *MAS* computes the secret shared parameter with U_j as $X_j = H(\alpha_j \,||S_{ID}) \oplus H(MK_S \,||A_j)$ for U_j.

This phase has two sub-phases, namely *access structure generation* and *smart card generation*, which are discussed below.

a: ACCESS STRUCTURE GENERATION

The *MAS* selects an access structure P_j for each user U_j. After getting the registration information from valid users, the *MAS* assigns each user an access structure. The access structures are implemented via an access tree. Every leaf node of the access tree is labeled with an attribute and the internal nodes are threshold gates. Access structures are represented using the logic expressions over the attributes. With the help of the access tree, the data access privileges of each user can be defined.

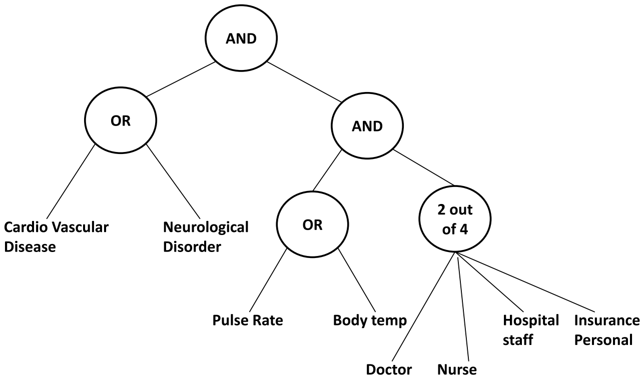


FIGURE 1. User access structure.

For example, consider a scenario as explained in [24]. The medical server can store information on many "in-body" diseases like cardiovascular problem, neurological disorder, etc. (in-body attribute). Suppose the *MAS* can measure some "on-body" parameters like body temperature, pulse rate, etc (on-body attribute). The medical records have multiple relevant users like doctor, nurse, hospital staffs, etc. Hence, a medical record stored in the server can be specified with these attributes [inbody = {cardiovascular disease, neurological disorder, cancer}, on-body = {pulse rate, body temperature} and owner = {doctor, nurse, hospital stsff}]. The medical application server provides each user an access policy via a user access tree. A user can decrypt data through its access tree only if it has matching attributes with the data sent by the medical server. A user U_j with the access structure is provided in Figure 1, who can decrypt the server data stored within a medical server that detects in-body disease likes cardiovascular disease or neurological disorder, and contains on-body measuring attributes as pulse rate or body temperature, and at least owned by 2 out-of 4 experts like doctor, nurse, hospital staff or medical insurance person.

For each user U_j, the server generates an access structure P_j and computes the secret key SK_j. Starting from the root node r of P_j and in the top-down manner, *MAS* also constructs a

User (U_j)	Server (MAS)
REGISTRATION PHASE	
Select ID_{U_j}, PW_j and r_j. Imprint B_j. Compute $(\alpha_j, \beta_j) = Gen(B_j)$, masked password $W_j = H(\alpha_j \vert\vert PW_j)$, and $A_{ID_j} = H(\alpha_j \vert\vert ID_{U_j} \vert\vert r_j)$. Choose access group id GID_j.	
$\xrightarrow[\textbf{(secure channel)}]{\langle A_{ID_j}, W_j, \alpha_j, GID_j \rangle}$	
	Store A_{ID_j}, GID_j and server id S_{ID}. Calculate $A_j = H(A_{ID_j} \vert\vert TS_{U_j})$, $R_{U_j} = H(W_j \vert\vert A_j \vert\vert GID_j)$ and $X_j = H(\alpha_j \vert\vert S_{ID_j}) \oplus H(MK_S \vert\vert A_j)$. Generate access structure P_j for each U_j. Compute secret key $SK_j = \langle \{ D_i = g^{\frac{q_i(0)}{t_i}} \}_{i \in \mathcal{L}} \rangle$.
	$\xleftarrow[\textbf{SMART CARD (secure channel)}]{\langle A_{ID_j} \oplus SC_{id_j}, GID_j, R_{U_j}, r_j \oplus \alpha_j, TS_{U_j} \oplus SC_{id_j}, P_j, SK_j, e: G_1 \times G_2 \rightarrow G_T, X_j, H(\cdot) \rangle}$
Store β_j, $Gen(\cdot)$, $Rep(\cdot)$ and τ in the smart card. Finally, $SC_j = \{A = (A_{ID_j} \oplus SC_{id_j})$, GID_j, $B = (TS_{U_j} \oplus SC_{id_j})$, $C = (r_j \oplus \alpha_j)$, P_j, SK_j, R_{U_j}, $e(\cdot)$, X_j, $H(\cdot)$, β_j, $Gen(\cdot)$, $Rep(\cdot)$, $\tau\}$.	Delete R_{U_j} from records. Store $\{A_j, MK_s, TS_{U_j}\}$ in its database.
LOGIN PHASE	
Insert smart card SC_j into card reader. Input PW_j and identity ID_j. Imprint B'_j. Compute $\alpha_j = Rep(B'_j, \beta_j)$, $W'_j = H(\alpha_j \vert\vert PW_j)$, $A'_{ID_j} = H(\alpha_j \vert\vert ID_{U_j} \vert\vert r_j)$, $SC'_{id_j} = (A_{ID_j} \oplus SC_{id_j}) \oplus A'_{ID_j}$, $TS'_{U_j} = (TS_{U_j} \oplus SC_{id_j}) \oplus SC'_{id_j}$, $A'_j = H(A_{ID_j} \vert\vert TS'_{U_j})$, and $R'_{U_j} = H(W'_j \vert\vert A'_j \vert\vert GID_j)$. Check if $R'_{U_j} \overset{?}{=} R'_{U_j}$. If verification holds, compute $S_j = X_j \oplus H(\alpha_j \vert\vert S_{ID})$. Select information type IT_i. Compute $N_j = H(S_j \vert\vert H(n_j \vert\vert GID_j) \vert\vert T_{U_j} \vert\vert IT_i)$.	
$\xrightarrow[\textbf{(public channel)}]{\langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j \vert\vert GID_j)) \rangle}$	

FIGURE 2. User registration and login phases of our proposed scheme.

random polynomial q_x of degree $d_x - 1$ using the Lagrange interpolation [45] for each node $x \in P_j$, where d_x is the degree of a node x. For each non-root node $x \in P_j$, it sets $q_x(0) = q_{parent(x)}(index(x))$, where $parent(x)$ is the parent of x, and x is the $index(x)^{th}$ child of its parent. In particular, we have $q_r(0) = y$. The user secret key SK_j is the output, which is derived as follows:

$$SK_j = \langle \{ D_i = g^{\frac{q_i(0)}{t_i}} \}_{i \in \mathcal{L}} \rangle,$$

where $\mathcal{L}$ denotes the set of leaf nodes and g is the generator of G_1.

b: SMART CARD GENERATION

The *MAS* generates a smart card with valid identity SC_{id_j} for user U_j with the following parameters: $A = (A_{ID_j} \oplus SC_{id_j})$; GID_j; $B = (TS_{U_j} \oplus SC_{id_j})$; $C = (r_j \oplus \alpha_j)$; P_j; SK_j; R_{U_j}; e: $G_1 \times G_1 \rightarrow G_T$; X_j; $H(\cdot)$. The *MAS* then deletes the user's secret parameter R_{U_j} from records as soon as the registration procedure of U_j is over. However, it keeps A_j and GID_j for each user U_j. Finally, U_j stores β_j, $Gen(\cdot)$, $Rep(\cdot)$ and τ into the smart card SC_j, where τ is the permissible error tolerance value used in $Rep(\cdot)$ function.

This registration phase is summarized in Figure 2.

3) LOGIN PHASE

The purpose of this phase is to login to the system by a legal user U_j, who wants to access any specific data from the *MAS*. This login phase is further summarized in Figure 2. U_j performs the following steps:

- Step 1: U_j first inserts his/her smart card SC_j into the card reader of a specific terminal and imprints his/her personal biometrics B_j. U_j also inputs his/her password PW_j and identity ID_j.
- Step 2: Using the fuzzy extractor reproduction procedure $Rep(\cdot)$ and stored β_j, SC_j computes $\alpha_j = Rep(B'_j, \beta_j)$,

User (U_j)	Server (MAS)
	Check $\|T_{U_j} - T^*_{U_j}\| \overset{?}{\leq} \triangle T$. If verification holds, compute $S'_j = H(MK_S\|\|A_j)$, $H(n_j\|\|GID_j) = TS_{U_j} \oplus (TS_{U_j} \oplus H(n_j\|\|GID_j)$. Check $H(S'_j\|\|H(n_j\|\|GID_j)\ \|\|T_{U_j}\|\|IT_i) \overset{?}{=} N_j$. If verification holds, check if received $IT_i \in GID_j$. Select session key K_s and random number α. Calculate $E_j = \{\{T_i{}^{\alpha}\}_{\forall i \in IT_i}\}_{\forall IT_i \in GID_j}$. Compute $E' = K_s Y^{\alpha}$. Generate a puzzle message PM and encrypt it as $E_{K_s}(PM)$. Compute $H(A_j\|\|PM\|\|E'\|\|T_s\|\|T_{U_j})$, $K_j = (X_j \oplus S'_j) \oplus H(H(n_j\|\|GID_j)\|\|RN_s)$. $\xleftarrow{\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s\|\|PM\|\|E'\|\|T_s\|\|T_{U_j}), E_j \rangle}$ **Public channel**
Check $\|T_s - T^*_s\| \overset{?}{\leq} \triangle T$. Compute $K'_j = H(W_j\|\|S_{ID}) \oplus H(H(n_j\|\|GID_j)\|\|RN_s)$. Check $K'_j \overset{?}{=} K_j$. If verification holds, using access tree, obtain K_s. Compute $PM^* = D_{K_s}(E_{K_s}(PM))$ Check if $H(RN_s\|\|PM^*\|\|E'\|\|T_s\|\|T_{U_j})$ $\overset{?}{=}$ received hash value. If verification holds, compute $PM' = H(PM^*\|\|RN_j\|\|T_i)$. Establish session key $SK_{U_j,S}$ as $SK_{U_j,S} = H(PM\|\|\ RN_j\|\|\ RN_s\ \|\|K_s\ \|\|T_{U_j}\ \|\|T_s\ \|\|A_j)$. $\xrightarrow{\langle H(PM'\|\|RN_s\|\|M_j), M_j, E_{K_s}(PM'), RN_j \rangle}$ **Public channel**	
	Decrypt PM as $PM' = D_{K_s}(E_{K_s}(PM'))$. Compute $PM'' = H(PM\|\|RN_j\|\|T_i)$ using its own PM. Check $PM'' \overset{?}{=} PM'$. If verification holds, access granted to U_j. Establish session key SK_{S,U_j} as $SK_{S,U_j} = H(PM\ \|\|RN_j\ \|\|RN_s\ \|\|K_s\ \|\|T_{U_j}\ \|\|T_s\|\|A_j)$.
Store session key $SK_{U_j,S}$ $(= SK_{S,U_j})$.	Store session key SK_{S,U_j} $(= SK_{U_j,S})$.

