Design and Analysis of Remote Authentication and Access Control Schemes for Wireless Communications

Thesis submitted for the

Doctor of Philosophy (Engineering)

Degree of Jadavpur University, Kolkata, India

By

Sandip Roy

Department of Information Technology
Jadavpur University
Kolkata, West Bengal 700 032, India

2018

Design and Analysis of Remote Authentication and Access Control Schemes for Wireless Communications

by

Sandip Roy

Thesis submitted for the

Doctor of Philosophy (Engineering)

Degree of Jadavpur University, Kolkata, India

Supervisors:

Dr. Santanu Chatterjee
Scientist D
Research Centre Imarat,
Defence Research and
Development Organization,
Hyderabad 500 069, India

Dr. Ashok Kumar Das
Associate Professor
Center for Security, Theory
and Algorithmic Research,
International Institute of
Information Technology,
Hyderabad 500 032, India

Dr. Samiran Chattopadhyay
Professor
Department of Information
Technology,
Jadavpur University,
Kolkata 700 032, India

2018

Jadavpur University
Kolkata 700 032, India

INDEX NO. 32/16/E

1. Title of the Thesis

Design and Analysis of Remote Authentication and Access Control Schemes for Wireless Communications

2. Name, Designation and Institution of the Supervisors:

1. Dr. Santanu Chatterjee
   Scientist D
   Research Centre Imarat
   Defence Research and Development Organization
   Hyderabad 500 069, India

2. Dr. Ashok Kumar Das
   Associate Professor
   Center for Security, Theory and Algorithmic Research
   International Institute of Information Technology
   Hyderabad 500 032, India

3. Dr. Samiran Chattopadhyay
   Professor
   Department of Information Technology
   Jadavpur University
   Kolkata 700 032, India

3. List of Publications:

1. Journal papers

1. Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Saru Kumari, and Minho Jo. "Chaotic Map-based Anonymous User Authentication Scheme with User Biometrics and Fuzzy Extractor for Crowdsourcing Internet of Things," in ***IEEE Internet of Things Journal***, Vol. 5, No. 4, pp. 2884 - 2895, 2018. (2018 SCI Impact Factor: 9.515)

2. Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, and Athanasios V. Vasilakos. "On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services," in ***IEEE Access***, Vol. 5, No. 1, pp. 25808-25825, 2017. (2018 SCI Impact Factor: 4.098)

3. Sandip Roy, Ashok Kumar Das, Santanu Chatterjee, Neeraj Kumar, Samiran Chattopadhyay, and Joel J. P. C. Rodrigues. "Provably Secure Fine-Grained Data Access Control over Multiple Cloud Servers in Mobile Cloud Computing Based Healthcare Applications," in ***IEEE Transactions on Industrial Informatics***, Vol. 15, No. 1, pp. 457-468, 2019. (2018 SCI Impact Factor: 7.377)

4. Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, and Athanasios V. Vasilakos. "Secure Biometric-Based Authentication Scheme using Chebyshev Chaotic Map for Multi-Server Environment," in ***IEEE Transactions on Dependable and Secure Computing***, Vol. 15, No. 5, pp. 824-839, 2018. (2018 SCI Impact Factor: 4.41)

5. Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, Alavalapati Goutham Reddy, Kisung Park, and YoungHo Park. "On the Design of Fine Grained Access Control with User Authentication Scheme for Telecare Medicine Information Systems," in ***IEEE Access***, Vol. 5, No. 1, pp. 7012-7030, 2017. (2018 SCI Impact Factor: 4.098)

2. International conference papers

1. Sandip Roy, Santanu Chatterjee, Samiran Chattopadhyay, and Amit Kumar Gupta "A Biometrics-based Robust and Secure User Authentication Protocol for e-Healthcare Service," in ***IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)***, pages 638-644, Jaipur, India, 2016.

4. List of Patents: Nil

5. List of Presentations in National/International Conferences

1. Sandip Roy, Santanu Chatterjee, Samiran Chattopadhyay, and Amit Kumar Gupta "A Biometrics-based Robust and Secure User Authentication Protocol for e-Healthcare Service," in ***IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)***, pages 638-644, Jaipur, India, 2016.

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled "Design and Analysis of Remote Authentication and Access Control Schemes for Wireless Communications", submitted by Sri Sandip Roy, who got his name registered on 20th October, 2016 for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of Santanu Chatterjee, Ashok Kumar Das and Samiran Chattopadhyay, and that neither his thesis nor any part of the thesis has been submitted for any degree or any academic award anywhere before.

1.

Signature of the Supervisor and date with Official Seal

Dr. Santanu Chatterjee
Scientist D
Research Centre Imarat,
Defence Research and
Development Organization,
Hyderabad 500 069, India

2.

Signature of the Supervisor and date with Official Seal

Dr. Ashok Kumar Das
Associate Professor
Center for Security, Theory
and Algorithmic Research,
International Institute of
Information Technology,
Hyderabad 500 032, India

3.

Signature of the Supervisor and date with Official Seal

Dr. Samiran Chattopadhyay
Professor
Department of Information
Technology,
Jadavpur University,
Kolkata 700 032, India

Acknowledgments

I would like to acknowledge the moral and intellectual supports given to me by my supervisors Dr. Santanu Chatterjee, Dr. Ashok Kumar Das and Dr. Samiran Chattopadhyay during my PhD program. Thanks to my supervisors' constant guidances and approaches for which this long and difficult journey becomes smooth and also very interesting.

First and foremost, I would like to thank my supervisor, Dr. Santanu Chatterjee for giving me the guidance, encouragement, counsel throughout my research and painstakingly reading my reports. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis.

I am extremely grateful to my other supervisor, Dr. Ashok Kumar Das. His way to do research as well as his attitude towards study and analysis of any particular subject influenced me immensely, and I still feel there is a lot to learn from him. Among other things, I have always admired his ability to discuss research problems from scratch to formalize rigorously, his scientific bravery in supporting ideas that sounded completely mental at first glance. In additio, his ideas and style of writing have always impressed me and it also influenced my own style of work. Most important of all was probably his calm and constant belief in my ability to do the PhD work.

I am also very thankful to my other supervisor, Dr. Samiran Chattopadhyay for helping me a lot when I started my PhD work at the Jadavpur University, Kolkata, India. I thank him for his wisdom, many fruitful discussions, moral support, and fruitful cooperations over various research wroks.

I thank Dr. Bhaskar Sardar, Head, Department of Information Technology, Jadavpur University for serving on my Doctoral Scrutiny Committee.

I would also like to thank my parents, my wife, my daughter, my son and my mother-in-law for their moral support and patience during the course of my research work. Finally, I would like to thank all of them whose names are not mentioned here but have helped me in any way to accomplish the work.

Department of Computer Science and Engineering
Asansol Engineering College
West Bengal, India **Sandip Roy**

Abstract

Wireless communication is susceptible to various kinds of security attacks, such as replay attack, man-in-the-middle attack, privileged-insider attack, impersonation attacks, online/offline guessing attacks, stolen-verifier attack and denial-of-service attack. Hence, to achieve hazard-free service, design of remote user authentication and remote access control mechanisms is highly essential in various applications that involve wireless communication. In this thesis, we aim to study remote user authentication and access control problems in the folliwing areas: 1) multi-server authentication in wireless medium, 2) user authentication in crowdsourcing Internet of Things (IoT) environment, 3) user authentication in distributed mobile cloud computing environment, and 4) fine-gained access control with user authentication in telecare medicine information system (TMIS).

In the first study, we propose a new authentication scheme for multi-server environments using Chebyshev polynomial and chaotic map. According to the proposed scheme, a user does not need to maintain different credentials to register with various servers. We use the user biometric along with password for authorization and access to various application servers. At the time of authentication, a session key is established between the respective server and user without involving the registration center (RC). This significantly reduces the communication cost, and it makes the authentication process faster and efficient. The proposed scheme is light-weight compared to other related schemes. Our scheme provides strong authentication, supports biometrics and password change phase, and dynamic server addition phase. We perform the formal security verification using the broadly-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) software tool to show that the presented scheme is secure. In addition, we use the formal security analysis using the Burrows-Abadi-Needham (BAN) logic along with the Real-Or-Random (ROR) model, and prove that the proposed scheme is secure against different known attacks. High security and significantly low computation and communication costs make our scheme is very suitable for multi-server environments as compared to other existing related schemes.

The second study is based on designing a new chaotic map-based anonymous three-factor user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing IoT environment. The three factors involved in the proposed scheme are: 1) smart card, 2) password and 3) personal biometrics. The proposed scheme avoids computationally expensive elliptic curve point multiplication or modular exponentiation operation, which are based on public key cryptosystem. Hence, it is lightweight and efficient. The formal security verification

using the widely-accepted verification tool, called the ProVerif 1.93, shows that the proposed scheme is secure. In addition, we present the formal security analysis using the both widely-accepted ROR model and BAN logic. With combination of high security and appreciably low communication and computational overheads, the proposed scheme is practical for battery limited devices used in crowdsourcing IoT environment.

In the third study, we propose a new secure and lightweight mobile user authentication scheme for distributed mobile cloud computing environment. The proposed protocol is based on one-way cryptographic hash function, bitwise exclusive-OR (XOR) operation and fuzzy extractor technique. The proposed scheme supports secure key exchange, and user anonymity and untraceability properties. The proposed scheme does not involve registration center (RC), smart card generator (SCG) or identity provider (IdP) in the authentication and key establishment process. Through the informal (non-mathematical) security analysis and also the rigorous formal security analysis using ROR model, it has been demonstrated that the proposed scheme is secure against possible well-known passive and active attacks, and also provides user anonymity. Moreover, we provide formal security verification through ProVerif 1.93 simulation tool for the proposed scheme. In addition, we perform the authentication proof of our proposed scheme using the BAN logic. Since the proposed scheme does not exploit any resource constrained cryptosystem, it has lowest computation cost in compare to the existing related schemes.

Final study involves on design of fine-grained data access control of server data with suitable authentication scheme in TMIS and e-healcare system. It is worth noting that none of the existing user authentication protocols designed for TMIS and e-healthcare applications provide any fine-grained access control of user sensitive data. The proposed scheme also provides user anonymity during any message communication that protects patient's privacy as the user never delivers his/her original identity to the the medical server. We present the formal security analysis using both the widely-accepted ROR model and BAN logic. The proposed scheme supports user anonymity, forward secrecy, and efficient password change without contacting the remote server. In addition, as compared to other related schemes proposed in TMIS, the proposed scheme is superior with respect to communication and computation costs. Better trade-off among security and functionality features, and communication and computation costs makes the proposed scheme suitable and practical for telecare medicine environment as compared to other existing related schemes.

Dissemination of Work

Chapter #4.

Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, and Athanasios V. Vasilakos. "Secure Biometric-Based Authentication Scheme using Chebyshev Chaotic Map for Multi-Server Environment," in ***IEEE Transactions on Dependable and Secure Computing***, Vol. 15, No. 5, pp. 824-839, 2018. (2018 SCI Impact Factor: 4.41)

Chapter #5.

Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Saru Kumari, and Minho Jo. "Chaotic Map-based Anonymous User Authentication Scheme with User Biometrics and Fuzzy Extractor for Crowdsourcing Internet of Things," in ***IEEE Internet of Things Journal***, Vol. 5, No. 4, pp. 2884 - 2895, 2018. (2018 SCI Impact Factor: 9.515)

Chapter #6.

Sandip Roy, Santanu Chatterjee, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, and Athanasios V. Vasilakos. "On the Design of Provably Secure Lightweight Remote User Authentication Scheme for Mobile Cloud Computing Services," in ***IEEE Access***, Vol. 5, No. 1, pp. 25808-25825, 2017. (2018 SCI Impact Factor: 4.098)

Chapter #7.

Santanu Chatterjee, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Neeraj Kumar, Alavalapati Goutham Reddy, Kisung Park, and YoungHo Park. "On the Design of Fine Grained Access Control with User Authentication Scheme for Telecare Medicine Information Systems," in ***IEEE Access***, Vol. 5, No. 1, pp. 7012-7030, 2017. (2018 SCI Impact Factor: 4.098)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid development of wireless communication and Internet technology facilitates us to enormously use various Internet-based applications. Nowadays, in several day-to-day affairs, people instantly avail on-line remote services, such as Internet banking, e-healthcare services, online shopping, smart home application, and smart vehicular systems. Users access these online services through a public channel. As wireless communication through a public channel is always susceptible to various kinds of threats and attacks, assurance of information security is highly essential for a hazard-free use of these services. User authentication is an important security mechanism through which a remote server verifies the authenticity of a user before providing any service to that user. Also, it prevents an unauthorized person from any illegal access to the remote services provided by the application server.

This thesis keeps an aim to study, analyze and design remote user authentication and access control protocols on four different application areas that require wireless communication. We consider the following application areas: 1) user authentication in a multiserver environment, 2) user authentication in crowdsourcing IoT environment, 3) user authentication in a distributed mobile cloud computing environment, and 4) fine-grained access control with user authentication in the telecare medicine information system (TMIS).

In the following subsections, we discuss various security requirements, functionality requirements and also identify the network model for each of these application areas.

1.1 User authentication in multi-server environment

According to the nature of the application environment, user authentication schemes can be applied to the single-server environment as well as multi-server environment. In a multi-

server environment, a user needs to access various services and these services are provided by a number of remote application servers. Implementation of single-server user authentication scheme into a multi-server environment is tedious and error-prone. This is because each user needs to login into the individual remote server in a separate way, and therefore, the user needs to remember many sets of identities and passwords. This process is not only inefficient, but also leads to compromise of the user identities and passwords. In lieu of resolving the problem of single-server user authentication protocols, various multi-server user authentication protocols have been proposed in recent years.

Multi-server architecture is one of the solution to pass up multiple registrations to different application servers. It further avoids use of multiple smart-card and login information, such as identity and password. Therefore, the concept of multi-server provides a good platform to reduce extra overhead than use of multiple communications to different application servers [136]. Secure communication schemes for authentication and session key agreement for the multi-server environment should provide various security requirements, which are described below [33], [141], [177], [230].

1.1.1 Network model of a multi-server system

In a multi-server system, users can access any application server irrespective of their geographical location, which makes it greatly worthwhile for various applications, such as e-commerce, e-business, e-documentation, and e-healthcare [51], [63], [144], [166]. Multi-server architecture makes it simpler by bringing all of them into one platform with a common infrastructure, such as one-time registration, one smartcard and same credentials. Thus, the users of multi-server architecture are beneficiary in terms of cost, efforts and time. Figure 1.1 provides a generalized network model of a multi-server architecture [166]. First of all, all the servers are registered in the system by a trusted registration center (RC). Also, the users need to register in the system and this process is done by the RC, where a user sends a registration request to the RC, and the RC then sends the registration response to the user. Next, the user logins to a server using the secret credentials, and after verification of the credentials, a mutual authentication occurs between that user and the accessed server. Only after successful mutual authentication, both the parties, such as the user and the server establish a session key, which is further used for their secure communication.