FIGURE 3. Authorization phase of our proposed scheme.

masked password $W'_j = H(\alpha_j\ \|\|PW_j)$, $r_j = C \oplus \alpha_j$, and computes $A'_{ID_j} = H(\alpha_j\ \|\|ID_{U_j}\ \|\|r_j)$. From the stored parameter A, U_j computes the smart card identity SC'_{id_j} as $SC'_{id_j} = A \oplus A'_{ID_j}$. Next, using this computed SC'_{id_j}, it finds out the user registration timestamp $TS'_{U_j} = B \oplus SC'_{id_j} \oplus SC'_{id_j}$. With the computed registration timestamp TS'_{U_j}, it then computes $A'_j = H(A'_{ID_j}\ \|\|TS'_{U_j})$ and computes $R'_{U_j} = H(W'_j\ \|\|A'_j\ \|\|GID_j)$. Finally, it checks if the condition $R'_{U_j} = R_{U_j}$ holds. If this verification does not hold, it indicates that U_j has entered one or more wrong parameters in giving his/her identity, password or biometrics, and the phase terminates immediately.

- Step 3: U_j selects the suitable information type IT_i for which he/she wants to access the server information. U_j then computes $S_j = X_j \oplus\ H(\alpha_j\ \|\|S_{ID_j})$. U_j selects a random secret value n_j and computes $N_j = H(S_j\ \|\| H(n_j\ \|\|GID_j)\ \|\|T_{U_j}\ \|\|IT_i)$, where T_{U_j} is the current time stamp of U_j.
- Step 4: U_j sends the message $\langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j\ \|\|GID_j))\rangle$ to the MAS via open channel.

4) AUTHORIZATION PHASE

In this phase, a mutual authentication between a user U_j and the server MAS takes place. At the end of this phase, both U_j and MAS establish a session key for their future secure communication. This phase is summarized in Figure 3. This phase involves the following steps:

- Step 1: After receiving of the user request message in the login phase, the MAS first checks the validity of the received timestamp T_{U_j} by the condition $|T_{U_j} - T^*_{U_j}| < \triangle T$, where $T^*_{U_j}$ is the time when the message is received by the MAS and $\triangle T$ is the maximum transmission delay. If the condition does not hold, it means that it is a replay message and the phase is terminated immediately by the MAS.
- Step 2: The MAS calculates $S'_j = H(MK_S\ \|\|A_j)$, $H(n_j\|\|\ GID_j) = TS_{U_j} \oplus\ (TS_{U_j}\ \oplus H(n_j\ \|\|GID_j))$ and

$N_j' = H(S_j' || H(n_j || GID_j) || T_{U_j} || IT_i)$, and then checks the condition $N_j' = N_j$. If this verification does not hold, the authentication request fails and the phase terminates.

- Step 3: The MAS further checks whether $IT_i \in IT$ and $IT_i \in GID_j$. If both conditions satisfy, the user group is authorized to access the requested information type. The MAS then selects an access session key K_s for accessing the data under information type IT_i such that U_j will only get the session key if he/she has proper access privilege.
- Step 4: The MAS selects a random number $\alpha \in Z_p$ and calculates $E_j = \{\{{T_i}^{\alpha}\}_{\forall i \in IT_i}\}_{\forall IT_i \in GID_j}$. The MAS computes $E' = K_s Y^{\alpha}$. Following the strategy of the challenge-response protocol, the MAS can create a puzzle message PM and computes an encrypted puzzle using its computed key K_s as $E_{K_s}(PM)$. Also, it generates a hash value $H(A_j || PM || E' || T_s || T_{U_j})$, where T_s is the current timestamp of the MAS. The MAS also computes $K_j = (X_j \oplus S_j') \oplus H(H(n_j || GID_j) || RN_s)$, where RN_s is a random nonce generated by the MAS. Note that $(X_j \oplus S_j')$ is equal to $H(B_j || S_{ID})$. Finally, the MAS sends the message $\langle K_j,\ RN_s,\ IT_i,\ E',\ E_{K_s}(PM),\ T_s,\ H(RN_s || PM || E' || T_s || T_{U_j}),\ E_j \rangle$ to to the user U_j via a public channel.
- Step 5: After receiving the message from the MAS in Step 4, U_j first checks if $|T_s - T_s^*| < \bigtriangleup T$ for checking the validity of the received timestamp T_s, where T_s^* is the time when the message $\langle K_j,\ RN_s,\ IT_i,\ E',\ E_{K_s}(PM),\ T_s,\ H(RN_s || PM || E' || T_s || T_{U_j}),\ E_j \rangle$ is received by U_j. Next, U_j (that is, the smart card SC_j) computes $K_j' = H(MB_j || S_{ID}) \oplus H(H(n_j || GID_j) || RN_s)$ and verifies it against the received value K_j. If this verification holds, it then proceeds for the next step; otherwise the phase is terminated immediately.
- Step 6: For accessing the attributes under information type IT_i, U_j decrypts the encrypted key K_s and retrives the puzzle message PM. For this purpose, U_j uses a recursive algorithm as follows. The decryption process starts from the leaf nodes of its own access tree P_j and continues in the bottom-up manner. U_j computes F_i for each leaf node x in P usng the following logic:
  If $(i \in \mathcal{I}_i)$, $F_i = e(D_i, E_i) = e(g^{q_x(0)/t_i}, g^{t_i \alpha}) = e(g, g)^{\alpha q_x(0)})$. Otherwise, set $F_i = \perp$ (null).
  If the access structure P_j "accepts" $\mathcal{I}_i$, it means all the attributes specified for the information type IT_i are matched with the user access structure and U_j will finally obtain $e(g, g)^{\alpha qr(0)} = e(g, g)^{\alpha y}$. Since $Y = e(g, g)^{y}$, U_j will obtain Y^{α}. So, using computed Y^{α}, U_j computes K_s as $K_s = E'(Y^{\alpha})^{-1} \pmod p$. Thus, U_j is able to decrypt the puzzle message PM using K_s. Otherwise, the decryption algorithm returns $\perp$ (null).
- Step 7: After getting the value of PM, U_j computes $H(RN_s || PM || E' || T_s || T_{U_j})$ and checks it with the received hash value in the login message. If these values are not equal, the phase terminates. Otherwise, U_j generates a random nonce RN_j and calculates $PM' = H(PM || RN_j || T_s)$, where T_s is the current timestamp of the MAS. U_j then sends the message $\langle H(PM' || RN_s || M_j),\ M_j,\ E_{K_s}(PM'),\ RN_j \rangle$ to the MAS for accessing data M_j. For future message communication, U_j creates a secret session key $SK_{U_j,S} = H(PM || RN_j || RN_s || K_s || T_{U_j} || T_s || A_j)$ shared with the MAS.
- Step 8: After receiving the message $\langle H(PM' || RN_s || M_j),\ M_j,\ E_{K_s}(PM'),\ RN_j \rangle$, the MAS decrypts the encrypted puzzle $E_{K_s}(PM')$ using K_s and gets PM'. It the computes $PM'' = h(PM || RN_j || T_s)$ with its own PM, T_s and the received RN_j. If $PM'' = PM'$, the MAS computes a hash value $H(PM' || RN_s || M_j)$ with received M_j and stored RN_s. If this computed hash value is same as that of the received hash value, the MAS grants the access permission for the data M_j to U_j for the current session. Finally, for the current session, the MAS also establishes a secret session key $SK_{S,U_j} = H(PM || RN_j || RN_s || K_s || T_{U_j} || T_s || A_j)$ for future message communication with U_j.

5) PASSWORD CHANGE PHASE

In this phase, any user U_j can change his/her password freely and completely locally without the help of the MAS. This phase contains the following steps:

- Step 1: U_j inserts his/her smart card into the card reader of a specific terminal and provides his/her identity ID_{U_j} and the old password PW_j^{old}, and also imprints his/her personal biometrics B_j'. After that SC_j computes $\alpha_j = Rep(B_j', \beta_j)$ and generates $W_j^{old} = H(\alpha_j || PW_j^{old})$. Further, SC_j computes $A_{ID_j} = H(\alpha_j || ID_{U_j} || r_j)$ and finds out the smart card identity SC_{id_j} from $(A_{ID_j} \oplus SC_{id_j})$. After that SC_j also computes registration timestamp TS_{U_j} from $(TS_{U_j} \oplus SC_{id_j})$ using the computed smart card identity SC_{id_j}. Furthermore, SC_j computes $A_j' = H(A_{ID_j} || TS_{U_j})$ using the computed value of A_{ID_j} and TS_{U_j}.
- Step 2. SC_j computes $R_{U_j}^{old} = H(W_j^{old} || A_j' || GID_j)$ and checks if the condition $R_{U_j}^{old} = R_{U_j}$ is satisfied. If they do not match, it means that U_j has entered his/her old password, identity as well as biometrics incorrectly, and the password change phase terminates immediately. Otherwise, SC_j asks U_j to enter a new changed password PW_j^{new} in the smart card.
- Step 3: The smart card SC_j computes the new masked password $W_j^{new} = H(\alpha_j || PW_j^{new})$ and $R_{U_j}^{new} = H(W_j^{new} || A_j' || GID_j)$.
- Step 4: Finally, SC_j replaces R_{U_j} with the newly computed masked password $R_{U_j}^{new}$ in its memory.

This phase is also summarized in Figure 4.

V. SECURITY ANALYSIS

In this section, through the formal and informal security analysis, we show that our scheme can resist various known attacks, which are given in the following subsections.