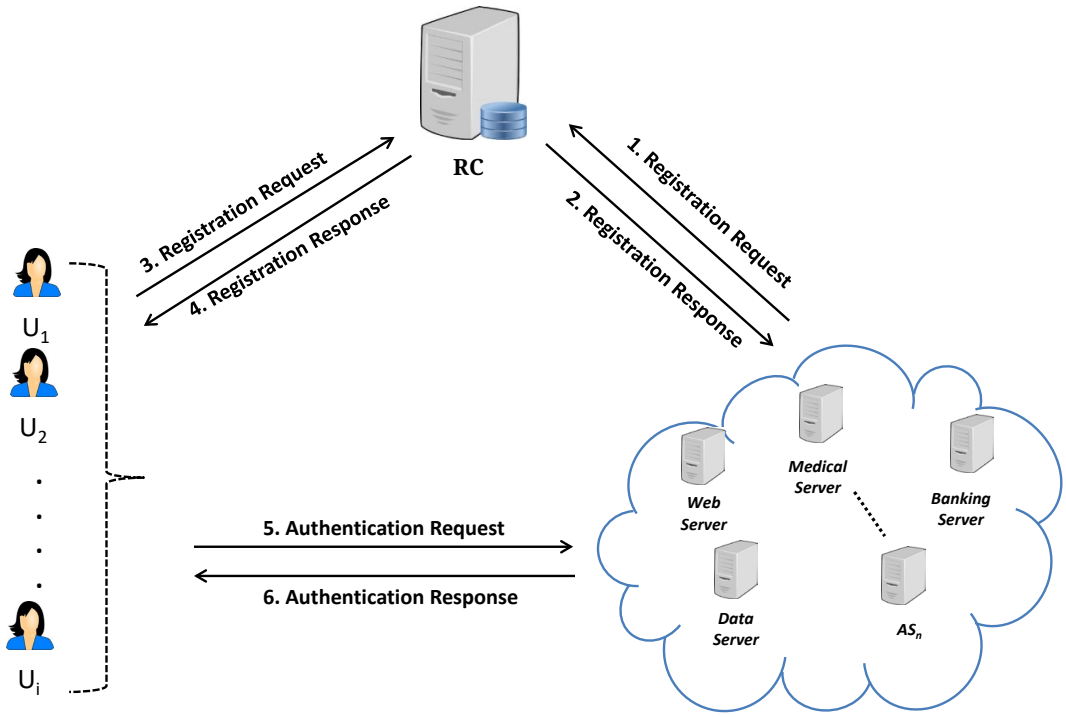


Figure 1.1: An architecture of muli-server system (Source: [166]).

1.1.2 Security requirements in multi-server environment

The following security requirements are essential in a multi-server environment:

- **Guessing attack:** An authentication protocol in a multi-server environment needs to defend guessing attack, where an attacker tries to guess the password or long-term secrets of an authorized user. This attack can be of two types: (i) offline guessing attack and (ii) online guessing attack. Generally, an online guessing attack is done by intercepting a communicated message, while an offline guessing attack is done by obtaining stored smart card data, mobile data and server data.

- **Replay attack:** Through replay attack, an adversary reuses an old message and replays it to an authorized user in order to deceive him/her. This attack is, therefore, an attempt by an unauthorized third party to record the transmitted messages.

- **Stolen-verifier attack:** An adversary may try to obtain user's password or other

secret credentials from one or more compromised server's verifier tables. Therefore, it is a basic security requirement that servers should not store any verifier or password tables directly.

- **Stolen smart card attack:** Usually, since the user smart card is not made up of tamper-resistant materials, an adversary might try to obtain stored information from a lost or stolen user smart card by various means. To prevent this attack, authentication protocols should be designed in such a way that even if the smart card is stolen/lost, an attacker should not be able to compute the secret credentials in polynomial time.

- **Insider attack:** Sometimes, a privileged user or a server administrator turns to be an adversary and tries to obtain secret credentials of other authorized users. An insider user can obtain the secret credentials of a registered user during the registration process of an authentication scheme, and then can try to misuse those credentials to mount other attacks, such as impersonation attacks.

- **Spoofing attack:** Through a spoofing attack, an adversary interrupts a message and substitutes it for his/her own message to deceive authorized communication parties in computing the wrong session key. In addition, the attacker can impersonate a legal user or a legal server to cheat the corresponding communication party and establish a common session key [230].

- **Known-key and session key security:** The security requirement of known key security demands that even if a session key is known to an adversary or a privileged user, he/she should not be able to compute other session keys. A communication protocol exhibits session key security (SK-security) if the session key cannot be obtained without any long-term secrets. Hence, it is essential to use both the temporal and long-term secrets in the construction of the session keys.

- **Perfect forward secrecy:** Perfect forward secrecy means that if one or more users or servers leave the network, they will not be able to compute shared session keys in future from previously obtained session keys.

- **Backward secrecy:** In a multi-server system, servers, and users can join system dynamically. Backward secrecy ensures that no entity should be able to compute previous session keys before their joining to the network.

- **Mutual authentication:** For any data communication, both user and server should contribute to mutually form a secret key in every session. Prior to this, they should verify the authenticity of one another.

- **User anonymity:** An adversary must not be able to compute an authorized user's original identity from any intercepted online message or any other stored parameters. Leakage, unintentional disclosure or any misuse of an authorized user's identity can break the user privacy that can affect the whole business.

1.1.3 Functionality requirements in multi-server environment

Traditional two party client-server authentication protocols may not provide a scalable solution for present network environments where personal and ubiquitous computing technologies are involved as they are now based on multi-server model. To achieve efficient authorized communication, multi-server based authentication protocols have been designed. The key feature of multi-server based protocols is one-time registration. The involvement of central authority in mutual authentication may be a bottleneck for a large network, and the servers may be semi-trusted [46]. A designed user authentication scheme for a multi-server environment should fulfill the following functional requirements [27], [129]:

- **Single registration:** Even if the system contains multiple servers, the user does not need to register them separately. The user should register only once with the registration center (RC) prior to making any login to any of the existing servers.

- **No verification or password table:** The registration center should not contain the password, identity or biometrics template of any user as the systems might turn vulnerable to various active attacks including registration center compromise attack, and secret leakage attack.

- **Low computation and communication costs:** In general, user smart card reader and mobile device have constrained resources with limited battery power. Hence, an authentication scheme should involve low computation and communication overheads at different phases of the login and authentication processes.

- **Efficient password change:** A user should be able to update or change his/her existing password in a local environment without involving the RC.

- **Dynamic server and user addition:** As the authentication scheme is inherently built for a multi-server environment, it should be scalable enough to dynamically add some new servers and users in the system.

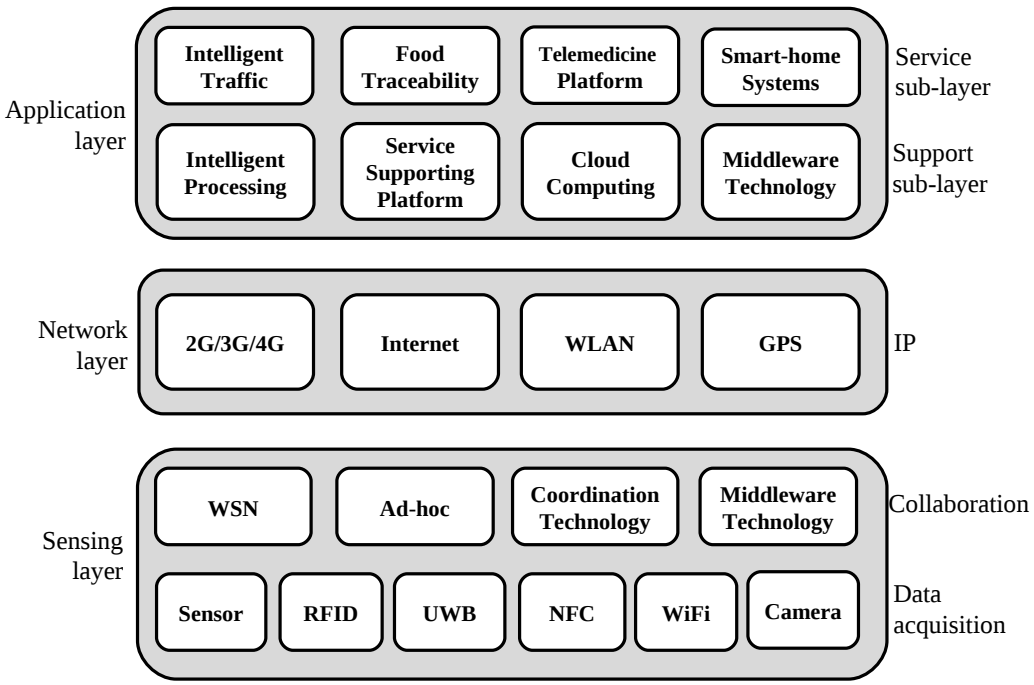
1.2 User authentication in crowdsourcing IoT environment

The Internet of Things (IoT) introduces a vision of a future Internet that aims to provide direct integration of the physical world into computer-based systems in order to reduce human involvement and produce more efficiency with cost optimization. IoT is a network of users, computing systems and well connected physical devices with sensing and actuating capabilities that communicate with each other devices using Internet communication protocol. The IoT is enabled by the latest developments in Radio-Frequency Identification (RFID), smart sensors, communication technologies, and Internet protocols [16]. The number of IoT devices is increased 31 percent year-over-year to 8.4 billion in 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach 7.1 trillion US dollars by 2020 [16].

The IoT is realized by three basic tasks: 1) perception, 2) transmission and 3) processing. Hence, any standard architecture of IoT contains three layers, namely, applications, network, and sensing. In the sensing layer, comprehensive perception is realized by various sensors through a collection of real-time data. Secure transmission and relay of data from the sensing layer to application layer are done by the network layer. Overall intelligent control is realized by the application layers through processing and intelligent analysis of the collected data [149]. Figure 1.2 shows a basic three-layer architecture model of IoT.

IoT is a data network that is for the people, by the people, community initiative [199]. Thus, IoT is considered as a crowdsourcing data network. The bandwidth is open and anyone can use it [199]. As communications of IoT are realized through the Internet infrastructure, it is not safe and it might provide ample space for the cyber attackers and other adversaries in the network. So, the security issues need to be considered seriously in the IoT environment.

User authentication plays a vital role in the IoT environment. A secure and efficient mutual authentication among IoT devices, users, and the gateway nodes must be established. In recent years, a number of user authentication schemes have been proposed. However, the security challenges are arising as IoT is being maneuvered in many new applications connecting many



WLAN: Wireless Local Area Network; **GPS:** Global Positioning System; **WSN:** Wireless Sensor Network; **RFID:** Radio Frequency Identification; **UWB:** Ultra-Wide Band; **NFC:** Near-Field Communication

Figure 1.2: Basic three layer architecture model of IoT (Source: [149]).

new devices. Users, IoT devices and gateway nodes need to be authenticated and authorized to prevent any illegal and adversarial activities on IoT data.

1.2.1 Network model of crowdsourcing IoT environment

IoT is composed of a large number of things (devices) that are connected through the Internet. IoT devices can be classified further into two categories [66]:

- **Physical object:** These can be smartphone, camera, sensor, vehicle, and drone.
- **Virtual object:** These include electronic ticket, agenda, book, and wallet.

IoT devices can conduct remote sensing, actuating (making an action) and support monitoring capabilities. IoT devices can be made smart enough so that they can operate without any human intervention [66]. Some commercial IoT smart devices include *Smart Door Locks*,

Connected Smart Kitchen, *Smart Home Apps*, *Smart Bike Locks & Trackers*, *Friday Smart Lock*, etc. [11]. For instance, the IoT smart device *Friday Smart Lock* has the following features [11]:

- It contains the interchangeable shells so that we can easily change out the look of the lock.
- Its battery life is about 3-4 months before needing recharge. It also works with Apple Homekit.
- It needs Bluetooth and Wifi Connectivity.

Figure 1.3 shows a generic IoT network architecture in which four different scenarios (e.g., home, transport, community and national) are depicted. Several smart devices, such as sensors and actuators are installed in various applications [38]. All the IoT smart devices are connected to the Internet via trusted gateway nodes (GWNs). The information accessed by the IoT devices can be also accessed by various users (e.g., a smart home user in a home application and a doctor in a healthcare application) [79]. Cyber-physical systems such as the smart grid, smart home and intelligent transportation are also parts of IoT ecosystem [19].

Figure 1.4 illustrates a generic IoT-based smart home application [188]. The smart devices are deployed into two groups: appliance and monitor. The devices installed in the appliance and monitor groups, known as the agents, communicate with the central controller via wireless communications. A user can control the smart home system by using the user interface. Moreover, the information gathered by any IoT smart device in the monitoring group can be accessed by a user [66].

1.2.2 Security requirements for crowdsourcing IoT

IoT not only has the same security issues as sensor networks, mobile communications networks and the Internet, but also has its specialties such as privacy issues, different authentication and access control network configuration issues, information storage and management and so on [111]. Data and privacy protection is one of the application challenges of IoT [10]. In IoT, RFID systems and sensors in wireless sensor networks (WSNs) perceive for the end of the information technology, which protect the integrity and confidentiality of information by the password encryption technology [15], [81], [111], [140], [198]. Figure 1.5 gives a summary of various kinds of security attacks in IoT environment [25], [149].

The following general security requirements are essential to secure an IoT network [66]:

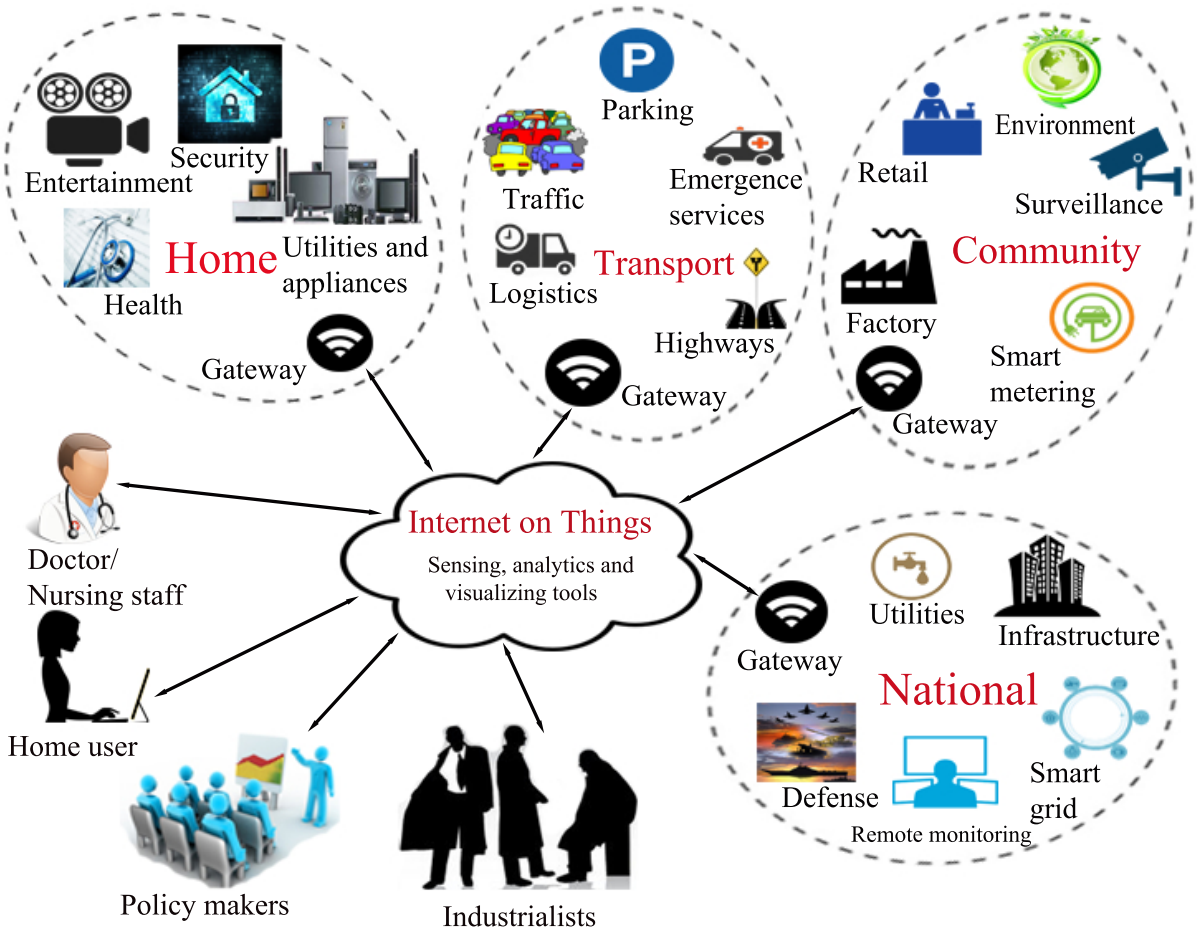


Figure 1.3: A general network model of IoT (Source: [38]).

- **Authentication:** It involves authentication of sensing devices, users and gateway nodes before allowing access to a restricted resource, or revealing crucial information.

- **Integrity:** The message or the entity under consideration must not be changed to ensure integrity.

- **Confidentiality:** Confidentiality or privacy of the wireless communication channel protects from the unauthorized disclosure of information.