A. FORMAL SECURITY ANALYSIS USING REAL-OR-RANDOM MODEL

We present the formal security analysis of the proposed fine-grained access control scheme through the widely-used

User (U_j)	Smart card (SC_j)
Insert smart card SC_j into card reader. Provide identity ID_{U_j} and old password PW_j^{old}. Imprint personal biometrics B_j'.	
	Calculate $\alpha_j = Rep(B_j', \beta_j)$, $W_j^{old} = H(\alpha_j \|\| PW_j^{old})$, $A_{ID_j} = H(\alpha_j \|\| ID_{U_j} \|\| r_j)$. Retrieve smart card identity SC_{id_j} from $(A_{ID_j} \oplus SC_{id_j})$. Compute registration timestamp TS_{U_j} from $(TS_{U_j} \oplus SC_{id_j})$ using SC_{id_j}, $A_j' = H(A_{ID_j} \|\| TS_{U_j})$, $R_{U_j}^{old} = H(W_j^{old} \|\| A_j' \|\| GID_j)$. Verify the condition $R_{U_j}^{old} = R_{U_j}$. If it is valid, SC_j asks U_j to enter new password PW_j^{new}.
Input new changed password PW_j^{new}.	
	Calculate new masked password $W_j^{new} = H(\alpha_j \|\| PW_j^{new})$, $R_{U_j}^{new} = H(W_j^{new} \|\| A_j' \|\| GID_j)$. Replace R_{U_j} with the newly computed $R_{U_j}^{new}$ in its memory.

FIGURE 4. **Password change phase of our proposed scheme.**

Real-Or-Random model [46]. Random oracles are considered under a formal security model. An adversary $\mathcal{A}$ can make several oracle queries, which model the adversary's capabilities in a real attack. To proof the formal security of our scheme, we consider all possible oracle queries.

We simulate various security attacks on the proposed scheme, say $\mathcal{P}$ through the following oracle queries:

- *Send(U_j/MAS, m)*: Through this query $\mathcal{A}$ sends a request message m to $\mathcal{P}^t$, and $\mathcal{P}^t$ replies to $\mathcal{A}$ according to the rules of the protocol.
- *Execute(U_j, MAS)*: This query enables $\mathcal{A}$ with a capability to eavesdrop message m communicated between U_j and MAS in an actual execution of the protocol.
- *Corrupt(U_j, a)*: Depending on respective value of a, this query returns user password, biometric string or smart card parameters to the adversary $\mathcal{A}$.
- *Reveal($\mathcal{P}^t$)*: The current session key SK generated by $\mathcal{P}^t$ (and its partner) is revealed to $\mathcal{A}$ through this query.
- *Test($\mathcal{P}^t$)*: Through this query $\mathcal{A}$ can send a request to $\mathcal{P}^t$ for the current session key SK and receive a *null* value, if no session key is generated. Otherwise, $\mathcal{P}^t$ can take decision according to the outcome of an unbiased flipped coin b. Basically, this query is used to measure the strength of the semantic security of the session key SK.

Definition 2: Upon receiving last expected protocol message, if $\mathcal{P}^t$ goes to an accept state, $\mathcal{P}^t$ is said to be accepted. The session identification (sid) is formed by the ordered concatenation of all communicated messages by $\mathcal{P}^t$.

Definition 3: Two instances $U_j^{t_1}$ and MAS^{t_2} are known to be partnered if the following conditions between $U_j^{t_1}$ and MAS^{t_2} are simultaneously satisfied: 1) both are in accept state, 2) both mutually authenticate each other and share the same sid, and 3) they are mutual partners of each other.

Definition 4 (Freshness): $\mathcal{P}^t$ is said to be fresh on simultaneous accomplishment of the three following conditions: 1) $\mathcal{P}^t$ is in accept state, 2) Reveal($\mathcal{P}^t$) query has never been requested to $\mathcal{P}^t$/partner of $\mathcal{P}^t$, and 3) only zero or one Corrupt($\mathcal{P}^t$,a) query has been requested to $\mathcal{P}^t$/partner of $\mathcal{P}^t$.

Definition 5 (Semantic Security): The advantage function of an adversary $\mathcal{A}$ in breaking the semantic security of the proposed fine-grained access control with user authentication scheme (FGUA) by guessing the correct bit b′ is defined by

$$Adv_{\mathcal{A}}^{FGUA} = |2.Pr[b = b'] - 1|.$$

Definition 6: A password authentication protocol with biometrics is semantically secure if the advantage function $Adv_{\mathcal{A}}^{FGUA}$ is negligibly greater than $\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}$, where q_s is the number of Send queries, $|\mathcal{D}|$ the size of password dictionary, l_b the extracted string length of user biometrics and ε_{bm} the probability of "false positive" [47].

Theorem 1: Let $\mathcal{A}$ be a polynomial time bounded adversary running within time upper bound $t_{\mathcal{A}}$. Suppose $\mathcal{A}$ makes H hash oracle queries, Send queries and Execute queries at most q_H, q_s and q_e times, respectively, in order to break the semantic security of the proposed fine-grained access control with user authentication scheme (FGUA). Then,

$$Adv_{\mathcal{A}}^{FGUA} \leq \frac{q_h^2 + 24q_h}{2^{l_h}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} + \frac{2q_s}{2^{l_n}},$$

where l_h refers to the string length of hash results, l_r is the string length of random numbers, l_n is the string length of parameter n, l_b, ε_{bm} and $|\mathcal{D}|$ are defined in Definition 6.

TABLE 2. Simulation of hash, reveal, test, corrupt and execute oracle queries.

Hash simulation query performs as follows: If the record (q, h) is found in list L_h corresponding to hash query $h(q)$, return h. Otherwise, select a string $h \in \{0,1\}^{l_h}$ and add (q, h) into L_h. If the query is initiated by $\mathcal{A}$, (q, h) is stored in $L_{\mathcal{A}}$.
Reveal($\mathcal{P}^t$) simulation query performs as follows: If $\mathcal{P}^t$ is in $accept$ state, the current session key SK formed by $\mathcal{P}^t$ and its partner is returned.
Test($\mathcal{P}^t$) simulation query performs as follows: Through *Reveal($\mathcal{P}^t$)* query, obtain current session SK and then flip a unbiased coin b. If $b = 1$, return SK. Otherwise, return a random string from $\{0,1\}^*$.
Corrupt(U_j,a) simulation query performs as follows: If $a = 1$, the query returns password PW_i of the user U_j. If $a = 2$, the query outputs biometrics B_i of U_j. If $a = 3$, the query returns the secret information stored in the user smart card SC_j.
Simulation of *Execute(U_j, MAS)* query occurs in succession with the simulation of $Send$ queries as follows: Let $H_1 = H(n_j \Vert GID_j)$ and $N_j = H(S_j \Vert H(n_j \Vert GID_j) \Vert T_{U_j} \Vert IT_i)$. U_j sends message Msg_1 to MAS, where $Msg_1 = \{N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\}$. Let $H_2 = H(RN_s \Vert PM \Vert E' \Vert T_s \Vert T_{U_j})$. MAS sends authentication message Msg_2 to U_j, where $Msg_2 = \{K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\}$. Let $H_3 = H(PM' \Vert RN_s \Vert M_j)$. U_j sends message Msg_3 to MAS, where $Msg_3 = \{H_3, M_j, E_{K_s}(PM'), RN_j\}$. Note that $\langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle \leftarrow Send(U_j, \textbf{start})$, $\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle \leftarrow Send(S, \langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle)$ and $\langle H_3, M_j, E_{K_s}(PM'), RN_j\rangle \leftarrow Send(U_j, \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle)$. Finally, $Msg_1 = \langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle$, $Msg_2 = \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle$ and $Msg_3 = \langle H_3, M_j, E_{K_s}(PM'), RN_j\rangle$ are returned.

Proof: We follow the formal security proof of this theorem as provided in [32]. We define a set of games G_i ($i = 0, 1, 2, 3, 4$) starting from the game G_0 and terminating at the game G_5. Let $Succ_i$ be an event defined as successful guessing of the bit b in *Test* query corresponding to each game G_i by the adversary $\mathcal{A}$.

Game G_0: This game and the real protocol in random oracles are assumed to be identical. Hence, we have,

$$Adv_{\mathcal{A}}^{FGUA} = |2Pr[Succ_0] - 1|. \tag{1}$$

Game G_1: All oracle queries (except *Send* query) are simulated in the game G_1. Working procedures of *Send*, *Reveal*, *Execute*, *Corrupt*, *Test* and *hash* queries are shown in Table 2. *Send* query is simulated in Table 3. We create three lists that record the outputs of different oracle queries: 1) list L_H answers hash oracle H queries, 2) list L_A stores outputs of random oracle queries, and 3) list L_T records transcripts between U_j and MAS. Due to the indistinguishability of simulation of G_1 and the real protocol execution of G_0, we obtain

$$Pr[Succ_1] = Pr[Succ_0]. \tag{2}$$

Game G_2: This game considers the collision situations with hash results and random numbers in the transcripts of all communicated messages in the login and authentication phases of our scheme. Following the birthday paradox, the collision probability of H hash oracle query is at most $\frac{q_h^2}{2^{l_h+1}}$. As authentication messages $Msg_2 = \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s|| PM|| E'|| T_s|| T_{U_j}), E_j\rangle$ and $Msg_3 = \langle H(PM' ||RN_s ||M_j), M_j, E_{K_s}(PM') ||RN_j\rangle$ contain random numbers RN_s and RN_j, respectively, the probability of random numbers collision is at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. So, we have,

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}}. \tag{3}$$

Game G_3: This game considers a situation where $\mathcal{A}$ obtains the correct message transcript luckily without active participation of hash oracles H. As the login and authorization phases of our scheme involve three messages Msg_1, Msg_2 and Msg_3 communications, we consider following three cases in G_3:

Case 1: In this case, we consider $Send(MAS, Msg_1)$ query and try to respond it. Hence, the hash value $N_j = H(S_j ||H(n_j ||GID_j) ||T_{U_j} ||IT_i) \in L_{\mathcal{A}}$ and $H(n_j ||GID_j) \in L_{\mathcal{A}}$ must hold; otherwise, the session will be terminated. The maximum calculated probability is up to $\frac{2q_h}{2^{l_h}}$. After successful verification, the MAS should output $(MB_j ||S_{ID}, *)$ to recover S_j with probability $\frac{q_h}{2^{l_h}}$. Again, as user password PW_j is not known to the MAS, it can not reveal the values of records $(MB_j||PW_j, *)$, $(MB_j||ID_{U_j}||r_j, *)$, $(A_{ID_j}||TS'_{U_j}, *)$ and $(W'_j||A'_j||GID_j, *)$, and the calculated probability is at most $\frac{4q_h}{2^{l_h}}$. Finally, to continue with the current session, the message $Msg_1 \in L_T$ should hold with string length n. For this, the probability is $\frac{q_s}{2^{l_n}}$.