- **Availability:** The relevant network services should be made available to authorized users even under denial-ofservice attacks on the system.

- **Non-repudiation:** It aims to prevent a mischievous entity from hiding his/her actions.

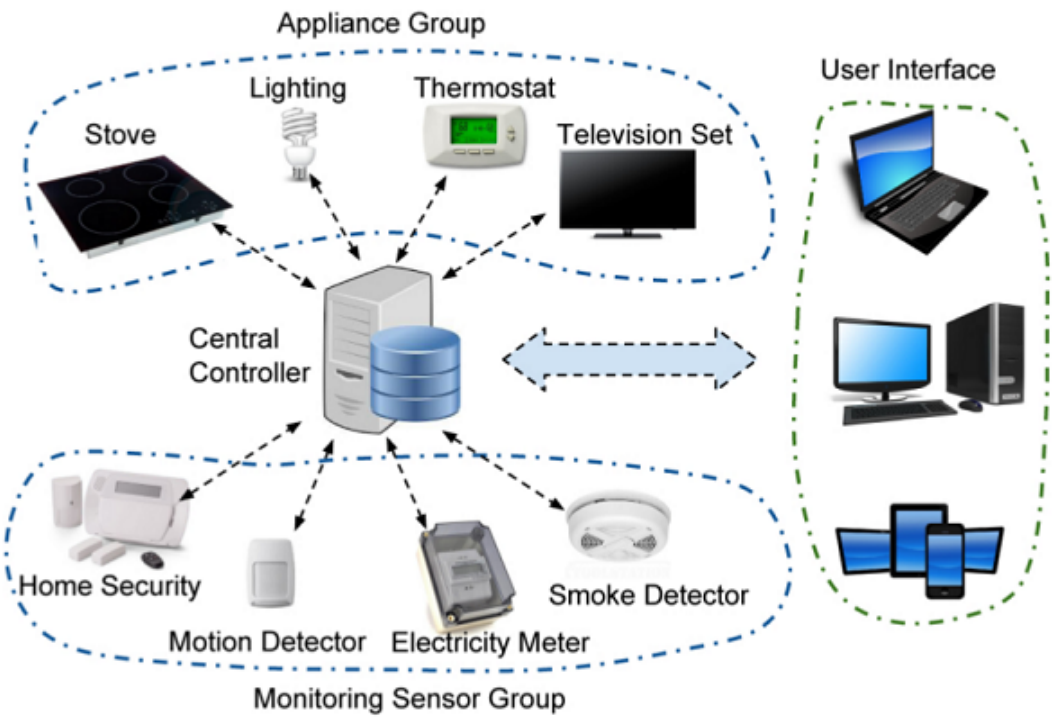


Figure 1.4: An IoT-based smart home application (Source: [188]).

- **Authorization:** It confirms that only the legitimate IoT sensing (smart) devices can supply information to network services.
- **Freshness:** It confirms that the information is fresh and the old messages cannot be replayed by any adversary.
- **Privacy:** The objective of this security requirement is to prevent private information from being leaked to malicious entities. Attacks on privacy are related to illegally gathering sensitive information about entities via eavesdropping.

Apart from the above security requirements, the following two important security properties should also be satisfied:

- **Forward secrecy:** If an IoT sensing node quits the network, any future messages after its exit must be prohibited.
- **Backward secrecy:** If a new IoT sensing node is added in the network, it must not have access to any previously transmitted message.

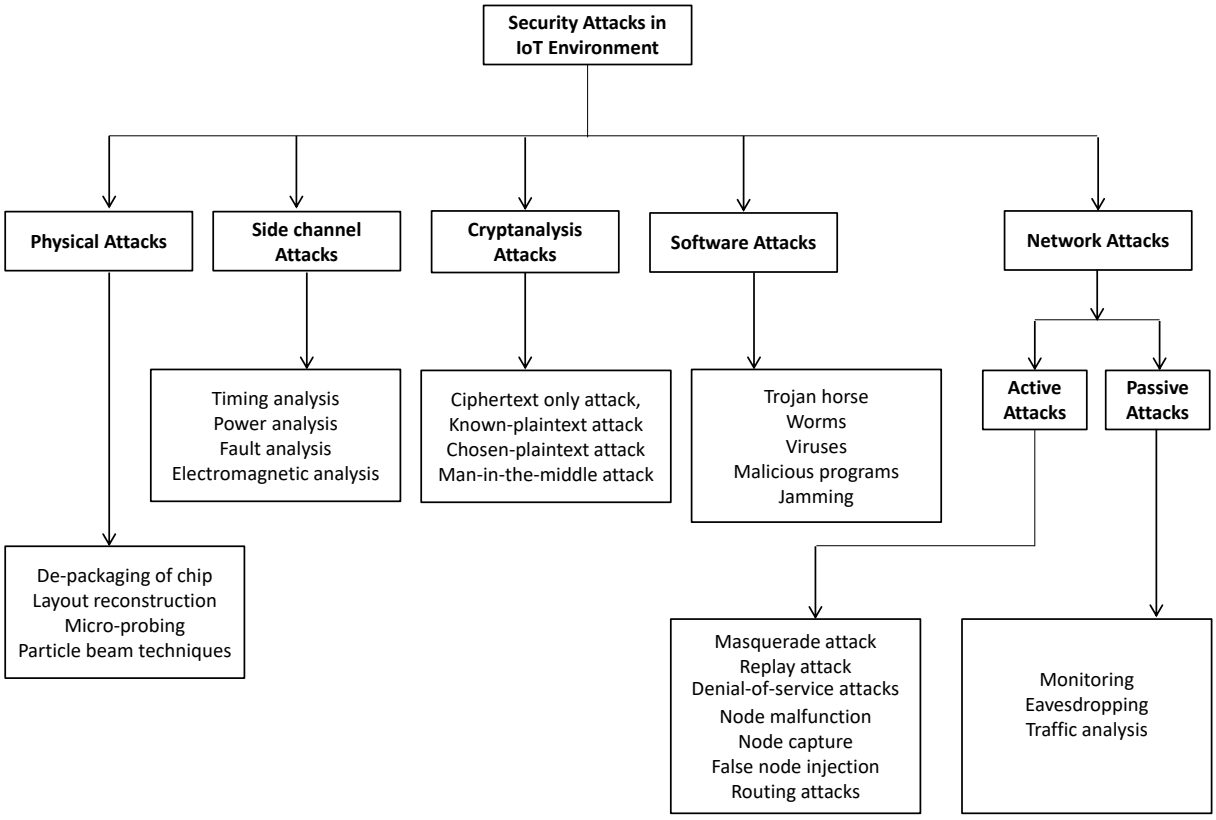


Figure 1.5: Summary of security attacks in IoT environment (Source: [25]).

1.2.3 Common security attacks in IoT environment

IoT exploits two basic technology, namely, WSN (Wireless Sensor Networks) and RFID (Radio Frequency Identification). Integration of these two technologies might provide many advantageous features to IoT environment. On the other side, this combination can invoke many security drawbacks into the system.

In this subsection, we list down a summary of most common types of security attacks applicable for IoT environment [90].

- **Botnets:** A botnet is a number of internet-connected private devices that are infected with malicious software and controlled as a group without the owner's knowledge. An adversary uses botnets to access private data of user device, executes distributed denial-of-service (DDoS) attack, and also sends spam messages. An IoT botnet is a group

of hacked computers, smart appliances and Internet-connected devices that have been co-opted for illicit purposes.

- **Distributed Denial-of-Service attack:** DDoS attack allows an attacker to successfully prohibit the availability of any service to an authorized user. Here, the adversary aims to keep the target user device too much "busy" through exploitation of various malwares, spams, or any other means of unnecessary services. As a result, the device data or application becomes unavailable to other devices in the network. Moreover, as the user device is often resource constrained in nature, device battery might drain in unnecessary communication or computation of various applications. DDoS does not usually try to steal information or leads to security loss, but the loss of reputation for the affected company can still cost a lot of time and money.

- **Data and identity theft:** IoT faces a serious security challenge in prevention of data and identity attack. From a stolen user device or intercepted message of an authorized user, an adversary might try to steal authorized user's data, thereby he/she may try to impersonate as a genuine user to access various services of the system. This scenario can create a huge risk in IoT enabled business. There are many scary real-life examples, where identity theft can affect a business beyond imagination of end users.

- **Social engineering:** Social engineering attacks are designed to target the user-computer interface to enable attackers to deceive a user into performing an action that will breach a system's information security [90]. Through this act, the end users are deceived or manipulated to deliver their private and secret information. The attacker exploits social engineering to breach user privacy and also to hack user device that can act as a gateway into other more powerful connected devices and sensitive information.

- **Man-in-ihe-middle attack:** A man-in-the-middle attack is a very serious threat into an IoT environment. Here, the adversary secretly intercepts between communication of two authorized users or "things" and deceive both by believing that they actually interact with legal parties with genuine messages. These attacks can be extremely dangerous in the IoT, because of the nature of the "things" being hacked. For example, these devices can be anything from garage door openers, smart TV's, industrial tools, and connected "things" like machinery and vehicles.

1.2.4 Functionality requirements for crowdsourcing IoT

In this subsection, we list down some basic functional requirements for crowdsourcing IoT environment.

- **Dynamically new device addition:** In an IoT crowdsourcing environment, it is very much essential that the authentication or access control protocol supports dynamic node addition facility. As a network device may be compromised by an adversary or device battery power may be exhausted, new device needs to be deployed into the existing network.

- **High scalability:** Support of high scalability is a basic functional requirement in a modern day IoT environment. With the increase in business volume, number and variety of IoT devices may increase exponentially in the IoT network. High scalability ensures that even if the number of sensing IoT devices are going to increase, the overall network performance should not be affected.

- **Diverse connectivity:** Probably the most familiar form of connectivity for the internet, and for IoT, is Ethernet. In addition to Ethernet, IoT devices can connect using a wide variety of other technologies. The connectivity objective is that an IoT platform should support as many modes of connection – wired and wireless – as possible. Wireless options include the following:

  - *ANT+ (pronounced ant plus):* It is a wireless protocol for monitoring sensor data such as a person's heart rate or a bicycl's tyre/tire pressure, as well as the control of systems like indoor lighting or a television set.
  - *Bluetooth:* Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.400 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs).
  - *General Packet Radio Services (GPRS):* It is a packet-based wireless communication service that promises data rates from 56 up to 114 Kbps and continuous connection to the Internet for mobile phone and computer users.
  - *EDGE (also known as Enhanced GPRS or EGPRS):* It is a data system used on top of GSM (Global System for Mobile communication) networks.

  - *Long-Term Evolution (LTE):* It is a standard for high-speed wireless communication for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA technologies.
  - *Near-field communication (NFC):* It is a set of communication protocols that enable two electronic devices, one of which is usually a portable device such as a smartphone, to establish communication by bringing them within 4 cm (1.6 in) of each other.
  - *Radio-Frequency Identification (RFID):* It uses the radio waves to read and capture information stored on a tag attached to an object.
  - *Wireless LAN (WLAN):* It is a wireless computer network that links two or more devices using wireless communication to form a local area network (LAN) within a limited area such as a home, school, computer laboratory, campus, and office building.
  - *ZigBee:* It is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection.

- **Mutual authentication:** Mutual authentication and key management are two important techniques to ensure secure communication in an IoT crowdsourcing environment. Intrinsically, IoT network contains various heterogeneous devices and because of its discriminating characteristics, the traditional key management protocols cannot be applicable in IoT. Mutual authentication between a remote end-user and a resource constrained sensor node inside a smart IoT environment is a challenging task.

- **Availability:** In an IoT crowdsourcing environment, real time device communication and control is done in keeping a real-world impact. For example, a user's home thermostat might need to be remotely controlled or a relief valve in an industrial plant needs to be opened or closed. Considering emergency of the situation, IoT platforms must therefore offer exceptionally high availability.

- **Minimum computation, communication and storage costs:** IoT environment usually contains various devices that are resource constrained and battery limited in nature. Many devices might have constraints in terms of storage memory too. Moreover,

IoT network devices might require frequent communication among them. Hence, an authentication protocol in an IoT environment should encompass minimum possible computation and communication overhead, as well as storage overhead.

1.2.5 IoT-based applications

Applications of IoT are diverse including infrastructure management in high-risk conditions, disaster management through environmental monitoring and providing remote health-care services, to list a few. In this subsection, we briefly mention some applications based on IoT crowdsourcing [19], [79]:

- **Wearable devices:** Health monitoring devices, navigation tools, and communication gadgets operate through wireless communication technologies, such as Bluetooth and local Wi-Fi. Wearable devices are usually resource constrained, and IoT application needs to be energy efficient [65], [67].

- **Telemedicine:** Through wireless sensor healthcare networks, connected wearable device, efficient and secure application software, IoT can provide remote service in an e-healthcare system.

- **Industrial IoT:** Through big data analytics, sensors and industrial IoT are enabling machines to become more consistent and accurate in communicating their data. Conventional automation methods can be transformed into machine-to-machine communication using wireless technologies and innovative hardware. Improved quality control and sustainability can also be achieved using industrial IoT technologies.

- **Retail:** IoT realizes the needs of customers as well as the needs of businesses. Through the services of comparison of product price, availability of a product in shops, and comparison of quality, IoT helps companies to improve their business and meet the needs of customers.

- **Energy management:** Smart grids collect data which is analyzed for behavior patterns of electricity suppliers and consumers to improve the economics and efficiency of usage. These are also highly needed because power outages at individual homes are detected quickly thereby providing a distributed energy system [98], [143], [228].

- **Agriculture:** Sensing for moisture in soil and nutrients, controlled watering of plants and determining customized fertilizers provides many advantages of using IoT in agriculture [99], [103], [125].

- **Smart home:** The appliances and devices in a smart home can communicate with each other and the surrounding environment. They enable controlling and customizing the home environment to provide efficient energy management, better security and user experiences in addition to saving time and money [215].

1.3 User authentication in mobile cloud computing environment

Mobile Cloud Computing (MCC) is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to mobile users, network operators, as well as cloud computing providers [4]. In MCC, mobile user's data is stored in remote distributed cloud servers and to get rid of the resource-constrained issue of existing mobile devices, data processing and execution responsibility are shifted to the cloud environment. MCC provides business opportunities for mobile network operators as well as cloud providers [4]. ABI Research is a market-foresight advisory firm providing strategic guidance on the most compelling transformative technologies. According to the ABI Research report about the increasing popularity of MCC [1], it was forecasted that within 2015, more than 240 million of mobile customers will use cloud services with an earning revenue of 5.2 billion US dollars [147].

In the past few years, researchers have made it possible to come up with several emerging applications of cloud computing for mobile users, including education and learning [48], cloud-assisted IoT [184], application processing [218], mobile social networks [113], cloud storage [50], cloud-based next generation cellular network [35], data sharing [48], cloud mobile media [218], [229], mobile commerce system [221], and mobile gaming [54].

Among various other challenges faced in MCC environment, trust, security, and privacy issues are more challenging in future days. This is due to several reasons, such as insecure public wireless transmission medium, resource-constraint mobile devices, distributed cloud storage and processing, and heterogeneous environments [147]. In order to prevent illegal access, cloud providers should support a secure authentication scheme for users using mobile devices. After authentication, the user can access the resources and available services from

the cloud service provider provided that they are mutually authenticated by each other.

In MCC, it is very much essential to establish and maintain end user's trust by protecting user privacy and data/application secrecy from adversaries. The general security requirements for MCC are summarized in the following [147] (adopted from ITU [7] and US National Security Agency [9]).

1.3.1 Network model of a mobile cloud computing system

An architecture for distributed mobile cloud computing is represented in Figure 1.6. The system model contains the following components: 1) mobile users (MU_i), 2) cloud server or cloud service provider (CS_j), and 3) trusted registration center (RC). The system contains a set of m legal mobile users, $M = \{MU_i | i = 1, 2, \cdots, m\}$, a set of n cloud servers, $N = \{CS_j | j = 1, \ldots, n\}$ and the trusted RC. A legal user or an unregistered external person may execute malicious activities in the system, called an adversary $\mathcal{A}$. From different cloud service providers, a mobile user can access multiple mobile cloud computing services. The RC needs not to be involved in the login and authentication processes.

To access a mobile cloud computing service, a mobile user MU_i requests the cloud service through an installed mobile App or web browser. After that a mutual authentication between MU_i and the cloud service provider CS_j is done by the user mobile App or web browser [169], [194]. Both MU_i and CS_j need to go through a secure mutual authentication process that should support the basic security requirements as mentioned in Section 1.3.3.

In MCC environment, we have the following issues:

- Mobile devices are connected to the mobile networks via the base stations (BTS) that establish and control the connections and functional interfaces between the networks and mobile devices.
- Mobile users' requests and information are transmitted to the central processors that are connected to servers providing mobile network services.
- The subscribers' requests are delivered to a cloud through the Internet.
- In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services.

Another simplified network model of mobile cloud computing is shown in Figure 1.7 (adapted from [161]). Mobile cloud computing has two components: 1) cloud computing and

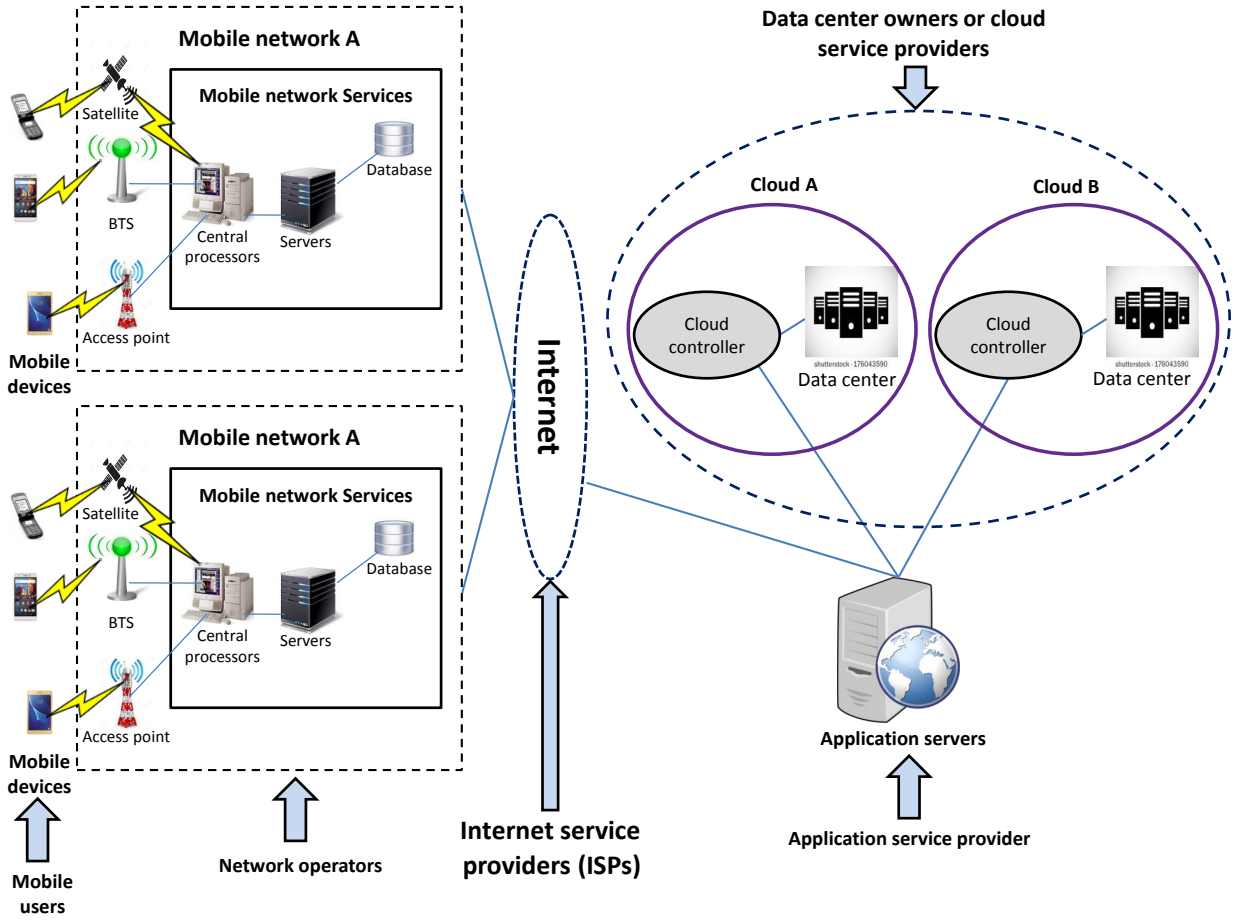


Figure 1.6: An architecture of distributed mobile cloud computing (Source: [169]).

2) mobile computing. Using hotspot, WiFi and GPRS, mobile devices (e.g., laptop, personal digital assistant (PDA) and smartphone) can be connected to a mobile network. Considering that a mobile device may be resource constrained, the task computing and processing of user data is migrated to cloud server side. The service request is sent to cloud via installed app or web browser of mobile device and the cloud controller or cloud management component allocates cloud resources to provide the service [161].

1.3.2 Security requirements for mobile cloud computing

The following are the security requirements for MCC environment:

- **Confidentiality:** The requirement of confidentiality demands to keep mobile user's data absolutely secret among various cloud services. As mobile user's data is transmitted and

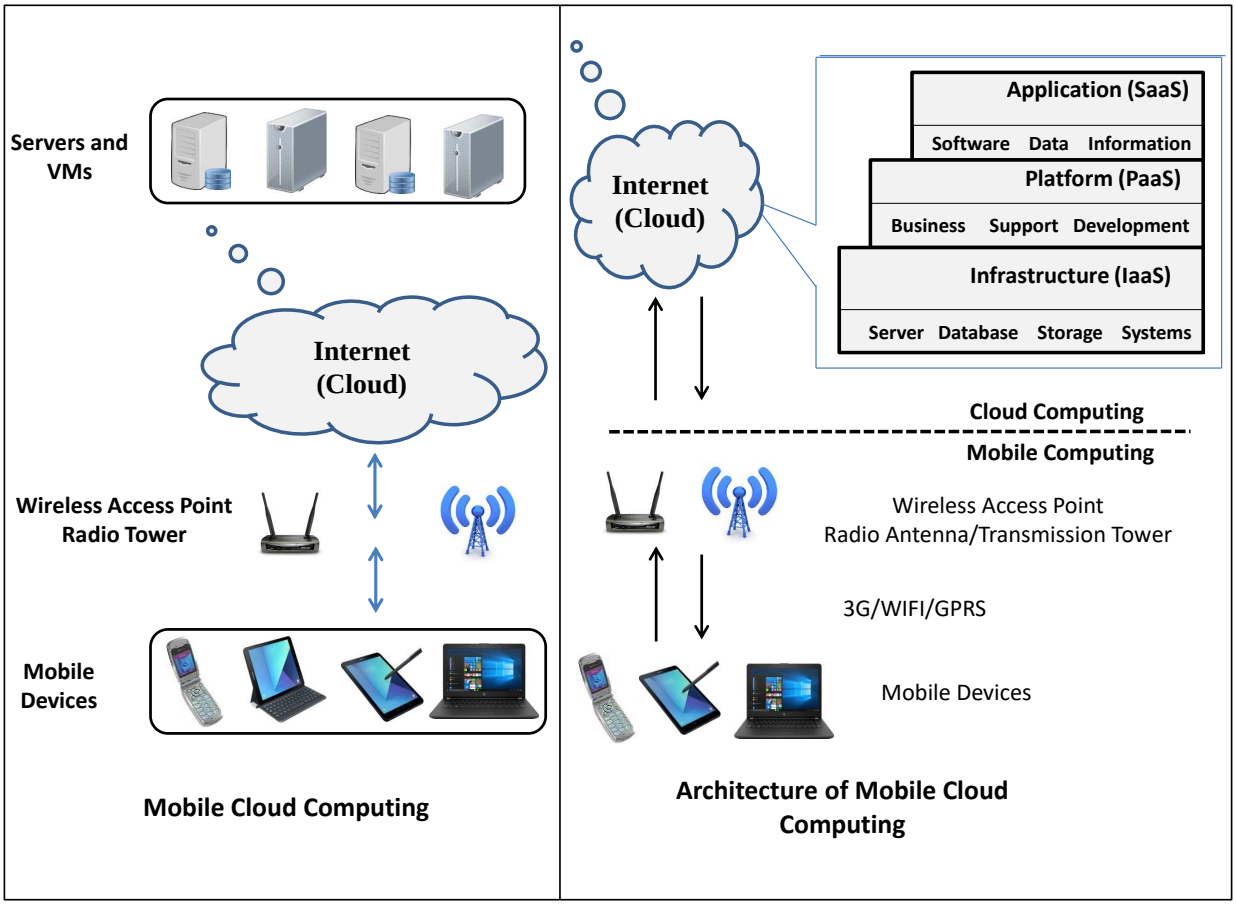


Figure 1.7: A basic diagram of mobile cloud computing (Source: [161]).

received through public channel, data stored and processed through public cloud server, confidentiality is a serious challenge in design of a security protocol.

- **Integrity:** In MCC, data integrity refers to the accuracy and consistency of the mobile user's data by preventing any unauthorized modification of data by adversaries. If data integrity is violated, the cloud service provider that stores/processes wrong data may cause a negative impact on overall business.

- **Availability:** In MCC, as the mobile user's data is stored and processed in cloud servers, it is essential that, whenever in demand, the services must be available to mobile users. The requirement of availability must also prevent DoS attack.

- **Authentication and access control:** Before communication of any data or service, mobile user and the cloud servers must execute mutual authentication for establishment

of mutually shared secret session keys. Access control will provide necessary permission for an access of the required cloud resources to the respective user(s).

- **Privacy:** The security objectives such as confidentiality, integrity and authentication persuade the privacy and these objectives preserve the privacy directly or indirectly of the cloud service users in mobile devices.

1.3.3 Security challenges for mobile cloud computing

The MCC exploits various technologies such as (i) partitioning, (ii) offloading, (iii) virtualization, (iv) outsourced storage, and (vi) mobile-cloud based application. As a result, MCC adopts several new security challenges along with traditional challenges. Figure 1.8 shows major security and privacy challenges in MCC. We present the list of potential security and privacy challenges within MCC, which are as follows [147].

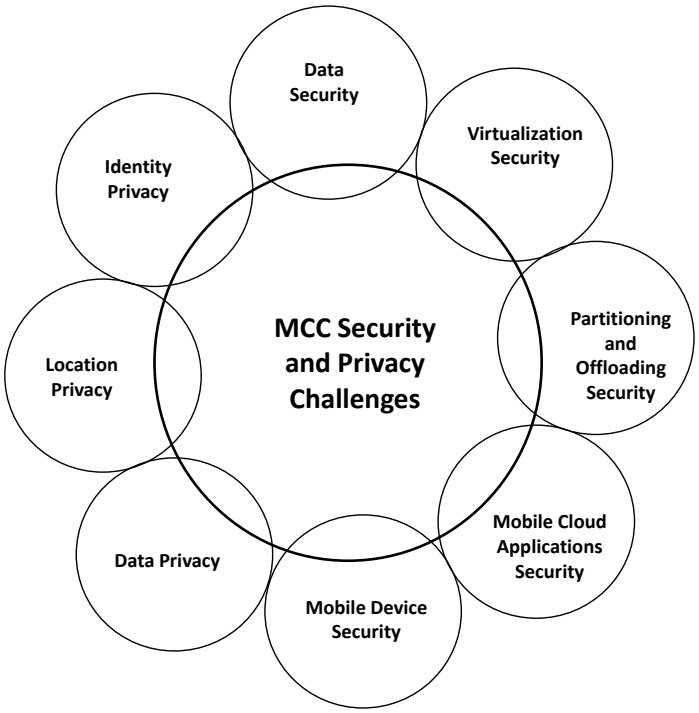


Figure 1.8: Major security and privacy challenges of mobile cloud computing (Source: [147]).

- **Privacy challenges:** One of the most serious challenges of MCC is to provide user's data privacy. Intrinsically, the mobile user's data are stored in various cloud servers

which are located at different distant geographical locations, processed and maintained by various cloud service providers. The mobile data has to shift from mobile device to various cloud servers via public network communications. Naturally, these distributed cloud communications, storage, and processing throw a big challenge towards the mobile user's data confidentiality and privacy.

- **Mobile cloud applications security challenges:** The cloud based mobile applications are susceptible to various kinds of security attacks which may lead to violate the integrity and confidentiality properties of mobile user's data and applications. In addition, the attacker can also install various malwares including virus, worm, Trojan, rootkit and botnet [159], [160], [162].

- **Mobile devices security challenges:** Loss of theft of user mobile device is a very common phenomenon. After obtaining user mobile device, an adversary might try to illegally access various data or application installed or stored in it to guess user's secret credentials and launch various security attacks into the system.

- **Data security challenges:** Mobile user's data is stored and processed at cloud server side, which is at service provider's end. As a result, data loss, theft, privacy breach or any type of data misuse is a matter of concern in MCC.

- **Virtualization security challenges:** In MCC, virtualization techniques are created by cloud service providers to offer cloud services to the mobile users. These techniques require pre-installing a virtual machine (VM) image of mobile device, and then its tasks are offloaded into the VM. However, virtualization techniques when applied to MCC generate several security challenges including communication security within the virtualized environment, VM to VM attack, security challenges regarding confidentiality of data, and unauthorized access [180].

- **Partitioning and offloading security challenges:** In MCC environment, the offloading process is executed at cloud server end with the access of cloud through public channel, and the mobile user has no control over it. As a result, data integrity, privacy, and confidentiality are at risk, and the adversary might try to make an unauthorized access to mobile users data for various mal-intentions.

1.3.4 Functionality requirements for mobile cloud computing

A user authentication scheme for distributed mobile cloud computing environment should satisfy the following functionality requirements:

- A trusted third party (i.e., identity provider (IdP) or smart card generator (SCG)) should not be involved during user login process. However, during registration phase, both mobile user MU_i and cloud server CS_j should register to trusted third party.
- The authentication process should avoid computationally expensive operations for the user mobile device. Also, storage requirement in user mobile should be kept as minimum as possible.
- As the mobile user might use resource constrained mobile device, mutual authentication between MU_i and CS_j should use lightweight cryptographic operations.
- Security protocol for an MCC environment should also support high scalability.

1.3.5 Applications of mobile cloud computing

In this subsection, we briefly mention some applications related to the mobile cloud computing:

- **Mobile commerce (M-commerce):** It allows business models for commerce using mobile devices. Some examples include mobile finance, mobile advertisement and mobile shopping.
- **Mobile learning (M-learning):** It combines e-learning and mobility. Cloud-based m-learning can solve limitations of low transmission rate, limited educational resources, and high cost of devices/network.
- **Mobile healthcare (M-healthcare):** It is to minimize the limitations of traditional medical treatment and to provide mobile users with convenient access to resources (e.g., medical records). Mobile Healthcare offers a variety of on-demand services on clouds.
- **Mobile game (M-game):** It generates high potentially market generating revenues for service providers. It can completely offload game engine requiring large computing resource (e.g., graphic rendering) to the server in the cloud, and thereby, it saves energy and increases the game playing time.

1.4 Security issues in telecare medicine information system

Telecare Medicine Information System (TMIS) for health-care delivery service requires information exchange among multiple systems, where different types of users with different access privileges are involved. In TMIS, users generally communicate via public channels. Considering the privacy of the patients, secure and authenticated access to the medical data located at the medical severs are required. 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. 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.

1.4.1 Network model of telecare medicine information system

TMIS contains one or more medical servers that keep electronic medical records (EMRs) of registered users, and these provide access to the EMRs via the Internet to the users, physicians, health educators, hospitals, public health organizations and homecare service providers [139]. Figure 1.9 provides a basic network architecture of an TMIS with a single server. To prevent unauthorized and illegal access to the patient's private medical data in the medical servers, authenticated, protected and secure access to the medical data are needed [176]. For this purpose, the medical server should remotely authenticate users with their smart cards and provide the requested access to the corresponding medical records after successful authentication [139].