Case 2: In this case, we consider the first authentication message Msg_2 sent by the MAS. To respond $Send(U_j, Msg_2)$ oracle query, $K_j = (X_j \oplus S'_j) \oplus H(H(n_j ||GID_j) ||RN_s) \in L_{\mathcal{A}}$ and $H(A_j|| PM|| E'|| T_s|| T_{U_j}) \in L_{\mathcal{A}}$ must hold with the total maximum probability $\frac{2q_h}{2^{l_h}}$. Further, as the MAS should check the value of N_j, so the record $H(S_j ||H(n_j ||GID_j) ||T_{U_j} ||IT_i) \in L_{\mathcal{A}}$ must be true with probability $\frac{q_h}{2^{l_h}}$. Finally, for a transcript message with random number RN_s, $Msg_2 \in L_T$ and we get the maximum probability as $\frac{q_s}{2^{l_r}}$.

TABLE 3. Simulation of send oracle queries.

Send simulation query performs as follows.

(a) For a $Send(U_j, \textbf{start})$ query, U_j gives the following response:
Compute $S_j = X_j \oplus H(\alpha_j||S_{ID})$, $N_j = H(S_j||H(n_j||GID_j)||T_{U_j}||IT_i)$ as in Figure 2.
Output $Msg_1 = \langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle$.

(b) For a $Send(S_j, \langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle)$ query, U_j gives the following response:
Verify whether $|T_{U_j} - T^*_{U_j}| \leq \triangle T$ and compute S'_j and $H(n_j||GID_j)$ as in Figure 3.
Check if the computed hash value is same as the received hash value N_j.
A mismatch rejects the session. Otherwise, check if received $IT_i \in GID_j$.
Generate session key K_s and random number α.
Further, the server MAS computes E_j, E', $E_{K_s}(PM)$ and hash value H_2 as given in Figure 3 and Table II.
Output $Msg_2 = \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle$.

(c) U_j answers $Send(U_j, \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle)$ query as follows.
Verify whether $|T_s - T^*_s| \leq \triangle T$ and then compute K'_j.
Check if $K'_j = K_j$ as given in Figure 3. A mismatch leads to termination of the session.
Otherwise, obtain K_s, PM^*, PM', H_2 and verify computed and received hash values as given in Figure 3 and Table II.
The MAS establishes the session key $SK_{U_j,S}$. Output $Msg_3 = \langle H_3, M_j, E_{K_s}(PM'), RN_j\rangle$.

(d) For a $Send(U_j, \langle H_3, M_j, E_{K_s}(PM'), RN_j\rangle)$ query, the MAS gives the following response:
Decrypt puzzle message PM as $PM' = D_{K_s}(E_{K_s}(PM'))$ and compute and verify PM''.
If verification holds successfully, establish SK_{S,U_j} as the session key as given Figure 3.

Finally, both U_j and MAS accept the successful termination of the session.

Case 3: In this case, we consider the second authentication message Msg_3 sent by U_j in reply to Msg_2. To respond $Send(MAS, Msg_3)$, the hash values $H(PM' \,||RN_s \,||M_j) \in L_{\mathcal{A}}$ and $H(RN_s \,||PM^*||\, E'||\, T_s||\, T_{U_j}) \in L_{\mathcal{A}}$ must hold; otherwise, the session will be terminated. The maximum calculated probability is up to $\frac{2q_h}{2^{l_h}}$. Finally, for a transcript message with random number RN_j, $Msg_3 \in L_T$, we get the maximum probability as $\frac{q_s}{2^{l_r}}$.

Considering all the above three cases, we have,

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{q_s}{2^{l_n}} + \frac{12q_h}{2^{l_h}}. \tag{4}$$

Game G_4: This game considers all online and offline attacks executed by the adversary $\mathcal{A}$. As our scheme provides three-factor authentication security, we need to consider guessing of both password and biometrics.

Case 1: To start the queries along with password PW_j and biometrics B_j, $\mathcal{A}$ requires all information stored in smart card of U_j. For this purpose, $\mathcal{A}$ executes $Corrupt(U_j, 3)$, which is composed of the following two sub-cases:

Case 1.1: For online password guessing, $\mathcal{A}$ runs query $Corrupt(U_j, 1)$. Here, $\mathcal{A}$ selects a password on-the-fly from dictionary $\mathcal{D}$ and then runs at most q_s times *Send(MAS,Msg$_1$)* query. The probability of this case is $\frac{q_s}{|\mathcal{D}|}$.

Case 1.2: It deals with passing of biometrics checking by $\mathcal{A}$ through query $Corrupt(U_j, 2)$. For each guessing, the probability is at most $\frac{1}{2^{l_b}}$, where l_b is the length of extracted secret biometric string. Moreover, we should consider the possible accidental guessing of "false positive" case with probability ε_{bm}. In general, it is observed that for fingerprints, $\varepsilon_{bm} \approx 2^{-14}$ [47]. As a whole, the guessing probability under this case is at most $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$.

It is obvious that the simulation of the games G_3 and G_4 are not distinguishable without execution of the above mentioned guessing attacks. So, we have,

$$|Pr[Succ_4] - Pr[Succ_3]| \leq \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm}\}.$$

Considering all above games, since $\mathcal{A}$ gains no advantage to guess the correct bit b, we get,

$$Pr[Succ_4] = \frac{1}{2}. \tag{5}$$

Using the triangular inequality, we have,

$$\begin{aligned} |Pr[Succ_0] - \frac{1}{2}| &= |Pr[Succ_1] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_3]| \\ &\quad + |Pr[Succ_3] - Pr[Succ_4]|. \end{aligned} \tag{6}$$

Using Equations (1)-(6), we obtain,

$$\begin{aligned} \frac{1}{2}Adv_{\mathcal{A}}^{FGUA} &= |Pr[Succ_0] - \frac{1}{2}| \\ &\leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}} \\ &\quad + \frac{2q_s}{2^{l_r}} + \frac{2q_s}{2^{l_n}} + \frac{12q_h}{2^{l_h}} \\ &\quad + \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned} \tag{7}$$

Finally, multiplying both sides by 2 in Equation (7) and rearranging the terms, we obtain the required result. Hence, the theorem is proved. □

B. AUTHENTICATION PROOF BASED ON BAN-LOGIC

The BAN logic is used in analyzing the security of authentication schemes [19], [48] in order to prove secure mutual authentication between communicating parties in a network. In this section, we provide authentication proof using the BAN logic and then demonstrate how the proposed scheme achieves mutual authentication between a user U_j and the medical server MAS.

The notations used in BAN logic analysis are defined as follows.

- $P \mid\equiv X$: Principal P believes statement X.
- $P \triangleleft X$: P sees the statement X.
- $\#(X)$: The formula X is fresh.
- $P \mid\sim X$: Principal P once said statement X.
- (X, Y): Formula X or formula Y is one part of the formula (X, Y).
- $P \Rightarrow X$: P has jurisdiction over statement X.
- $\langle X\rangle_Y$: This represents X combined with the formula Y.
- $P \overset{K}{\longleftrightarrow} Q$: P and Q may use the shared key K to communicate. K is good in that it will be known only by P and Q.
- $P \overset{X}{\rightleftharpoons} Q$: Formula X is a secret known only to P and to Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another.
- SK: Session key used in the current session.

The rules given below describe the main logical postulates of the BAN logic [48], [49]:

- **Rule 1.** (Message-meaning)
$$\frac{P \mid\equiv Q \overset{K}{\rightleftharpoons} P, P \triangleleft \langle X\rangle_K}{P \mid\equiv Q \mid\sim X}.$$
- **Rule 2.** (Nonce-verification)
$$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}.$$
- **Rule 3.** (Freshness-conjuncatenation)
$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}.$$
- **Rule 4.** (Jurisdiction)
$$\frac{P \mid\equiv Q \Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}.$$
- **Rule 5.** (Other inference)
$$\frac{P \mid\equiv (X, Y)}{P \mid\equiv X}, \quad \frac{P \triangleleft (X, Y)}{P \triangleleft X},$$
$$\frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X}, \quad \frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X}.$$

According to the analytic procedures of the BAN logic, the proposed protocol will satisfy the following goals:

- **Goal 1.** $U_j \mid\equiv (U_j \overset{SK}{\longleftrightarrow} MAS)$.
- **Goal 2.** $S \mid\equiv (U_j \overset{SK}{\longleftrightarrow} MAS)$.

The generic types of our proposed protocol are given below:

Message 1. $U_j \rightarrow MAS$: $(H(S_j \,||H(n_j \,||GID_j) \,||T_{U_j} \,||IT_i) \,||IT_i \,||T_{U_j} \,||(TS'_{U_j} \oplus H(n_j \,||GID_j)))$.

Message 2. $MAS \rightarrow U_j$: $(X_j \oplus S'_j) \oplus H(H(n_j \,||GID_j) \,||RN_s) \,||RN_s \,||IT_i \,||E' \,||E_{K_s}(PM) \,||T_s \,||H(RN_s \,||PM \,||E' \,||T_s \,||T_{U_j} \,||K_j) \,||E_j)$.

Message 3. $U_j \rightarrow MAS$: $(H(PM' \,||RN_s \,||M_j) \,||M_j \,||E_{K_s}(PM') \,||RN_j \,||T^1_{U_j})$.

The idealized form of the proposed protocol are given below.

Message 1. $U_j \rightarrow MAS$: $(\langle R_j, T_{U_j}, IT_i)\rangle_{S_j}, IT_i, T_{U_j}, \langle R_j\rangle_{TS_{U_j}})$.

Message 2. $U_j \rightarrow MAS$: $(\langle X_j, R_j, RN_s\rangle_{S_j}, RN_s, IT_i, E', \{PM\}_{K_s}, \langle RN_s, PM, E', T_s, T_{U_j}, K_j\rangle_{S_j}, E_j)$.

Message 3. $U_j \rightarrow MAS$: $(\langle RN_s, M_j, RN_j, T_i\rangle_{PM}, M_j, \langle PM, RN_j, T_i, T^1_{U_j}\rangle_{K_s}, RN_j)$.

Regarding the initial state of the scheme, we make the following basic assumptions to further analyze the proposed scheme.

- **A.1:** $U_j \mid\equiv \#(T_s)$;
- **A.2 (a):** $MAS \mid\equiv \#(T_{U_j})$; **A.2 (b):** $MAS \mid\equiv \#(T^1_{U_j})$;
- **A.3:** $U_j \mid\equiv MAS \Rightarrow (T_s, RN_s, K_s, PM)$;
- **A.4:** $MAS \mid\equiv U_j \Rightarrow (TS_{U_j}, RN_j, A_j, TU_j, T^1_{U_j})$;
- **A.5:** $U_j \mid\equiv (TS_{U_j}, RN_j, A_j, K_s, TU_j, T^1_{U_j})$;
- **A.6:** $MAS \mid\equiv (T_s, RN_s, K_s, PM, TS_{U_j}, A_j)$;
- **A.7:** $U_j \mid\equiv (U_j \overset{S_j}{\rightleftharpoons} MAS)$;
- **A.8:** $MAS \mid\equiv (U_j \overset{S_j}{\rightleftharpoons} MAS)$;
- **A.9:** $U_j \mid\equiv (U_j \overset{K_s}{\rightleftharpoons} MAS)$;
- **A.10:** $MAS \mid\equiv (U_j \overset{K_s}{\rightleftharpoons} MAS)$.

Based on the above-mentioned assumptions and the logical postulates of the BAN logic, we analyze the idealized form of the proposed scheme, and provide the main procedures of proof as follows.

The MAS receives one login message (Msg_1) and one authentication message (Msg_3) from U_j. Both these messages contribute to achieve Goal 2. According to the Msg_1, we obtain the following:

- S_1: $MAS \triangleleft (\langle R_j, T_{U_j}, IT_i\rangle_{S_j}, IT_i, T_{U_j}, \langle R_j\rangle_{TS_{U_j}})$.
- S_2: According to the inference rule (Rule 5), we obtain $MAS \triangleleft \langle R_j, T_{U_j}, IT_i\rangle_{S_j}$.
- S_3: According to A.8 and Rule 1, we obtain $MAS \mid\equiv U_j \mid\sim (R_j, T_{U_j}, IT_i)$.
- S_4: According to A.2(a) and Rule 3, we obtain $MAS \mid\equiv \#(R_j, T_{U_j}, IT_i)$.
- S_5: According to S_3, S_4 and Rule 2, we obtain $MAS \mid\equiv U_j \mid\equiv (R_j, T_{U_j}, IT_i)$.
- S_6: According to A.4 and Rule 4, we obtain $MAS \mid\equiv (R_j, T_{U_j}, IT_i)$.
- S_7: According to S_6 and Rule 5, we obtain $S_j \mid\equiv T_{U_j}$.

According to Msg_3, we obtain the following:

- S_8: $MAS \lhd (\langle RN_s, M_j, RN_j, T_i\rangle_{PM}, M_j, \langle PM, RN_j, T_i, T^1_{U_j}\rangle_{K_s}, RN_j)$.
- S_9: According to the inference rule (Rule 5), we obtain $MAS \lhd \langle PM, RN_j, T_i, T^1_{U_j}\rangle_{K_s}$.
- S_{10}: According to A.10 and Rule 1, we obtain $MAS \mid\equiv U_j \mid\sim (PM, RN_j, T_i)$.
- S_{11}: According to A.2(b) and Rule 3, we obtain $MAS \mid\equiv \#(PM, RN_j, T_i)$.
- S_{12}: According to S_{10}, S_{11} and Rule 2, we obtain $MAS \mid\equiv U_j \mid\equiv (PM, RN_j, T_i)$.
- S_{13}: According to A.4 and Rule 4, we obtain $MAS \mid\equiv (PM, RN_j, T_i)$.
- S_{14}: According to S_{13} and Rule 5, we obtain $MAS \mid\equiv RN_j$.
- S_{15}: According to $A.6$, we get $MAS \mid\equiv T_s, MAS \mid\equiv RN_s$, $MAS \mid\equiv K_s$, $MAS \mid\equiv PM$ and $MAS \mid\equiv A_j$.
- S_{16}: According to the proposed scheme, $SK = H(PM|| RN_j|| RN_s ||K_s ||T_{U_j} ||T_s ||A_j)$. So, according to the results of S_7, S_{14} and S_{15}, we obtain $MAS \mid\equiv (U_j \overset{SK}{\longleftrightarrow} MAS)$. **(Goal 2)**

According to Msg_2, we obtain the following:

- S_{17}: $U_j \lhd (\langle X_j, R_j, RN_s\rangle_{S_j}, RN_s, IT_i, E', \langle PM\rangle_{K_s}, \langle RN_s, PM, E', T_s, T_{U_j}\rangle\rangle_{S_j}, E_j)$.
- S_{18}: According to Rule 5, we obtain $U_j \lhd \langle RN_s, PM, E', T_s, T_{U_j}, K_j\rangle_{S_j}$.
- S_{19}: According to A.7 and Rule 1, we obtain $U_j \mid\equiv MAS \mid\sim (RN_s, PM, E', T_s, T_{U_j})$.
- S_{20}: According to A.1 and Rule 3, we obtain $U_j \mid\equiv \#(RN_s, PM, E', T_s, T_{U_j})$.
- S_{21}: According to S_{19}, S_{20} and Rule 2, we obtain $U_j \mid\equiv MAS \mid\equiv (RN_s, PM, E', T_s, T_{U_j})$.
- S_{22}: According to A.3 and Rule 4, we obtain $U_j \mid\equiv (RN_s, PM, E', T_s, T_{U_j})$.
- S_{23}: According to S_{22} and Rule 5, we obtain $U_j \mid\equiv RN_s$, $U_j \mid\equiv PM$ and $U_j \mid\equiv T_s$.
- S_{24}: According to A.5 and Rule 5, we get $U_j \mid\equiv RN_j$, $U_j \mid\equiv A_j$, $U_j \mid\equiv T_{U_j}$ and $U_j \mid\equiv K_s$.
- S_{25}: According to the proposed scheme, $SK = H(PM ||RN_j ||RN_s ||K_s ||T_{U_j} ||T_s ||A_j)$.
  Finally, according to S_{23} and S_{24}, we obtain $U_j \mid\equiv (U_j \overset{SK}{\longleftrightarrow} MAS)$. **(Goal 1)**

From the Goals 1 and 2, it is clear that the secure mutual authentication between U_j and MAS is achieved.

C. DISCUSSION ON OTHER ATTACKS

In this section, through the informal security analysis we show that our scheme is also secure against the following known attacks.

1) STOLEN SMART CARD ATTACK

Suppose the user U_j's smart card SC_j with id SC_{id_j} is lost or stolen. By monitoring the power consumption [2], [3], an attacker $\mathcal{A}$ can extract all the stored information from SC_j, which include $A = (A_{ID_j} \oplus SC_{id_j})$, GID_j, $B = (TS_{U_j} \oplus SC_{id_j})$, $C = (r_j \oplus MB_j)$, P_j, SK_j, R_{U_j}, $e : G_1 \times G_1 \rightarrow G_T$, X_j, $H(\cdot)$, β_j, $Gen(\cdot)$, $Rep(\cdot)$ and τ. It is to be noted that the user identity ID_{U_j}, password PW_j and biometric B_j are not directly stored in SC_j. To retrieve them, $\mathcal{A}$ need to know ID_{U_j}, PW_j and B_j from stored $(A_{ID_j} \oplus SC_{id_j})$ and R_{U_j}. From $A_{ID_j} = H(\alpha_j|| ID_{U_j} ||r_j)$, $\mathcal{A}$ has no feasible way to know the user's id ID_{U_j} or biometric B_j. Due to one-way property of the hash function $H(\cdot)$, it is considered to be a computationally infeasible problem. In addition, user id ID_{U_j}, biometric B_j and password PW_j can not be retrieved from $R_{U_j} = H(W_j ||A_j ||GID_j) = H(H(\alpha_j ||PW_j) ||H(A_{ID_j} ||TS_{U_j}) ||GID_j)$ due to the one-way property of $H(\cdot)$. Moreover, $\mathcal{A}$ has no feasible way to obtain user id, password or biometric even if the brute force search is applied, because he/she has to guess ID_{U_j}, B_j and PW_j simultaneously. As a result, our scheme prevents stolen smart card attack or smart card breach attack.

2) REPLAY ATTACK

Replay attack is considered to be one of the most common attacks in any security protocol. Suppose in the login phase, an attacker $\mathcal{A}$ intercepts and replays the transmitted message $\langle N_j, IT_i, T_{U_j}, (TS_{U_j} \oplus H(n_j ||GID_j))\rangle$, where $N_j = H(S_j|| H(n_j ||GID_j) ||T_{U_j} ||IT_i)$. The MAS discards the message if $|T_{U_j} - T^*_{U_j}| > \triangle T$, where $T^*_{U_j}$ is the timestamp when the MAS receives this message and $\triangle T$ is the maximum transmission delay. In the authorization phase, the MAS sends the message $\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s|| PM ||E' ||T_s ||T_{U_j})$, $E_j\rangle$ to U_j, where $K_j = (X_j \oplus S'_j) \oplus H(H(n_j ||GID_j) ||RN_s)$ and RN_s is a server generated random nonce selected for each session. Use of the server timestamp T_s, if this message is replayed by $\mathcal{A}$ to the MAS, the timestamp validation of T_s will fail, and the message will be discarded by the MAS too. Thus, the proposed scheme protects the replay attack.

3) PRIVILEGED INSIDER ATTACK

Using the privileged insider attack, a genuine privileged user, say U_m of the MAS may turn out to be a malicious user, and also may try to achieve password of other legal user U_j. However, according to our scheme, U_j does not submit the original password PW_j in the MAS. Rather, he/she stores $\langle A_{ID_j}, W_j\rangle$, where $A_{ID_j} = H(\alpha_j ||ID_{U_j} ||r_j)$ and $W_j = H(\alpha_j ||PW_j)$. Any privileged insider U_m can not obtain user's id ID_{U_j}, password PW_j or biometric B_j from A_{ID_j} or W_j as it is computationally infeasible due to one-way property of $H(\cdot)$. Therefore, a malicious insider U_m cannot obtain the user secret credentials, and the proposed scheme has the ability to defend the privileged insider attack.