1.4.2 Security requirements in telecare medicine information system

As per the security requirements of TMIS, an authentication protocol should prevent the following security attacks [139]:

- **Denial-of-Service (DoS) attack:** Through this attack, an attacker may deny the services between the TMIS users/patients and the medical servers [174].

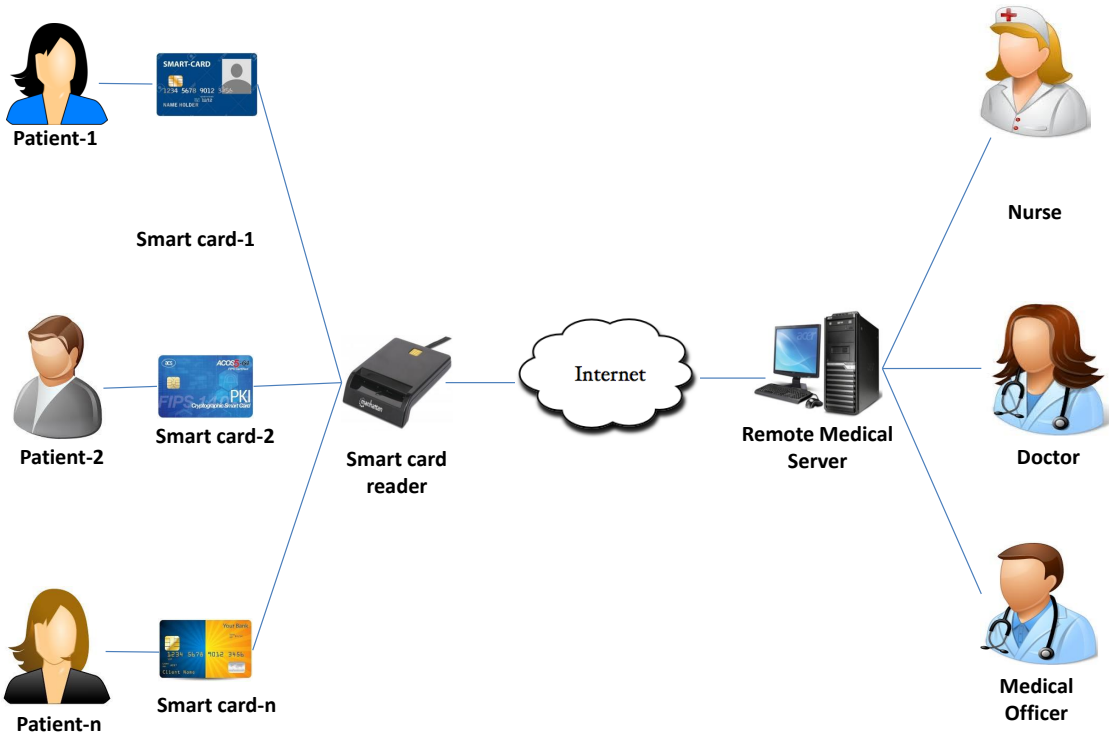


Figure 1.9: Network architecture of TMIS with single server.

- **Password guessing attack:** In this attack, an attacker tries to guess the password of a patient either in online or offline mode through the transmitted messages and some stored secret information in the system [150]. Generally, online guessing attack is done through intercepting communicated messages, while offline guessing attack is done by obtaining stored smart card data, mobile data, and server data. In a user authentication scheme, such kind of attack should be protected.

- **Privileged insider attack:** In this kind of attack, the system manager or a privileged-insider of the medical server may attempt to know the login details including password of any genuine patient. In user authentication scheme, it should be taken care that the login details of any user should not be compromised by any privileged-insider or server administrator.

- **TMIS server impersonation attack:** Through this attack, a compromised TMIS server may try to fabricate a fake login response message to a user to convince that it

is a legal message. A user authentication protocol for TMIS should prevent this kind of attack [73], [120].

- **User impersonation attack:** This attack enables an attacker to impersonate another legal user by generating a fake login message in order to login into the TMIS.
- **Man-in-the-middle attack:** In a man-in-the-middle attack, an attacker may intercept the messages during transmission and can change/delete/modify the content of the messages delivered to the recipient. This type of attack should be protected by a user authentication scheme.
- **Replay attack:** A replay attack is an offensive action in which an attacker tries to deceive another legitimate user in the network through the reuse of information obtained in a protocol. Thus, this attack indicates an attempt by an unauthorized third party to record the exchanged messages during transmissions. In TMIS, replay attacks should be prevented by using timestamp and random numbers that are embedded in the transmitted messages [137].
- **Stolen-verifier attack:** This type of attack happens when the TMIS server stores any verifier/password table for verification of patient's authenticity. The attacker can steal any patient's login identity or password from the stolen-verifier table. In this attack, the adversary being either a privileged user or an external party can modify the passwords or the patient verification tables stored in the medical server's database [68].
- **Spoofing attack:** An attacker makes an interrupt by changing the routing information and keys in TMIS system under such type of attack [187].
- **User anonymity:** An adversary must not be able to compute patient's original identity from any intercepted message. As a patient's identity is usually short and has a certain format, an adversary may find the ID within polynomial time by executing exhaustive guessing attack [148].

Figure 1.10 shows major security and privacy preservation requirements in TMIS [176].

1.4.3 Functionality requirements in telecare medicine information system

The following functionality requirements are essential in an TMIS environment:

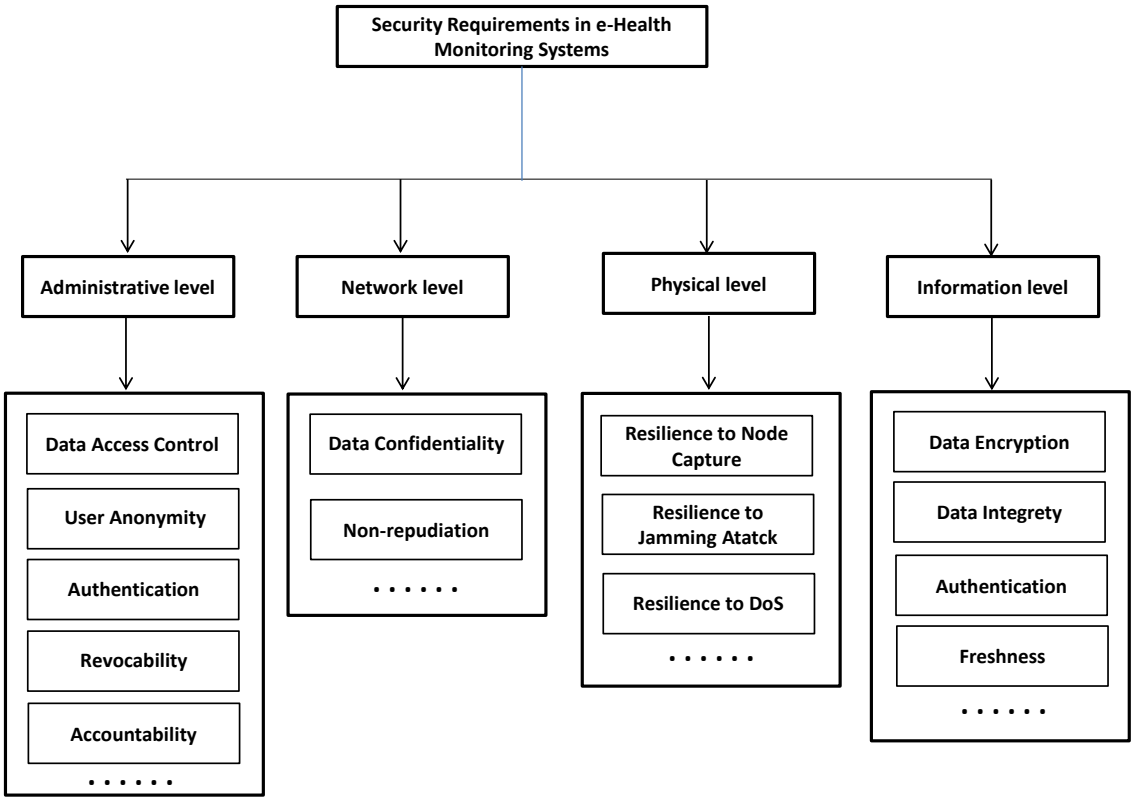


Figure 1.10: Overview of security attacks in TMIS (Source: [176]).

- **Fine-grained data access control:** The basic objectives of practical telecare-medical and e-healthcare systems 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.

- **Mutual authentication between user and TMIS server:** After run of the protocol,

the TMIS server should believe that the remote user is a legitimate registered client. The user also believes that the communicating party is the authorized TMIS server.

- **TMIS server not knowing password and/or biometric:** The medical server should not have any information about the registered user's password and personal biometrics in case of a biometric-based user authentication scheme. This is extremely required because several users may apply the same password to access different servers in the real-life applications. As a result, if a privileged insider of the medical server knows the password or biometrics of a user, he/she may impersonate the user for accessing the services from other medical servers.

- **Freedom of password and/or biometric update:** A user should be allowed to change/update freely his/her password as well as biometric template without contacting the TMIS server. The server should be unaware of the change of the user's password and biometric template.

- **Efficiency:** A user authentication protocol for TMIS should be designed in such a way that it requires the minimum number of message/packet transmissions during the login and authentication phases for user mobile devices/smart card. In addition, it should be also computationally efficient, and the storage requirement in each user's mobile device or smart card.

1.5 Motivation and objective of the work

Single server authentication mechanism is not suitable for a multi-server environment as a user needs to register with various servers with different credentials. The key feature of multi-server based protocols is one-time registration. The involvement of central authority in mutual authentication may be a bottleneck for a large-scale network. Moreover, an adversary may be able to compromise servers. Most of the recently proposed multi-server authentication protocols suffer from two serious drawbacks. First, they failed to achieve several security properties while maintaining the best performance level. Second, many of the protocols incur more performance overheads as they involve the registration center (RC) in login and authentication processes. This motivates us to design a new lightweight, robust and secure scheme in multi server environment. In addition, a multi-server authentication scheme also faces several challenges. The designed scheme should avoid multiple registrations for the individual servers. The servers or the RC must not store any verification or password table in order to avoid

the stolen verifier attack. Furthermore, the designed scheme should support high scalability with dynamic servers and users joining and revocation. In short, the proposed authentication scheme for multi-server environment should fulfill the security and functionality requirements as mentioned in Sections 1.1.2 and 1.1.3.

Due to rapid development of Internet technology and other wireless communications, people have started getting benefits of e-healthcare systems. In some situations, sharing of the patient information in a protected online environment with a group of medical professionals is very much essential, and for these types of treatments where multiple professionals are involved, crowdsourcing Internet of Things (IoT) in e-healthcare services is required. IoT has an enormous threat to security and privacy due to its heterogeneous and dynamic nature. Authentication is one of the most challenging security requirements in IoT environment, where a user (external party) can directly access information from the devices provided that the mutual authentication between user and IoT devices happens.

The recent proliferation of mobile devices such as smartphones and wearable devices has given rise to crowdsourcing IoT applications. Data collected by the mobile devices with small or big volumes can be further processed, analyzed and mined to support multifarious promising services with intelligence. In e-healthcare applications data collected by the mobile devices are stored in various medical servers. The information is then accessed from the medical servers for monitoring and diagnosing a patient by a legal user (for example, a doctor). Unfortunately, ever-growing use of the Internet offers malicious users and attackers ample opportunity to gain unauthorized illegal access of medical data by exploiting various kinds of network and information attacks. Most of the existing schemes cannot properly fulfill the security and functionality requirements mentioned in Sections 1.2.2 and 1.2.2. To protect important and private medical information, design of a proper security protocol for crowdsourcing in e-healthcare services requires more attention from the researchers. This necessitates maneuvering a wide range of remote user authentication protocols for providing access of the services to authorize users only.

As user's mobile device generally operates through battery limited equipment, mobile user authentication mechanism should consume less computation, communication and storage costs. However, most of the existing authentication schemes for mobile cloud computing environments are based on resource consuming cryptosystems, such as bilinear pairing. This necessitates the design of an efficient mobile user authentication scheme that could avoid such cryptosystems without degrading overall security of the system. A careful study on existing authentication schemes under mobile cloud computing environment reveals that most of those

schemes have security flaws as mentioned in Sections 1.3.3 and 1.3.4. Hence, design of more secure and efficient authentication scheme is needed in this domain.

In TMIS, different types of users send different types of data requests to the medical servers. The users of this system are of heterogeneous types in nature that include patients, doctors, health staffs, insurance persons, and medical researchers. The access privilege of the users, 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 to provide proper user authentication. But, most of them do not deliver 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. The proposed authentication protocol with fine-grained access control should fulfill the security and functionality requirements as mentioned in Sections 1.4.2 and 1.4.3.

1.6 Summary of contributions

The contributions of the thesis are summarized in the following subsections.

1.6.1 Biometric-based user authentication for multi-server environment

The first contribution of the thesis is to design a new authentication scheme for multi-server environment. In the proposed scheme, we use the Chebyshev chaotic map along with biometric and password verification with authorization and access to various application servers. We only use the Chebyshev chaotic map, cryptographic hash function and symmetric key encryption/decryption in the proposed scheme. In our scheme, at the time of authentication, a session key is established between the respective server and user without involving the RC. This significantly reduces the communication cost and makes the authentication process

faster and efficient. Our scheme provides strong authentication, and also supports biometrics and password change phase by a legitimate user at any time locally, and also the dynamic server addition phase. We perform the formal security verification using the broadly-accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool to show that the presented scheme is secure. In addition, we use the formal security analysis using the Burrows-Abadi-Needham (BAN) logic along with random oracle model and prove that our scheme is secure against different known attacks. High security, and significantly low computation and communication costs make our scheme is very suitable for multi-server environment as compared to other existing related schemes.

1.6.2 Biometric-based anonymous user authentication for crowdsourcing IoT environment

In the second contribution of the thesis, we aim to propose a new secure three-factor user authentication protocol based on the extended chaotic maps. The three factors involved in the proposed scheme are: 1) smart card, 2) password and 3) personal biometrics. As the proposed scheme avoids computationally expensive elliptic curve point multiplication or modular exponentiation operation, it is lightweight and efficient. E-healthcare service is one of the important services for the crowdsourcing IoT applications that facilitates remote access or storage of medical server data to the authorized users via wireless communication. As wireless communication is also susceptible to various kinds of threats and attacks, remote user authentication is highly essential for a hazard-free use of these services. The formal security verification using the widely-accepted verification tool, called the ProVerif 1.93, shows that the presented scheme is secure. In addition, we present the formal security analysis using the both widely-accepted Real-Or-Random (ROR) model and Burrows-Abadi-Needham (BAN) logic. With the combination of high security, and appreciably low communication and computational overheads, our scheme is very much practical for battery limited devices for the healthcare applications as compared to other existing related schemes.

1.6.3 Biometric-based anonymous user authentication for mobile cloud computing services environment

The third contribution of the thesis involves designing of mobile user authentication scheme for a distributed mobile cloud computing environment, which supports secure key exchange, and user anonymity and untraceability properties. As user's mobile device generally operates

through battery limited equipments, mobile user authentication mechanism should consume minimum possible computation, communication and storage costs. Since the proposed scheme does not exploit any resource constrained cryptosystem, it has the lowest computation cost in compare to existing related schemes. No trusted third party, like IdP, SCG or RC, is involved in user login and authentication phases. This reduces overall communication and computation time of the proposed scheme. The proposed scheme has the ability to resist various known attacks, which are evident through the rigorous formal security proof through random oracle model and BAN logic, the formal security verification using theProVerif 1.93 simulation tool as well as through informal security analysis.

1.6.4 Fine-gained access control with user authentication for TMIS environment

The fourth and final contribution is on development of fine-grained data access control with an efficient authentication mechanism in TMIS environment. 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 along with biometric based user authentication scheme, specially tailored for TMIS. The proposed scheme supports user anonymity, forward secrecy, and efficient password change without contacting the remote server. We present the formal security analysis using both the widely-accepted Real-Or-Random (ROR) model and BAN logic. In addition, the proposed is superior with respect to communication and computation costs as compared to other related schemes proposed in TMIS. Moreover, better trade-off among security and functionality features, and communication and computation costs makes the proposed scheme suitable and practical for telecare medicine system environment as compared to other existing related schemes. To the best of our knowledge, this work is the first one to realize distributed fine-grained data access control with authentication for TMIS.

1.7 Organization of the thesis

The organization of the thesis is as follows.

In **Chapter 1**, we discussed various security requirements and functionlity requirements of the application areas like multi-server environment, crowdsourcing IoT, mobile cloud computing and TMIS. We also discussed on their network models and architectures. We then

addressed the motivation and objective of the research work. We also summarize the contributions of the research work presented in this thesis.

In **Chapter 2**, we discuss the mathematical preliminaries used in the thesis. We briefly present the fundamentals of biometrics verification including biohashing and fuzzy extractor. We then discuss on Chebyshev polynomial and chaotic map, elliptic curve and its properties. Next, we discuss on fundamentals of bilinear pairing and attribute-based encryption. Finally, we discuss the BAN logic and its properties.

In **Chapter 3**, we give an overview of the related works on user authentication and access control in multiserver environment, user authentication on crowdsourcing IoT, user authentication for mobile cloud computing services and fine-gained access control with user authentication for TMIS.

In **Chapter 4**, we design a new lightweight, robust and secure user authentication scheme in multi server environment. In the proposed scheme, at the time of authentication, session key is established between the respective server and user without involving the RC. The proposed scheme supports the essential security and functionality features needed for a multi-server environment. Compared to all recent schemes, the proposed scheme incurs much low computation and communication overheads.

In **Chapter 5**, we present a three-factor, extended chaotic map based secure and efficient remote user authentication scheme for crowdsourcing IoT environment. The proposed scheme has low computation and communication costs as compared to those for the existing related schemes. We also introduce an efficient mechanism for revocation of lost smart card of a legitimate user.

In **Chapter 6**, we propose a new secure and lightweight mobile user authentication scheme for mobile cloud computing. The scheme is bassed on cryptographic hash, bitwise XOR and fuzzy extractor functions only. The proposed scheme supports secure key exchange, and user anonymity and untraceability properties. No trusted third party like registration center (RC) is involved during the user login and authentication phases. As compared to existing related schemes, the scheme has the low communication cost.

In **Chapter 7**, we propose a novel fine-gained access control scheme with user authentication for TMIS. The proposed scheme provides group-based user authentication depending on the access rights provided for the genuine users. The proposed scheme supports user anonymity, forward secrecy, and efficient password change without contacting the remote server.

Finally, in **Chapter 8** we summarize the work done, highlight the contribution and suggest

some directions for possible future research work.

Chapter 2

Mathematical Preliminaries

In this chapter, we discuss some fundamental mathematical preliminaries, which are applied to design and analyze the proposed schemes in Chapters 4–7. First, we discuss on the properties of one-way hash function in Section 2.1. In Section 2.2, we discuss the fundamental concepts of biometrics verification using biohashing and fuzzy extractor functions. In Section 2.3, we briefly describe Chebyshev polynomial and chaotic map along with chaotic map-based discrete logarithm problem. Elliptic curve and its properties are discussed in Section 2.4. In Section 2.5, we discuss on bilinear map, bilinear pairing and attribute based encryption techniques. We use the Burrows-Abadi-Needham (BAN) logic to prove the mutual authentication of the proposed protocols. Finally, in this chapter, we discuss basic notations and logical postulates of BAN logic in Section 2.6.

2.1 One-way cryptographic hash function

A cryptographic hash function is an algorithm which accepts a variable length block of data as input and produces a fixed-size bit string as output, known as cryptographic hash value or message digest. Hash function can be applied to a large set of inputs which will produce outputs that are evenly distributed, and apparently random. A change to any bit or bits in input data results, with high probability, in a change to the hash value. The hash functions are often used to determine whether the data in transit between two communicating entities in the network have changed or not. Thus, the hash function is used to provide data integrity.

Mathematically, a one-way cryptographic hash function $h : \{0,1\}^* \rightarrow \{0,1\}^l$ takes an arbitrary-length input $x \in \{0,1\}^*$, and produces a fixed-length (say, l-bits) output $h(x) \in \{0,1\}^l$, called the message digest or hash value. The hash function may be the fingerprint of

a file, a message, or other data blocks, and has the following attributes [189], [191].

- h can be applied to a data block of all sizes.
- For any given input x, the message digest $h(x)$ is easy to operate, enabling easy implementation in software and hardware.
- The output length of the message digest $h(x)$ is fixed.
- Deriving the input x from the given hash value $y = h(x)$ and the given hash function $h(\cdot)$ is computationally infeasible. This property is called the *one-way* property or *preimage resistance* property.
- For any given input x, finding any other input $y \neq x$ so that $h(y) = h(x)$ is computationally infeasible. This property is referred to as *weak-collision resistant* property or *second preimage resistance* property.
- Finding a pair of inputs (x, y), with $x \neq y$, so that $h(x) = h(y)$ is computationally infeasible. This property is referred to as *strong-collision resistant* property.

The formal definition of a one-way hash function $h(\cdot)$ is given as follows [175], [190].

Definition 2.1 (One-way hash function)**.** *A one-way collision-resistant hash function* $h : \{0,1\}^* \rightarrow \{0,1\}^l$ *is a deterministic algorithm that takes an input as an arbitrary length binary string* $x \in \{0,1\}^*$ *and outputs a binary string* $h(x) \in \{0,1\}^l$ *of fixed-length* l*. The formalization of an adversary* $\mathcal{A}$*'s advantage in finding collision is as follows.*

$$Adv_{\mathcal{A}}^{HASH}(t) = Pr[(x, x') \leftarrow_R \mathcal{A} : x \neq x' \text{ and } h(x) = h(x')],$$

where $Pr[X]$ *denotes the probability of an event* X*, and* $(x, x') \leftarrow_R \mathcal{A}$ *denotes the pair* (x, x') *is randomly selected by* $\mathcal{A}$*. In this case, the adversary* $\mathcal{A}$ *is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary* $\mathcal{A}$ *with the execution time* t*. An* (ϵ, t)*-adversary* $\mathcal{A}$ *attacking the collision resistance of* $h(\cdot)$ *means that* $\mathcal{A}$*'s the runtime is at most* t *and that* $Adv_{(A)}^{HASH}(t) \leq \epsilon$*.*

There are many applications of the hash functions, for examples, in the field of cryptology and information security, notably in digital signatures, message authentication codes (MACs), and other forms of authentication. Thus, a hash function becomes the basis of many cryptographic protocols. One fundamental property of a hash function is that its outputs are very

sensitive to small perturbations in its inputs. For example, SHA-1 is a secure hash algorithm [6]. Quark [20], [21] is a family of cryptographic hash functions proposed recently, which is designed for extremely resource-constrained environments like wireless sensor networks (WSNs) and radio-frequency identification (RFID) tags. Quark can be used as pseudo-random function (PRF), message authentication code (MAC), pseudo-random number generator (PRNG), and key derivation function as other hash functions are also used for these purposes. Thus, the lightweight hash function, Quark is more computationally efficient as compared to SHA-1. For better security, SHA-256 algorithm can be also used [61].

2.2 Biometrics verification

In this section, we discuss the following two techniques, which can be applied for biometric verification purpose.

2.2.1 Biohashing

User biometric data are inherently associated with different types of noises and uncertainties. Repeated acquisitions of biometric impressions of the same person might have variations up to some extent. As a result, use of traditional cryptographic hash function (for example, SHA-1 hash function [6]) over the biometric data may cause a significant change in the result. Biohashing is a certain class of specially formulated hash function that are invariant to these subtle changes of input biometrics. In order to resolve the issue of high false rejection problem, a two-factor authenticator based on iterated inner products between tokenized pseudo-random number and the user specific fingerprint features was proposed [110]. Biohashing is used to map a user's biometric features onto user-specific random vectors to generate a code, called the BioCode and then discretizes the projection co-efficients into zero or one [110], [134].

Biometric keys, such as iris, fingerprint and palmprint, are now increasingly used in several authentication protocols due to their uniqueness property [153], [233]. 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 [55]. Jain et al [101] reported that a biometric verification system may suffer from (i) false match or false accept problem and (ii) false nonmatch or false reject problem. They also reported the state-ofthe-art error rates of various common biometric traits [101].

2.2.2 Fuzzy extractor

Recently, the fuzzy extractor method has been used effectively in extracting biometric key from a given user biometric input [153], [186]. 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 [154].

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 $\mathcal{B}_2$, where $\mathbb{Z}^+$ represents the set of all positive integers.
- 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 $\lambda = m - 2\log(\frac{1}{\delta}) + O(1)$ [69], [153].
- *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 $\alpha = Rep(\mathcal{B}', \beta)$.

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.

2.3 Chebyshev polynomial and chaotic map

In this section, we discuss Chebyshev chaotic maps and the following two intractable problems.

Definition 2.2 ([118]). *A Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n is defined using the following recurrence relation:*

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

The first few Chebyshev polynomials are given below:

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned}$$

Definition 2.3 ([29], [118]). *The Chebyshev polynomials can be alternatively defined as follows:*

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

where the trigonometric functions $cos(x)$ and $arccos(x)$ defined as $cos : R \rightarrow [-1, 1]$ and $arccos : [-1, 1] \rightarrow [0, \pi]$, and the $cos(x)$ function has period 2π.

Definition 2.4 (Chaotic property [118]). *The interval $[-1, 1]$ is invariant under the action of the map T_p:$[-1, 1] \rightarrow [-1, 1]$. Thus, the Chebyshev polynomial restricted to the interval [-1,1] is the well-known chaotic map for all $p > 1$, which has a unique absolutely continuous invariant density*

$$\mu(x)dx = \frac{dx}{\pi\sqrt{1 - x^2}},$$

with positive Lyapunov exponent $\lambda = lnp$, where $ln = \log_e$.

For $p = 2$, the Chebyshev map reduces to the well-known logistic map.

Definition 2.5 (Semi-group property [118], [232]). *Let r and s be two positive integers and $x \in [-1, 1]$. The semi-group property of the Chebyshev polynomials can be defined as follows:*

$$\begin{aligned} T_r(T_s(x)) &= cos(r \cdot cos^{-1}(cos(s \cdot cos^{-1}(x)))) \\ &= cos(rs \cdot cos^{-1}(x)) \\ &= T_{sr}(x) \\ &= T_s(T_r(x)). \end{aligned}$$

Bergamo *et al.* [29] described an attack that allows 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

$$s' = \left\{ \frac{arccos(T_s(x)) + 2k\pi}{arccos(x)}, k \in \mathcal{Z} \right\},$$

where $\mathcal{Z}$ is the set of all integers.

Definition 2.6 ([232]). *The semi-group property of the enhanced Chebyshev polynomial holds on the interval $(-\infty, +\infty)$ and is defined as follows:*

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \pmod{p},$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime number. Obviously,

$$T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x)) \pmod{p},$$

where $Z_p^* = \{a | 0 < a < p, \gcd(a, p) = 1\} = \{1, 2, \ldots, p-1\}$.

Definition 2.7 (Chaotic Maps-based Discrete Logarithm Problem **(CMDLP)** [82]). *For any given x and y, it is computationally infeasible to find an integer s such that $T_s(x) = y$. The advantage probability of $\mathcal{A}$ to solve CMDLP is given by*

$$Adv_{\mathcal{A}}^{CMDLP}(t_2) = Pr[\mathcal{A}(x, y) = r : r \in Z_p^*, y = T_r(x) \pmod{p}.$$

Definition 2.8 (Chaotic Maps-based Diffie-Hellman **(CMDH)** problem [82]). *Given a random tuple $\langle x, T_r(x) \pmod{p}, T_s(x) \pmod{p} \rangle$, it is hard for a polynomial time bounded algorithm $\mathcal{A}$ to compute $T_{rs}(x) \pmod{p}$. The probability that $\mathcal{A}$ can find the solution of the CMDHP is defined as follows:*

$$Adv_{\mathcal{A}}^{CMDH} = Pr[\mathcal{A}(x, T_r(x) \pmod{p}, T_s(x) \pmod{p}) = T_{rs}(x) \pmod{p}) = r : r \in Z_p^*].$$

2.4 Elliptic curve and its properties

In this section, we discuss on elliptic curve and its properties. The discussion includes rules for adding points on elliptic curve and the elliptic curve discrete logarithm problem.

2.4.1 Elliptic curve over finite field

Let a and $b \in Z_p$ be two constants, where $Z_p = \{0, 1, \ldots, p-1\}$ and $p > 3$ be a prime number, such that $4a^3 + 27b^2 \neq 0 \pmod{\text{p}}$. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field or Galois field $GF(p)$ is the set $E_p(a, b)$ of solutions $(x, y) \in Z_p \times Z_p$ to the congruence

$$y^2 = x^3 + ax + b \pmod{\text{p}}$$

together with a special point $\mathcal{O}$, called the point at infinity or zero point.

The condition $4a^3 + 27b^2 \neq 0 \pmod{\text{p}}$ is the necessary and sufficient to ensure that the equation $x^3 + ax + b = 0$ has a non-singular solution [151]. Otherwise, if $4a^3 + 27b^2 = 0 \pmod{p}$, the corresponding elliptic curve is called a singular elliptic curve. Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points in $E_p(a, b)$. Then $P + Q = \mathcal{O}$ implies that $x_Q = x_P$ and $y_Q = -y_P$. We have $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E_p(a, b)$. More precisely, a well-known theorem due to Hasse asserts that the number of points on $E_p(a, b)$, which is denoted by $\#E$, satisfies the following inequality [189]:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}.$$

In other words, an elliptic curve $E_p(a, b)$ over Z_p has roughly p points on it. In addition, $E_p(a, b)$ forms an abelian group or commutative group under addition modulo p operation.

2.4.2 Point addition on elliptic curve over finite field

Let G be the base point on $E_p(a, b)$ whose order be n, that is, $nG = G + G + \ldots + G$ (n times) $= \mathcal{O}$. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points on elliptic curve $y^2 = x^3 + ax + b \pmod{\text{p}}$, with $P \neq -Q$, then $R = (x_R, y_R) = P + Q$ is computed as follows [117], [189]:

$$\begin{aligned} x_R &= (\lambda^2 - x_P - x_Q) \pmod{\text{p}}, \\ y_R &= (\lambda(x_P - x_R) - y_P) \pmod{\text{p}}, \\ \text{where } \lambda &= \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} \pmod{p}, \text{if } P \neq Q \\ \frac{3{x_P}^2 + a}{2y_P} \pmod{p}, \text{if } P = Q. \end{cases} \end{aligned}$$

2.4.3 Scalar multiplication on elliptic curve over finite field

In elliptic curve cryptography, multiplication is defined as repeated additions. For example, if $P \in E_p(a, b)$, then $5P$ is computed as $5P = P + P + P + P + P$.

Example 2.2: Consider two points $P = (11, 3)$ and $Q = (9, 7)$ in the elliptic curve $E_{23}(1, 1)$ [52]. All the points of $E_{23}(1, 1)$ are shown in Table 2.1 as well as in Figure 2.1.

Table 2.1: Points over the elliptic curve $E_{23}(1, 1)$ (Source: [52]).