4) MAN-IN-THE-MIDDLE ATTACK

Through the man-in-the-middle attack, an adversary $\mathcal{A}$ may try to modify the intercepted login or authorization messages. Suppose an adversary $\mathcal{A}$ intercepts the login and authorization messages $Msg_1 = \langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j|| GID_j))\rangle$, $Msg_2 = \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s|| PM|| E'|| T_s|| T_{U_j}), E_j\rangle$ and $Msg_3 = \langle H(PM' ||RN_s ||M_j), M_j, E_{K_s}(PM') ||RN_j\rangle$, and tries to modify these messages.

To modify the message Msg_1, $\mathcal{A}$ needs to modify the parameters IT_i, T_{U_j}, $(TS_{U_j} \oplus H(n_j||GID_j))$. Use of the server timestamp T_s, random nonce RN_s and the hash value $H(RN_s|| PM|| E'|| T_s|| T_{U_j})$ prevents any possibility of modification of any parameter in the message Msg_2. In a similar way, to modify the message Msg_3, $\mathcal{A}$ needs PM and K_s. Due to the one-way property of $H(\cdot)$ and symmetric encryption/decryption, it is quite difficult task for $\mathcal{A}$ to modify the messages Msg_1, Msg_2 and Msg_3 to convert to legal valid messages. Hence, the proposed scheme is free from the man-in-the-middle attack.

5) OFFLINE AND ONLINE PASSWORD GUESSING ATTACKS

Executing the power analysis attacks [2], [3], an attacker $\mathcal{A}$ can extract all the stored information from a lost or stolen smart card SC_j of U_j. To obtain the user identity ID_{U_j}, password PW_j and biometric key α_j, $\mathcal{A}$ has to guess α_j and ID_{U_j} simultaneously from $A_{ID_j} = H(\alpha_j ||ID_{U_j} ||r_j)$. Similarly, to obtain the password PW_j, the attacker need to guess PW_j, α_j and ID_{U_j} simultaneously from $R_{U_j} = H(W_j|| A_j|| GID_j) = H(H(\alpha_j ||PW_j) ||H(A_{ID_j} ||TS_{U_j}) ||GID_j)$. Due to the one-way property of $H(\cdot)$, correct guessing of password from the parameter R_{U_j} is a computationally infeasible problem. So, our scheme can resist offline password guessing attack. Furthermore, by eavesdropping or intercepting the messages Msg_1, Msg_2 and Msg_3, $\mathcal{A}$ cannot guess or obtain the password PW_j, biometric key α_j or identity ID_{U_j} of the user U_j. Thus, the proposed scheme can also resist online password guessing attack.

6) USER IMPERSONATION ATTACK

Using the user impersonation attack, an adversary or a malicious user $\mathcal{A}$ can try to masquerade as a legitimate user and try to login to the server MAS. However, our proposed scheme can resist this attack due to the following arguments:

- $\mathcal{A}$ needs to input a correct value of password PW_j, biometric B_j or identity ID_{U_j} to prove its authenticity to the smart card system as a genuine user. However, we have analyzed that $\mathcal{A}$ has no feasible way to guess these parameters.
- $\mathcal{A}$ can try to generate a replay login message $Msg_1 = \langle N_j, IT_i, T_{U_j}, (TS_{U_j}\oplus H(n_j|| GID_j))\rangle$ and submit it to the MAS. However, a duplicate value of the random nonce n_j and validity of timestamp will reveal that the message is a replayed one and not an original message. To modify the parameters in Msg_1, $\mathcal{A}$ needs to change the value of N_j, where $N_j = H(S_j|| H(n_j|| GID_j)|| T_{U_j}|| IT_i)$ and $S_j = X_j\oplus H(\alpha_j ||S_{ID})$. As $\mathcal{A}$ does not know the correct value of B_j and ID_{U_j}, he/she cannot modify N_j correctly.

Hence, the proposed scheme is able to resist user impersonation attack.

7) SERVER IMPERSONATION ATTACK

An adversary $\mathcal{A}$ can masquerade as a server and try to respond with valid message to the user U_j. As already mentioned above, $\mathcal{A}$ cannot successfully replay and/or modify the authorization messages Msg_2 and Msg_3 due to usage of one-way hash function $H(\cdot)$ and secret parameters. So, the proposed scheme also resists server impersonation attack.

8) DENIAL-OF-SERVICE (DoS) ATTACK

In our scheme, during the login phase, U_j sends the message Msg_1 to the MAS that includes the registration timestamp of U_j. At the time of authorization, the MAS checks the authenticity of this message, and sends an encrypted key and an encrypted puzzle message to U_j. This message includes the user current timestamp T_{U_j} and random nonce RN_s. U_j checks the authenticity of this message, obtains the key and decrypts the puzzle message. U_j then sends a data request M_j to the MAS including the server timestamps RN_s and user random nonce RN_j. Finally, after successful authentication and verification, the MAS replies this data request encrypted with the session key and MAS sends an acknowledgment to U_j. If an attacker blocks the messages from reaching the MAS and U_j, both of them will know about malicious dropping of such control messages. Furthermore, any wrong input in ID_{U_j}, PW_j and B_j does not allow the authentication verification successfully by the smart card SC_j locally. Thus, the proposed scheme has the ability to resist the DoS attack.

9) KNOWN SESSION KEY SECRECY

The proposed scheme is protected against a compromised session key due to the following reason. Suppose the session key $SK_{U_j,S}$ $(= SK_{S,U_j}) = H(PM|| RN_j|| RN_i|| K_s|| T_{U_j}|| T_i|| A_j)$ is compromised by an adversary $\mathcal{A}$. The session key is a hashed output of the parameters that includes the ephemeral secrets A_j and K_s as well as the temporal values RN_j, RN_s, T_{U_j} and T_s. Due to the use of timestamps and random nonces, A_j and K_s, the session key $SK_{U_j,S}$ is unique for each session. Hence, compromise of a particular session key does not affect other session keys, and as a result, the proposed scheme provides the known session key secrecy property.

10) PARALLEL SESSION AND REFLECTION ATTACKS

As already discussed above, from any of the eavesdropped messages Msg_1, Msg_2 and Msg_3, an attacker $\mathcal{A}$ can neither obtain the correct password PW_j nor the biometrics key α_j of a legal user U_j. Hence, from any eavesdropped messages, $\mathcal{A}$ can not create a valid login request message, and thus, he/she can not start a new session with the MAS by masquerading as a legal user. Thus, our scheme protects the parallel session and reflection attacks.

VI. FUNCTIONALITY ANALYSIS

In this section, we show that the following functional requirements are fulfilled by the proposed scheme.

A. FINE-GRAINED ACCESS CONTROL

Only authentication is not sufficient to provide access permission to the user U_j in TMIS. The proposed scheme is designed in such a way that after successful authentication, U_j can access only those information for which he/she has access permission. We have used the Key-Policy Attribute-Based Encryption (KP-ABE) [22] in order to achieve the fine-grained access control with full granularity for accessing right data by a right user. In the proposed scheme, the secret session key $SK_{S,U_j}(= SK_{U_j,S})$ is generated between an authentic user U_j and the MAS to encrypt future messages for a particular session. The current session key can be formed by U_j if the user's access structure P_j “accepts" $\mathcal{I}_i$. This means that all the attributes specified for the information type IT_i need to match with the user access structure and then only that U_j will get $e(g, g)^{\alpha qr(0)} = e(g, g)^{\alpha y}$ to finally obtain Y^{α}. Using the value of Y^{α}, the user U_j further computes K_s.

B. USER ANONYMITY

During the login phase, U_j sends the message $Msg_1 = \langle N_j, IT_i, T_{U_j}, (TS_{U_j} \oplus H(n_j|| GID_j))\rangle$ to the MAS. Suppose an attacker $\mathcal{A}$ eavesdrops this login request message. As this message does not contain the user id ID_{U_j}, $\mathcal{A}$ can not obtain the user id by eavesdropping this message. Moreover, the original identity of U_j is not delivered to the MAS. Instead, the MAS receives the unique anonymous parameter N_j from U_j. So, even if a server spoofing attack is executed, the original identity of U_j is not revealed to $\mathcal{A}$. This shows that the proposed scheme preserves the user anonymity property.

C. MUTUAL AUTHENTICATION

In authorization phase of our scheme, both U_j and the MAS verify the authenticity of one another through mutual authentication. The MAS sends message $\langle K_j, RN_s, IT_i, E', T_s, E_{K_s}(PM), H(RN_s|| PM|| E'|| T_s|| T_{U_j}), E_j\rangle$ to U_j. After receiving this, U_j verifies whether $K_j' = H(W_j|| S_{ID_j}) \oplus H(H(n_j|| GID_j) ||RN_s)$ holds or not. An unsuccessful verification leads to termination of the phase immediately. Further, using the access tree, U_j obtains the key K_s. Using K_s, U_j decrypts the encrypted puzzle message $E_{K_s}(PM)$. After getting PM, U_j checks the authenticity of the message by matching computed $H(RN_s ||PM ||E' ||T_i ||T_{U_j})$ with the received hash value. If this verification fails, U_j stops the phase terminates immediately. In addition, U_j sends the message $\langle H(PM' ||RN_s ||M_j), M_j, E_{K_s}(PM'), RN_j\rangle$ to the MAS, who decrypts the encrypted puzzle $E_{K_s}(PM')$ and gets the puzzle PM'. Also, the MAS verifies computed $PM'' = H(PM|| RN_j|| T_i)$ with its own PM, T_i and the received RN_j. If this verification fails, the MAS terminates the phase immediately. Due to this mutual verification from both U_j and the MAS, they can correctly verify the authenticity of one another.

D. SECURE SESSION KEY ESTABLISHMENT

In the authorization phase, both U_j and MAS individually establish the same session key $SK_{U_j,S}$ and SK_{S,U_j} for future communication. Here, $SK_{U_j,S}$ $(= SK_{S,U_j}) = H(PM ||RN_j ||RN_s ||K_s ||T_{U_j} ||T_s ||A_j)$. Before establishing this session key, both U_j and MAS mutually authenticate each other. This guarantees that the communicated parameters and messages are resistant to replay attack, man-in-the-middle attack and impersonation attacks. Hence, the established session key is secure against different attacks.

TABLE 4. Security comparison with existing authentication schemes for TMIS.

	Awasthi-Srivastava [63]	Jiang *et al.* [23]	Mishra *et al.* [24]	Our
SF_1	X	✓	✓	✓
SF_2	X	✓	✓	✓
SF_3	✓	✓	✓	✓
SF_4	X	✓	✓	✓
SF_5	✓	✓	✓	✓
SF_6	✓	✓	✓	✓
SF_7	X	✓	✓	✓
SF_8	✓	✓	✓	✓
SF_9	X	X	✓	✓
SF_{10}	X	X	✓	✓

Note: SF_1: stolen smart card attack; SF_2: off-line password guessing attack; SF_3: on-line password guessing attack; SF_4: strong replay attack; SF_5: man-in-the-middle attack; SF_6: privileged insider attack; SF_7: user impersonation attack; SF_8: server impersonation attack; SF_9: denial-of-service attack; SF_{10}: known session key secrecy.
X: insecure against a particular attack; ✓: secure against a particular attack.

TABLE 5. Functionality comparison with existing authentication schemes for TMIS.

	Awasthi-Srivastava [63]	Jiang *et al.* [23]	Mishra *et al.* [24]	Our
FN_1	X	X	X	✓
FN_2	X	X	X	✓
FN_3	X	✓	✓	✓
FN_4	X	✓	✓	✓
FN_5	X	✓	✓	✓
FN_6	X	X	✓	✓

Note: FN_1: attribute based access control; FN_2: group based access control; FN_3: user anonymity provision; FN_4: forward secrecy; FN_5: secret session key establishment; FN_6: efficient password change.
X: does not support a particular feature; ✓: supports a particular feature.

TABLE 6. Execution timings of various cryptographic operations.

Term	Description	Time (in seconds)
T_h	One-way cryptographic hash function	0.00050
T_{Ch}	Chebyshev map operation	0.02102
T_{hc}	One way chaotic-hash operation	0.02102
T_{enc}/T_{dec}	Symmetric key encryption/decryption	0.00870
T_m	Elliptic curve point multiplication	0.06308
T_{fe}	Fuzzy extractor operation	$\approx T_m$

E. EFFICIENT PASSWORD CHANGE

In the password change phase of our proposed scheme, a legal user U_j alone can change his/her password without involvement of the MAS. This phase is designed in a way such that if U_j enters wrong old password, the phase terminates immediately. U_j can not set the new password if he/she enters a wrong old password by mistake or unknowingly. Doing so, it resists a possible denial-of-service from the system. Furthermore, the smart card SC_j of U_j never stores the modified password directly. It is then free from stolen smart card attack and password guessing attack too. If U_j gives a correct old password, SC_j of U_j computes new masked password $W^{new}{}_j = H(\alpha_j|| PW_j^{new})$ and $R^{new}{}_{U_j} = H(W^{new}{}_j|| A_j'|| GID_j)$. Finally, SC_j replaces old $R^{old}{}_{U_j}$ with the new masked password $R^{new}{}_{U_j}$, and stores it into the memory of the smart card. Thus, the password change phase entirely takes place locally without contacting the MAS by a leagl user U_j only. Hence, the proposed scheme supports efficient password change phase.

TABLE 7. Computational cost comparison.

Phase /Scheme	Entity	Awasthi-Srivastava [63]	Jiang *et al.* [23]	Mishra *et al.* [24]	Our
Login	User side	$3T_{hc}$	$T_{enc} + 2T_{Ch}$	$4T_h + 2T_{Ch}$	$7T_h + T_{fe}$
	Server side	–	–	–	–
Authorization/ Authentication	User side	T_{hc}	$T_h + T_{Ch}$	T_h	$6T_h + 2T_{enc/dec}$
	Server side	$3T_{hc}$	$T_h + 2T_{dec} + 3T_{Ch}$	$5T_h + T_{Ch}$	$7T_h + 2T_{enc/dec}$
Total cost		$7T_{hc}$	$2T_h + 6T_{Ch} + 3T_{enc}/T_{dec}$	$10T_h + 3T_{Ch}$	$20T_h +\ 4T_{enc}/T_{dec} + T_{fe}$
Execution time (in milliseconds)		147.14	153.22	68.06	107.88

F. FORWARD SECRECY

Forward secrecy ensures that a session key which is derived from a set of long-term keys as well as temporal information cannot be compromised, if one of the long-term keys is compromised in future. According to the proposed scheme, if user's long-term key K_s is compromised, an adversary $\mathcal{A}$ can try to compute the session key $SK_{S,U_j} = H(PM|| RN_j|| RN_s|| K_s|| T_{U_j} ||T_s ||A_j)$. However, $\mathcal{A}$ still requires to compute the long-term secret A_j. Since $A_j = H(A_{ID_j} ||TS_{U_j})$ and $A_{ID_j} = H(\alpha_j ||ID_{U_j} ||r_j)$, $\mathcal{A}$ can not compute A_{ID_j} without knowing the biometric key α_j and ID_{U_j} simultaneously. Also, computing A_{ID_j} from $A_j = H(A_{ID_j} ||TS_{U_j})$ is computationally infeasible task by the adversary $\mathcal{A}$. This shows that the proposed scheme achieves forward secrecy property.

VII. SECURITY, FUNCTIONALITY AND PERFORMANCE COMPARISON

In this section, we perform the security, functionality and performance comparisons among some related authentication schemes proposed in TMIS. The results show that in-spite of providing unique access privilege through the attribute based access control and group based access control, the proposed scheme can resist several well-known attacks.

A. SECURITY COMPARISON

We compare security of the proposed scheme with existing related authentication schemes for TMIS [12], [13], [50]. A detailed comparison on different security attacks are tabulated in Table 4. It is clear from this table that our proposed scheme overcomes most of the security weaknesses of the existing related schemes.

B. FUNCTIONALITY COMPARISON

In Table 5, we compare different functionalities of our proposed scheme with existing related authentication schemes for TMIS. For comparison, we have considered the same schemes as mentioned in the previous section. A study of the tabulated result shows that none of the existing schemes provide fine-grained access control and group based user access control in TMIS.

C. PERFORMANCE COMPARISON

In this section, we compare the comparisons of comuputational costs, communication costs, security features and functionality features aming the proposed scheme and other related schemes, such as the schemes of Jiang *et al.* [12], Mishra *et al.* [13] and Awasthi and Srivastava [50].

Table 6 shows the execution times for various cryptographic operations which are required for analysis of computational cost measurement for our proposed scheme and other schemes. The results shown in Table 6 are based on an existing experiment conducted on an Intel Pentium IV 2600 MHz processor with 1024 MB RAM [51]. We ignore the computation cost of bitwise XOR operation as it is significantly low as compared to other operations. We further assume that $T_{hc} \approx T_{Ch}$. In addition, $T_{fe} \approx T_m = 0.06308$ seconds [52].

In Table 7, we analyze the efficiency on computation costs of the proposed scheme and the existing schemes for TMIS [12], [13], [50]. For all these given schemes, we separately tabulate the user side and server side computational costs for all the login and authorization phases of the proposed scheme. The computational costs of the proposed scheme, Awasthi-Srivastava's scheme, Jiang *et al.*'s scheme and Mishra *et al.*'s scheme are 107.88, 147.14, 153.22, and 68.06 milliseconds, respectively. The proposed scheme requires less computational cost as compared to that for Awasthi-Srivastava's scheme and Jiang *et al.*'s scheme. Though the proposed scheme requires more computational cost as compared to that for Mishra *et al.*'s scheme, it provides various security and functionality features as shown in Tables 4 and 5.

In Table 8, we tabulate the number of bits required for each message communication in the proposed fine-grained access control scheme. We assume that bit size of the identity, timestamps and random numbers are 160 bits, 32 bits and 128 bits, respectively. The hash output is 160 bits (if we take $H(\cdot)$ as SHA-1 [43]), the block size of symmetric encryption/decryption (for example, if we apply AES-128 [4]) is 128 bits, and the prime number is 160 bits. Since registration phase is executed only once, we concentrate on the login and authentication/authorization phases for calculation of communication and computation costs.

The communication costs for transmission of the messges $Msg_1 = \langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j|| GID_j))\rangle$, $Msg_2 = \langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s|| PM|| E'|| T_s|| T_{U_j}), E_j\rangle$ and $Msg_3 = \langle H(PM' ||RN_s ||M_j), M_j, E_{K_s}(PM') ||RN_j\rangle$ are

TABLE 8. Message sizes.

Message	Size (in bits)
$\langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j\|\|GID_j))\rangle$	384
$\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s,$ $H(RN_s\|\|PM\|\|E'\|\|T_s\|\|T_{U_j}\|\|K_j), E_j\rangle$	$800 + \|E_j\|$
$\langle H(PM'\|\|RN_s\|\|M_j), M_j, E_{K_s}(PM'), RN_j\rangle$	576

TABLE 9. Comparison of commuinication costs.

Scheme	Total bits requred
Awasthi-Srivastava [63]	544
Jiang *et al.* [23]	896
Mishra *et al.* [24]	704
Our	$1760 + \|E_j\|$

384, $800 + |E_j|$ and 576 bits, respectively, where $|E_j|$ denotes the number of bits present in E_j.

We compare the total amount of bits needed for message exchanges among the proposed scheme and other related existing schemes. From Table 9, it is noted that our scheme requires a sum total of $1760 + |E_j|$ bits, whereas the schemes of Awasthi and Srivastava, Jiang *et al.* and Mishra *et al.* require 544, 896 and 704 bits, respectively. Though the proposed scheme requires more communication cost as compared to other schemes, it provides various security and functionality features as shown in Tables 4 and 5.

VIII. CONCLUSION

We have presented a new fine grained access control scheme with user authentication for TMIS. The proposed scheme uses both the user password and biometric to provide better security as compared to password based authentication schemes. The proposed scheme provides group-based user authentication depending on the access rights provided for the genuine users in TMIS. The proposed scheme is tested for its security using the formal security under the widely-accepted Real-Or-Random model and also mutual authentication using the broadly-used BAN logic. In addition, the informal security analysis shows that the proposed scheme is also resistant to various known attacks. The proposed scheme also provides better security and functinality features as compared to other existing schemes for TMIS. Furthermore, the communication and compuational costs of the proposed scheme are comparable with those for other existing schemes for TMIS.

Future work includes implementing and evaluating the proposed scheme in a real-world environment. This would allow us to fine-tune the scheme, if necessary, to offer better security and performance in a real-world deployment.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback on the paper which helped them to improve its quality and presentation.

REFERENCES

[1] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 1999, pp. 388–397.