$(0, 1)$	$(6, 4)$	$(12, 19)$	$(0, 22)$	$(6, 19)$	$(13, 7)$	$(1, 7)$	$(7, 11)$	$(13, 16)$
$(1, 16)$	$(7, 12)$	$(17, 3)$	$(3, 10)$	$(9, 7)$	$(17, 20)$	$(3, 13)$	$(9, 16)$	$(18, 3)$
$(4, 0)$	$(11, 3)$	$(18, 20)$	$(5, 4)$	$(11, 20)$	$(19, 5)$	$(5, 19)$	$(12, 4)$	$(19, 18)$

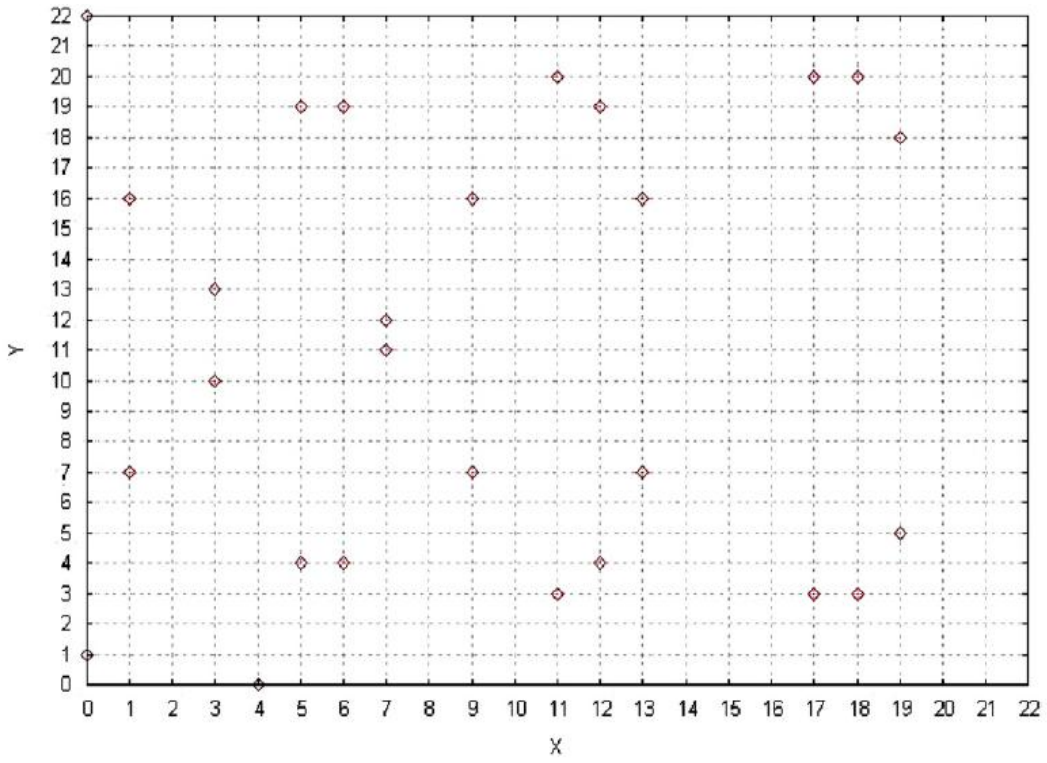


Figure 2.1: Example of elliptic curve in case of $y^2 = x^3 + x + 1 \pmod{23}$ (Source: [52])

Consider two points $P = (11, 3)$ and $Q = (9, 7)$ in $E_{23}(1, 1)$. In this case, $P \neq -Q$. In order to compute $R = P + Q = (x_R, y_R)$, we first compute λ as

$$\lambda = \frac{7 - 3}{9 - 11} \pmod{23} = 21.$$

Thus, x_R and y_R are derived as

$$x_R = (21^2 - 11 - 9)(\text{mod } 23) = 7,$$
$$y_R = (21(11 - 7) - 3)(\text{mod } 23) = 12.$$

As a result, $P + Q = (7, 12)$.
To calculate $2P$, we must first derive λ as follows:

$$\lambda = \frac{3(11^2) + 1}{2 \times 3} (\text{mod } 23) = 7.$$

Hence, $R = P + P = (x_R, y_R)$ is computed as

$$x_R = (7^2 - 11 - 11)(\text{mod } 23) = 4,$$
$$y_R = (7(11 - 4) - 3)(\text{mod } 23) = 0,$$

and, thus $2P = (4, 0)$.

2.4.4 Elliptic curve discrete logarithm problem

Let $E_p(a, b)$ be an elliptic curve modulo a prime p. Given two points $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$, for some positive integer k, where $Q = kP$ represents the point P on elliptic curve $E_p(a, b)$ be added to itself k times. Then, the elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and Q. It is computationally easy to calculate Q given k and P, but it is computationally infeasible to determine k given Q and P, when the prime p is large [189].

2.5 Bilinear pairing and attribute-based encryption

In this section, we discuss the bilinear pairing along with decisional bilinear Diffie-Hellman (BDH) assumption. We then discuss on Key-Policy Attribute-Based Encryption (KP-ABE) [77].

2.5.1 Bilinear pairing and its computational assumptions

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear map is an injective function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following three properties:

- ***Bilinearity :*** For all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$, a, $b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
- ***Non-degeneracy:*** $e(g_1, g_2) \neq 1$, 1 is the identity in $\mathbb{G}_T$.
- ***Computability :*** There is an efficient algorithm to compute $e(u, v)$ for each $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

We say that $\mathbb{G}_1$ is a bilinear group if the group operation in $\mathbb{G}_1$ and the bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ are both efficiently computable. It is to be noticed that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$ [77].

Definition 2.9 (**Decisional Bilinear Diffie-Hellman (DBDH) assumption**). *Suppose a challenger chooses $a, b, c, z \in \mathbb{Z}_p$ at random. The DBDH assumption is that no probabilistic polynomial-time algorithm $\mathcal{B}$ is to be able to distinguish the tuple $\langle A = g^a,\ B = g^b,\ C = g^c,\ Z = e(g, g)^{abc}\rangle$ from the tuple $\langle A = g^a, B = g^b, C = g^c, Z = e(g, g)^z\rangle$ with more than a negligible advantage [173]. The advantage is then given by*

$$\left|\ Pr[\mathcal{B}(A, B, C, e(g, g)^{abc}) = 0] - Pr[\mathcal{B}(A, B, C, e(g, g)^z) = 0]\ \right|$$

where the probability is taken over the random choice of the generator g, the random choice of a, b, c, $z \in \mathbb{Z}_p$ [77].

Definition 2.10 (**Decisional Modified Bilinear Diffie-Hellman (DMBDH) assumption**). *Suppose a challenger chooses a, b, c, $z \in \mathbb{Z}_p$ at random. The DMBDH assumption is that no polynomial-time adversary is to be able to distinguish the tuple $\langle A = g^a,\ B = g^b,\ C = g^c,\ Z = e(g, g)^{\frac{ab}{c}}\rangle$ from the tuple $\langle A = g^a,\ B = g^b,\ C = g^c,\ Z = e(g, g)^z\rangle$ with more than a negligible advantage [173].*

2.5.2 Key-policy attribute-based encryption (KP-ABE)

In this section, we discuss the KP-ABE cryptosystem that provides for fine-grained sharing of encrypted data [77]. The process contains four steps, namely 1) setup, 2) encryption, 3) key generation, and 4) decryption.

Let $\mathbb{G}_1$ be a bilinear group of prime order p of size k, and let g be a generator of $\mathbb{G}_1$. In addition, let $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote the bilinear map. Further, assume the Lagrange coefficient $\triangle_{i,S}$ for $i \in \mathbb{Z}_p$ and a set S of elements in $\mathbb{Z}_p$ as $\triangle_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$, where each attribute has a unique element in $\mathbb{Z}_p^*$.

- **Setup:** Universe of attributes is $\mathcal{U} = \{1, 2, \ldots, n\}$ and each attribute $i \in \mathcal{U}$ is chosen along with number t_i uniformly at random from $\mathbb{Z}_p$. In addition, y is chosen uniformly at random in $\mathbb{Z}_{\text{।}}$. The published public parameters PK are

$$T_1 = g^{t_1}, \ldots, T_{|\mathcal{U}|} = g^{t_{|\mathcal{U}|}}, Y = e(g, g)^y,$$

  and the master key MK is $t_1, \ldots, t_{|U|}, y$.

- **Encryption (M, γ, PK):** To encrypt a message $M \in \mathbb{G}_2$ under a set of attributes γ, a random value $s \in \mathbb{Z}_p$ is chosen and the ciphertext is published as:

$$E = (\gamma, E' = MY^s, \{E_i = T_i^s\}_{i \in \gamma}).$$

- **Key Generation $(\mathcal{T}, MK)$:** The algorithm outputs a key that enables the user to decrypt a message encrypted under a set of attributes γ if and only if $\mathcal{T}(\gamma) = 1$. The algorithm proceeds as follows. First, a polynomial q_x is chosen for each node x (including the leaves) in the tree $\mathcal{T}$. These polynomials are chosen in the following way in a top-down manner starting from the root node r.

  For each node x in the tree, the degree d_x of the polynomial q_x is set to be one less than the threshold value k_x of that node, that is, $d_x = k_x - 1$. Now, for the root node r, set $q_r(0) = y$ and d_r number of other points of the polynomial q_r randomly to define it completely. For any other node x, set $q_x(0) = q_{parent(x)}(index(x))$ and choose d_x other points randomly to completely define q_x .

  Once the polynomials have been decided, for each leaf node x, we set the following secret value to the user:

$$D_x = g^{\frac{q_x(0)}{t_i}},$$

  where $i = att(x)$, and $att(x)$ represents the attributes of x. The set of above secret values forms the decryption key D.

- **Decryption(E, D):** Decryption procedure is a recursive algorithm. We first define a recursive algorithm DecryptNode(E, D, x) that takes as input the ciphertext $E = (\gamma, E', \{E_i\}_{i\in\gamma})$, the private key D (we assume the access tree $\mathcal{T}$ is embedded in the private key), and a node x in the tree. It outputs a group element of $\mathbb{G}_2$ or $\perp$.

  Let $i = att(x)$. If the node x is a leaf node, we have:

$$\text{DecryptNode}(E, D, x) = \begin{cases} e(D_x, E_i) = e(g^{\frac{q_x(0)}{t_i}}, g^{s.t_i}) = e(g,g)^{sq_x(0)}, & \text{if } i \in \gamma \\ \perp, \text{ otherwise} \end{cases}$$

2.6 BAN logic and its properties

The BAN logic is widely-used tool for analyzing the security of authentication schemes [34]. BAN is a logic of belief. The intended use of BAN is to analyze authentication protocols by deriving the beliefs that honest principals correctly executing a protocol can come to as a result of the protocol execution.

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

- $P \mid\equiv X$: Principal P believes statement X.
- $P \lhd X$: 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.

In the BAN-logic based analysis, the protocol is first idealized into messages containing assertions, then assumptions are stated, and finally conclusions are inferred based on the assertions in the idealized messages and those assumptions. The rules given below describe the main logical postulates of the BAN logic [34], [193]:

- **Rule 1.** MMR (Message-meaning rule) $\frac{P|\equiv Q\overset{K}{\rightleftharpoons}P, P\triangleleft\langle X\rangle_K}{P|\equiv Q|\sim X}$.
- **Rule 2.** NVR (Nonce-verification rule) $\frac{P|\equiv\#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$.
- **Rule 3.** FCR (Freshness-conjuncatenation rule) $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$.
- **Rule 4.** JR (Jurisdiction rule) $\frac{P|\equiv Q\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$.
- **Rule 5.** AR (Additional inference rules) $\frac{P|\equiv(X,Y)}{P|\equiv X}$, $\frac{P\triangleleft(X,Y)}{P\triangleleft X}$, $\frac{P|\equiv Q|\sim(X,Y)}{P|\equiv Q|\sim X}$, $\frac{P|\equiv Q|\equiv(X,Y)}{P|\equiv Q|\equiv X}$.

The security analysis of an authentication protocol using the BAN logic is done using following four steps [193]:

- **Step 1.** Idealize the protocol.
- **Step 2.** Write assumptions about the initial states.
- **Step 3.** Annotate the protocol. For each message transmission of the form "$P \rightarrow Q : M$" in the protocol, assert that Q received M.
- **Step 4.** Use the logic to derive the beliefs held by protocol principals.

It is worth noting that the BAN logic is mainly used in proving the mutual authentication between two communicating parties in the network.

2.7 Summary

In this chapter, we have reviewed mathematical preliminaries on various topics that have been used in designing different schemes. These include discussion on one way hash function, fuzzy extractor techniques for biometric verification, Chebyshev polynomial and chaotic map based cryptography, elliptic curve cryptography, bilinear pairing, and attribute based encryption. In particular, we have discussed those cryptographic techniques which are useful to design lightweight authentication and access control protocols for various applications in wireless communication in Chapters 4–7.

Chapter 3

Review of Related Works

In this chapter, we give an overview of the related works on user authentication and access control in mult-iserver environment, user authentication on crowdsourcing Internet of Things (IoT), user authentication for mobile cloud computing services and fine-gained access aontrol with user authentication for telecare medicine information system.

3.1 Authentication in multi-server environment

In 1981, Lamport proposed a ground-breaking authentication scheme for a two-party client server environment [123]. Following this pioneering work, several single server authentication schemes have been introduced [84], [94], [121], [195], [214], [220]. Unfortunately, these schemes cannot be efficiently applied in a true distributed system containing multiple servers as the users need to remember a specific password for each server and register with them separately.

Recently, various two-factor and three-factor authentication schemes have been designed for single-server environment. An anonymous two-factor authentication for consumer roaming service was proposed by He *et al.* [86]. Khan *et al.* [114] proposed an efficient chaotic revocable biometrics authentication scheme with privacy-protecting mechanism. Chaudhry *et al.* [47] also proposed an efficient elliptic curve cryptography (ECC)-based biometric authentication scheme.

In past few years, several two-factor authentication and key agreement schemes have been proposed for multi-server environment. Depending on the usage of basic cryptographic techniques, existing authentication schemes on multi-server domain can be classified into several types. First category is based on public-key based authentication schemes. In spite of providing higher security, these schemes involve higher computation overhead, and thus these are

inefficient for battery-limited mobile devices. Second category contains authentication scheme that used ECC scalar point multiplication operation and modular exponentiation operations [83], [88], [104], [106], [115], [153], [230]. Third category is based on multi-server authentication schemes that exploit bilinear pairing operation on ECC [74], [92], [197], [234]. Fourth category is based on authentication and key-exchange protocols based on Chebyshev chaotic map based cryptography [93], [109]. Compared to ECC and RSA based schemes, Chebyshev chaotic map based schemes are more suitable for devices with limited battery life and smaller computation power. Final category is based on lightweight hash-based authentication schemes that use one way hash operations, message authentication code (MAC) and bitwise exclusive-or (XOR) operations for key establishment [51], [107], [129], [132], [232].

In 2013, Yoon and Yoo [230] developed an ECC-based authentication scheme for multi-server environment that uses user biometrics and smart card. Kim *et al.* [115] pointed out some serious security flaws of that scheme, such as password guessing attack. They devised an enhanced version of this scheme to remove its security pitfalls. Unfortunately, as pointed by He [83], both the schemes of Yoon-Yoo and Kim *et al.* are vulnerable to different attacks, specially privileged insider and impersonation attacks. Moreover, as both these two schemes store biometric template in the smart card, if any attacker can obtain the smart card and launch a password guessing attack, he/she might impersonate the server as a genuine user.

Keeping an aim to remedy the security flaws of the earlier related schemes, He and Wang [88] proposed a three-factor multi-server based authentication scheme and also analyzed how the same is resistant to all possible active and passive attacks. However, Odelu *et al.* [153] analyzed that He-Wang's scheme [88] fails to prevent session specific ephemeral secret key leakage attack, and as a consequence, it becomes vulnerable to many serious attacks such as replay attack, impersonation attack, and also fails to support strong user anonymity property.

Table 3.1 contains a summary of literature survey of some state-of-art three-factor user authentication protocols designed for multi-server environment. This table contains various relevant protocols with basic cryptographic technique used for authentication purpose, along with their strengths and weaknesses.

3.2 Authentication in crowdsourcing IoT environment

A detailed survey on IoT and its security aspects can be found in [78], [95], [227]. The key components in the IoT are the sensors and smart objects. Integration of the wireless sensor networks (WSNs) into the IoT environment has been well-studied in the literature [71]. Several

Table 3.1: Existing three factor user authentication protocols in multi-server environment.