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

[4] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[5] D. Wang and C. G. Ma, "Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards," *J. China Universities Posts Telecommun.*, vol. 19, no. 5, pp. 104–114, 2012.

[6] D. Wang, C. G. Ma, P. Wang, and Z. Chen, "Robust smart card based password authentication scheme against smart card security breach," *Cryptol. ePrint Archive*, vol. 2012, no. 439, pp. 1–35, 2012. [Online]. Available: http://eprint.iacr.org/2012/439.pdf

[7] D. He, C. Jianhua, and Z. Ru, "A more secure authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 37, no. 3, pp. 1989–1995, 2012.

[8] H. Chen, J. Lo, and C. Yeh, "An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems," *J. Med. Syst.*, vol. 36, no. 6, pp. 3907–3915, Dec. 2012.

[9] Y. F. Chang, S. H. Yu, and D. R. Shiao, "A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *J. Med. Syst.*, vol. 37, p. 9902, Apr. 2013.

[10] A. K. Das and A. Goswami, "A secure efficient uniqueness and-anonymity-preserving remote user authentication scheme for connected health care," *J. Med. Syst.*, vol. 37, no. 3, p. 9948, Jun. 2013.

[11] C. Guo and C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 6, pp. 1433–1440, 2013.

[12] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "Robust chaotic map-based authentication and key agreement scheme with strong anonymity for telecare medicine information systems," *J. Med. Syst.*, vol. 38, p. 12, Feb. 2014.

[13] D. Mishra, J. Srinivas, and S. Mukhopadhyay, "A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 38, no. 120, Oct. 2014.

[14] A. K. Das and B. Bruhadeshwar, "An improved and effective secure password-based authentication and key agreement scheme using smart cards for the telecare medicine information system," *J. Med. Syst.*, vol. 37, p. 9969, Sep. 2013.

[15] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ECC-based user anonymity preserving session initiation authentication protocol using smart card," *Peer-Peer Netw. Appl.*, vol. 9, p. 171, Sep. 2016.

[16] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[17] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Comput. Conf.*, 1979, pp. 313–317.

[18] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Heidelberg, Germany: Springer-Verlag, 2005, pp. 457–473.

[19] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology*, vol. 196. Springer, 1984, pp. 37–53.

[20] D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing," in *Advances in Cryptology—CRYPTO 2001*, Santa Barbara, CA, USA, Aug. 2001, pp. 213–229.

[21] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Proc. IMA Int. Conf. Cryptogr. Coding*, Cirencester, U.K., Dec. 2001, pp. 360–363.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2006, pp. 89–98.

[23] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 673–686, Apr. 2011.

[24] S. Chatterjee and S. Roy, "Cryptanalysis and enhancement of a distributed fine-grained access control in wireless sensor networks," in *Proc. IEEE Int. Conf. Adv. Comput. Commun. Inf. (ICACCI)*, New Delhi, India, Sep. 2014, pp. 2074–2083.

[25] S. Ruj, A. Nayak, and I. Stojmenovic, "Distributed fine-grained access control in wireless sensor networks," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, Anchorage, AK, USA, May 2011, pp. 352–362.

[26] S. Chatterjee and A. K. Das, "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1752–1771, 2015.

[27] S. Chatterjee, A. K. Gupta, and G. Sudhakar, "An efficient dynamic fine grained access control scheme for secure data access in cloud networks," in *Proc. IEEE Int. Conf. Electr. Comput. Commun. Technol. (ICECCT)*, Coimbatore, India, Mar. 2015, pp. 1–8.

[28] S. Chatterjee, A. K. Gupta, V. K. Mahor, and T. Sarmah, "An efficient fine grained access control scheme based on attributes for enterprise class applications," in *Proc. Int. Conf. Signal Propag. Comput. Technol. (ICSPCT)*, Ajmer, India, 2014, pp. 313–317.

[29] V. Odelu, A. K. Das, and A. Goswami, "An efficient biometric-based privacy-preserving three-party authentication with key agreement protocol using smart cards," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4136–4156, 2015.

[30] W. Stallings, *Cryptography and Network Security: Principles and Practices*. 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

[31] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[32] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 4, pp. 33:1–33:16, 2010, doi: 10.1109/TDSC.2016.2616876.

[33] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. System Secur.*, vol. 13, no. 4, p. 33, 2010.

[34] P. Bergamo and P. D'Arco, A. D. Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Trans. Circuits Syst.*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005.

[35] K. Simoens, J. Bringer, H. Chabanne, and S. Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 833–841, Apr. 2012.

[36] Q. Zhang, Y. Yin, D. Zhan, and J. Peng, "A novel serial multimodal biometrics framework based on semisupervised learning techniques," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1681–1694, Sep. 2014.

[37] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 771–780, Dec. 2010.

[38] M. A. Pathak, B. Raj, S. D. Rane, and P. Smaragdis, "Privacy-preserving speech processing: Cryptographic and string-matching frameworks show promise," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 62–74, Mar. 2013.

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

[40] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 14, pp. 4–20, Dec. 2004.

[41] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *Adv. Cryptology-Eurocrypt*, Interlaken, Switzerland: Springer, 2004, pp. 523–540.

[42] A. K. Das, D. Mishra, and S. Mukhopadhyay, "An anonymous and secure biometric-based enterprise digital rights management system for mobile environment," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 3383–3404, 2015.

[43] (Apr. 1995). National Institute of Standards and Technology (NIST), U.S. Department of Commerce. *Secure Hash Standard, FIPS PUB 180-1*. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenot%ice.pdf

[44] National Institute of Standards and Technology (NIST), U.S. Department of Commerce. (Nov. 2001). *Advanced Encryption Standard FIPS PUB 197*. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[45] F. B. Hildebrand, *Introduction to Numerical Analysis*, 2nd ed. New York, NY, USA: Dover, 1974.

[46] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. 8th Int. Workshop Theory Pract. Public Key Cryptogr. (PKC)*, vol. 3386. Les Diablerets, Switzerland, 2005, pp. 65–84.

[47] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," *Appl. Cryptogr. Netw. Secur.*, Apr. 2008, pp. 277–295.

[48] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[49] P. F. Syverson and I. Cervesato, "The logic of authentication protocols," in *in Revised Versions of Lectures Given During the IFIP WG 1.7 International School on Foundations of Security Analysis and Design on Foundations of Security Analysis and Design: Tutorial Lectures* (FOSAD). London, U.K.: Springer-Verlag, 2001, pp. 63–136.

[50] A. K. Awasthi and K. Srivastava, "A biometric authentication scheme for telecare medicine information systems with nonce," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 37, p. 9964, Oct. 2013.

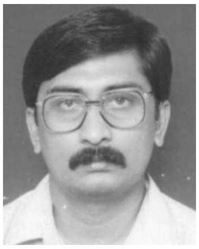
[51] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications* (SCI Series). Berlin, Germany: Springer, 2011.

[52] D. He, N. Kumar, J.-H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 30–37, Feb. 2014.

SANTANU CHATTERJEE received the Ph.D. degree in computer science and engineering and the master's degree in computer science and engineering from Jadavpur University, India. He is currently a Scientist with the Research Center Imarat in Directorate of ICT, Defence Research and Development Organization, Hyderabad, India. He has authored over 20 papers in international journals and conferences. His current research interests include cryptography, wireless sensor network security, data mining, and enterprise resource planning.

SANDIP ROY received the M.Tech. degree in computer science and technology from the West Bengal University of Technology, India. He is currently pursuing the Ph.D. degree in computer science and engineering from Jadavpur University, Kolkata, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Asansol Engineering College, India. His current research interests include cryptography, wireless sensor network security, and access control. He has authored four international journal and conference papers in his area of research.

ASHOK KUMAR DAS (M'17) received the Ph.D. degree in computer science and engineering, the M.Tech. degree in computer science and data processing, and the M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently an Assistant Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India. His current research interests include cryptography, wireless sensor network security, hierarchical access control, data mining, security in vehicular ad hoc networks, smart grid and cloud computing, and remote user authentication. He has authored over 130 papers in international journals and conferences in the above-mentioned areas. He has served as a Program Committee Member in many international conferences. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the Editorial Board of the *KSII Transactions on Internet and Information Systems* and the *International Journal of Internet Technology and Secured Transactions* (Inderscience). He is a Guest Editor of the *Computers & Electrical Engineering* (Elsevier) for the special issue on big data and IoT in e-healthcare.



SAMIRAN CHATTOPADHYAY received the Ph.D. degree from Jadavpur University and the master's and bachelor's degrees from IIT Kharagpur, Kharagpur. He is currently a Professor with the Department of Information Technology, Jadavpur University, Kolkata, India. He is having over 25 years of teaching experience at Jadavpur University, 4 years of industry experience, and 12 years of technical consultancy in the reputed industry houses. He has authored over 110 papers in international journals and conferences. His research area includes algorithms, wireless networks, network security, intelligent computing, bio informatics, cloud computing, and distributed computing.

NEERAJ KUMAR (M'16) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra, India, in 2009. He was a Post-Doctoral Research Fellow with Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has guided many research scholars leading to Ph.D. and M.E./M.Tech. He has authored over 160 technical research papers published in leading journals and conferences from the IEEE, Elsevier, Springer, and John Wiley. Some of his research findings are published in top cited journals, such as the IEEE Transactions on Industrial Electronics, the IEEE Transactions on Dependable and Secure Computing, the IEEE Transactions on Intelligent Transportation Systems, the IEEE Transactions on Consumer Electronics, the *IEEE Network*, the *IEEE Communications*, the *IEEE Wireless Communications*, the IEEE Internet of Things Journal, the IEEE Systems Journal, *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and *Computer Communications*.

ALAVALAPATI GOUTHAM REDDY (S'15) received the M.Tech. degree in computer science and engineering from Christ University, India, in 2013, and the Ph.D. degree in information security from Kyungpook National University, South Korea, in 2017. He is currently a Post-Doctoral Fellow with the KINDI Laboratory, Qatar University, Qatar. He holds several publications in the area of cryptographic authentication protocols. His primary research interests revolve around cryptography and information security. He is a Student Member of ACM.

KISUNG PARK received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics Engineering. His research interests include authentication, computer networks, Internet of Things, VANET, and information security.

YOUNGHO PARK received the B.S., M.S., and Ph.D. degrees from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively, all in electronic engineering. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronics Engineering, Kyungpook National University, South Korea. His research interests include information security and computer networks.