Protocol	Cryptographic primitive	Protocol strengths	Protocol weakness
Geng ***et al.*** [74]	Bilinear pairing	1) pioneer scheme introducing user authentication in multiserver environment.	1) high performance overhead. 2) no formal security analysis. 3) no user revocation method.
Yoon ***et al.*** [230]	ECC	1) involves RC in authentcation, 2) exploits EC cryptosystem.	1) cannot resist masquerade, privileged insider, password guessing attacks [115].
Kim ***et al.***[115]	ECC	1) security enhancement over [230], 2) introduces user biometrics in multiserver authentication.	1) lacking user anonymity. 2) incorrect login and password change phase [83].
Hsu ***et al.*** [93]	Chebyshev polynomial	1)light-weight & efficient. Uses enhanced Chaotic maps.	1) lacking user revocation and dynamic server addition.
Chuang ***et al.*** [51]	Hash function	1) efficient and lightweight 2) apts for real time applications.	1) prone to server spoofing, DoS, user impersonation & SK-compromise attacks [144].
Mishra ***et al.*** [144]	Hash function	1) security enhancement over [51], 2) high efficiency, 3) low performance cost.	1) exposed to masquerading, replay, DoS attacks [133]. 2) lacking forward secrecy [201].
Lu ***et al.*** [133]	Hash function	1) provides cryptanalysis over [144], 2) communication is very low, 3) computatioin cost is very low.	1) lacking clock synchronization, impersonation, man-in-middle attack, no forward secrecy [164].
He ***et al.*** [88]	ECC	1) first to use fuzzy extractor based three-factor user authentication in multi-server.	1) involves RC in authentication, 2) lacking user revocation [153], 3) temporary key leakage attack.
Li ***et al.*** [131]	ECC	1) 3-factor authentication with MAC & Fuzzy Extractor 2) reducible to 2-factor scheme	1) high communication cost. 2) authentication process has seven message transmission.
Odelu ***et al.*** [153]	ECC	1) provides cryptanalysis on [88] 2) highly secure and robust.	1) involves RC in authentication. 2) high communication cost. 3) high computation overhead.
Wang ***et al.*** [201]	Hash function	1) provides cryptanalysis on [144], 2) very low computation cost, 3) apts for battery limited device.	1) no user untraceability [167]. 2) no clock synchronization [167]. 3) prone to impersonation attacks.

authentication schemes for WSNs and IoT environment have been proposed [87], [89], [96],

[122], [219].

In recent past, many smart card and password based two factor remote user authentication protocols for e-healthcare applications have been also proposed [126], [127]. Unfortunately, many of them suffer from offline password guessing attack, and lack of user anonymity or smart card stolen smart attack. Once an attacker obtains the smart card of a genuine user, he/she can obtain stored information by power analysis attack [142]. It is quite a common tendency for users to input low entropy and easy-to-remember type identity and password, which gives an opportunity to an attacker to guess passwords in polynomial time using password guessing attack [105], [220].

To eliminate the possibility of chosen plain text attack, the researchers used combination of password and biometric templates along with random nonce as smart card parameters [24], [55]. Application of user biometrics based schemes can impose more security than traditional two-factor password based authentication schemes.

Researchers have used various cryptographic operations for achieving user authentication in e-healthcare service application. These include exponentiation operations of RSA cryptosystem, point multiplication on elliptic curve cryptography, chaotic hash operations using Chebyshev polynomial, and so on. User authentication schemes based on chaotic maps have shown better performance as compared to the traditional cryptographic schemes, especially which use RSA and ECC public key cryptosystems.

Table 3.2 tabulates various recent user authentication protocols designed for e-healthcare service systems in an IoT environment. The table contains the basic cryptography used in the protocols, along with brief description about their security pitfalls and functionality limitatations.

3.3 Authentication in mobile cloud computing environment

Rapid development and implementation of many services in mobile cloud computing necessitate extensive research on security issues [17], [217], [226], [235], [236]. Wang and Wang [207] first attempted to explore the underlying rationales for preserving user privacy property in two-factor authentication schemes. They pointed out that without any public-key techniques, it is quite impossible to come up with a privacy-preserving scheme using only lightweight cryptographic primitives, such as one-way cryptographic hash functions. Wang and Wang

Table 3.2: Existing user authentication protocols designed for healthcare application in IoT environment.

Protocol	CP	Security improvement done by	Security drawbacks and functionality limitations
Hao *et al.* [82]	Chaotic map	Lee *et al.* [126]	Violates contributory property of key agreements. session key can be formed by server side only.
Guo and Chang [80]	Chaotic map	Lee [126]	Violates contributory property of key agreements, malicious server can determine session key in advance.
Awasthi *et al.* [24]	Hash function	Das *et al.* [24]	Lacking user anonymity, incorrect password change, fails to preserve session key secrecy.
Chang *et al.* [39]	Hash function	Das [60]	Design flaws in login and password change phase prone to privileged, man-in-the middle attack.
Lee *et al.* [126]	Chaotic map	Li *et al.* [127]	Service misuse attacks for non-registered users lack of user identity in authentication process.
Li *et al.* [127]	Chaotic map	Roy *et al.* [168]	Design flaws in login and authentication phases design flaws in password change & DoS attack.
Wen *et al.* [72]	Modular exponentiation	Xie [222]	Prone to off-line password guessing attack, and lack of user's anonymity & perfect forward secrecy.
Xie *et al.* [222]	ECC	Xu *et al.* [224]	Vulnerable to the de-synchronization attack, the server has too much storage burden.
Jiang *et al.* [108]	Chaotic map	Mishra *et al.* [146]	Involves security against DoS attack. Design flaws in password change phase.
Das [60]	Hash function	Kim and Lee [116]	Lack of forward secrecy. Vulnerability against lost smart card attacks.
Xu *et al.* [225]	ECC	Roy *et al.* [170]	vulnerable to stolen smart card, offline and online password guessing attack and DoS attack.

CP: the cryptographic primitive used to design the scheme.

[208] also presented some security failures of previous user authentication schemes. Three important suggestions made by them while analyzing those user authentication schemes are as follows [208]: 1) user anonymity preservation with the help of public key techniques, 2) application of fuzzy-verifier to have trade-off between usability and security, and 3) privileged insider attack protection using salt values (i.e., random nonces).

Wang *et al.* [208] suggested some evaluation metrics for designing anonymous two-factor user authentication scheme and also pointed out how to make an acceptable trade-off among usability, security and privacy. Huang *et al.* [97] also suggested that design of password-based

authentication schemes using smart cards requires the importance to elaborate security models along with the formal security analysis. Huang *et al.* [97] also proposed a generic multi-factor authentication scheme that uses user password, smart-card and biometrics as three factors, and observed that a stand-alone authentication in which a user can be authenticated correctly, even if the connection to the remote server is down. Ma *et al.* [135] highlighted three principles for designing more robust user authentication schemes.

Huang *et al.* [94] investigated a systematic approach for authenticating clients using three factors: 1) password, 2) smart card and 3) biometrics. They proposed a generic and secure model to upgrade a two-factor authentication scheme to a three-factor authentication scheme. It is interesting to observe that their conversion not only significantly improved the information assurance at low cost, but also protected client privacy in distributed systems.

Wang and Wang [209] presented a proposal of a new authentication scheme. It meets simplicity, practicability, and strong notions. In addition, they provided an adversary model and a criteria set which can provide a benchmark for the evaluation of current and future two-factor authentication schemes. Moreover, Wang *et al.* [203] pointed out that there are at least seven different attacking scenarios and those attack scenarios may lead to the failure of an authentication scheme in arriving truly two-factor security. They also conducted a large-scale comparative evaluation of 26 representative two-factor schemes, and their results synopsis the request for better measurement when assessing new authentication schemes.

In a distributed mobile cloud computing environment, user mobile devices are quite resource constrained in nature and single sign-on (SSO) technique does not provide a practical solution. Recently, Odelu *et al.* [156], and Gope and Das [75] proposed secure authentication schemes for mobile cloud computing. Odelu *et al.* [156] exploits ECC point multiplication and asymmetric bilinear pairing operation for authentication and key establishment phase, while Gope-Das's scheme [75] is based on hash chain method. In Gope-Das's scheme [75], a mobile subscriber is allowed to obtain the ubiquitous services only up to a specific time-period (say n times), where the access duration strictly depends on the principle that the cloud user has paid for services.

Table 3.3 contains a summary of literature survey of user authentication protocols tailored for mobile cloud computing environment. This table contains various relevant protocols along with their security drawbacks and other functionality limitations.

Table 3.3: Existing user authentication protocols designed for distributed mobile cloud computing environment.

Protocol	Cryptographic primitive	Security limitations	Functionality limitations
Sun *et al.* [192]	ECC	Stolen smart card & replay attacks.	1) no multi-server support. 2) no efficient password change. 3) no efficient biometric change.
Yoon and Yoo [230]	ECC	Ephemeral key leakage, stolen smart card, & privileged insider attacks.	1) high computation. 2) no user anonymity. 3) no formal security proof.
He and Wang [88]	ECC	DoS, impersonation, & strong replay attacks.	1) high computation & communication costs. 2) lack of session key security.
Shen *et al.* [183]	ECC	Lack of session key security, ephemeral secret key leakage attack.	1) high computation. 2) no user anonymity. 3) lack of secure authentication.
Tsai and Lo [194]	Bilinear pairing	User impersonation & server impersonation attacks.	1) high computation. 2) lacks strong user anonymity. 3) no efficient password change.
Li *et al.* [128]	ECC	Offline password guessing, privileged insider & stolen smart card attacks.	1) no efficient password change. 2) no efficient biometric change.
Tseng *et al.* [196]	Bilinear pairing	No user credential's privacy, does not provide user untraceability	1) lacks session key security. 2) lacks strong user anonymity. 3) lacks efficiency in login phase.

3.4 Fine-gained access control with user authentication in TMIS

Over last few years, researchers have developed numerous password based authentication schemes using smart card in the field of TMIS [153], [205], [206]. 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 [85]. These schemes can resist stolen smart card attack, off-line password guessing and impersonation

attacks.

User anonymity preserving scheme with dynamic ID based authentication was proposed for TMIS by Chen *et al.* [49]. A series of enhanced anonymity preserving authentication schemes have been proposed in order to provide better security to the system and also to withstand security drawbacks of the earlier schemes [39], [60], [205].

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 [80]. To enhance its security, functionality and performance related to computation and efficiency, several chaotic map based user authentication schemes with smart card have been proposed [60], [80], [108], [127], [146].

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 [59], [80], [108], [145].

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 [181] and Blakley [30] introduced a tree access structure based cryptographic technique known as secret-sharing schemes. Sahai and Waters proposed Fuzzy Identity-Based Encryption (FIBE) [173] 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 [181], and it is also based on several primitive works of IBE [31], [53].

A much enriched form of ABE, called Key-Policy Attribute-Based Encryption (KP-ABE) was developed by Goyal *et al.* [77] 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 WSN [231]. Yu *et al.*'s scheme exploits the fundamental cryptographic concepts of KP-ABE technique [77]. Chatterjee and Roy then proposed fine-grained user access control scheme with attribute based encryption using elliptic curve cryptography for hierarchical WSN [43]. KP-ABE techniques are also used in various applications like cloud security [124], [172], enterprise class applications [172] and WSN security [43], [118], [172].

Chatterjee and Das [40] 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 [42] and enterprise class applications [41]. In addition, Odelu *et al.* proposed a privacy-preserving three-party authentication suitable for battery-limited mobile devices [154]. 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.

Figure 3.4 shows how fine-grained access control (FGAC) models have evolved in various applications of wireless communications. The table shows the techniques used and the limitations/drawbacks of existing state-of-art protocols for fine-grained data access control.

Table 3.4: Evolution of fine-grained access control (FGAC) schemes and their properties.

Author & Year	**Protocol name**	**Protocol properties (strengths & limitations)**
Shamir [181] (1984)	Identity-based encryption (IBE)	1) Introduces a novel cryptographic concept called IBE. 2) Public keys are an arbitrary string, like email, date etc.
Sahai and Waters [173] (2005)	Fuzzy Identity based encryption (FIBE)	1) Fuzzy IBE can be used for "attribute-based encryption". 2) Exploits identity-based encryption using user biometric. 3) Attributes in user's identity forms user private key set.
Goyal *et al.* [77] (2006)	Key-policy attribute based encryption (KP-ABE)	1) Provides fine-grained sharing of encrypted data. 2) Provides user collusion resilience. 3) Encryption-decryption using attributes & access tree.
Yu *et al.* [77] (2011)	Fine-grained access control for WSNs	1) Realizes distributed FGAC for WSNs. 2) Exploits ABE, tailors, and adapts it for WSNs.
Ruj *et al.* [172] (2011)	Distributed FGAC for WSNs	1) Distributed FGAC with multiple trusted authorities. 2) Supports user join, revocation, and change of access tree.
Chatterjee and Das [40] (2015)	User access control using ABE	1) Novel ECC-based user access control scheme with ABE. 2) Avoids bilinear pairing in data encryption & decryption. 3) Low performance overhead compared to related schemes.
Chatterjee *et al.* [44] (2018)	FGAC for wireless body area network (FGAC-WBAN)	1) It is the first FGAC scheme in a large-scale WBAN. 2) Supports dynamic node, user addition and revocation. 3) Computation and communication costs are quire low.

3.5 Summary

In this chapter, we have presented an overview of state-of-art of the related works in the areas of multiserver authentication in wireless medium, authentication in crowdsoucing IoT environment, authentication in mobile cloud computing environment and fine-gained access control with user authentication in e-healthcare system. However, it is noted that most schemes proposed in the literature are either vulnerable to different attacks or they require high computational overheads. This literature survey guides us to find out merits and demerits of existing protocols and motivates us to propose more efficient, secure and lightweight protocols in the aforementioned application areas.

Chapter 4

Biometric-Based User Authentication for Multi-Server Environment

The existing schemes in multi-server authentication proposed in the literature involve high computation overhead as they are based on expensive elliptic curve scalar multiplication and modular exponentiation operations. All the existing schemes involve the trusted registration center (RC) in both the login and authentication phases, which can be avoided in the designed scheme in multi-server environment. This motivates us to design a new lightweight, robust and secure scheme in multi server environment in this chapter. A multi-server authentication scheme also faces several challenges. The designed scheme should avoid multiple registrations for the individual servers. The servers or the RC must not store any verification or password table in order to avoid the stolen verifier attack. Furthermore, the designed scheme supports high scalability with dynamic servers and users joining, and revocation.

In this chapter, we aim to propose a new biometric-based authentication mechanism using Chebyshev chaotic map.

4.1 Research contributions

The research contributions in this chapter are discussed as follows.

- We propose a biometric-based authentication scheme using Chebyshev chaotic map, which offers smaller key size, faster computation and higher efficiency for multi-server environment.

- The proposed scheme does not require to maintain any kind of identity-verification table

for identity or password verification which preserves user anonymity property.

- In the proposed scheme, at the time of authentication, a session key is established between the respective server and user without involving the *RC*. This significantly reduces the communication cost and makes the authentication process faster and efficient. However, the existing schemes involve the *RC* for establishing secret session keys at the time of authentication.

- The rigorous formal security analysis using the widely-accepted Real-Or-Random (ROR) model and BAN logic, formal security verification using the popular AVISPA tool as well as informal security analysis show the robustness of the proposed scheme against various well-known attacks.

- The proposed scheme supports the security and functionality features including efficient password change phase, re-registration phase and dynamic server addition phase.

- Compared to the existing recent related schemes, the proposed scheme incurs the low computation and communication overheads. Therefore, the proposed scheme is very suitable for resource constrained and battery powered devices.

4.2 Threat model

We assume that in multi-server authentication, the end-points entities cannot in general be trustworthy, and also communication happens through insecure public channels as per the Dolev-Yao threat model (DY model) [70]. Under the DY model, an adversary not only can eavesdrop the messages, but also can modify, delete or change the contents of the messages being transmitted over the insecure channels.

We further assume that smart card is not tamper resistant. So, an attacker can retrieve any critical information stored in smart-card using sophisticated power analysis attacks [119], [142], [157]. Power analysis is a form of side channel attack in which the attacker studies the power consumption of a cryptographic hardware device (such as a smart card, integrated circuit). The attack can non-invasively extract cryptographic keys and other secret information from the device. Power analysis attack is of two types, namely, 1) simple power analysis (SPA) involves visually interpreting power traces, or graphs of electrical activity over time, and 2) differential power analysis (DPA) is a more advanced form of power analysis which can

allow an attacker to compute the intermediate values within cryptographic computations by statistically analyzing data collected from multiple cryptographic operations [119].

Table 4.1: Notations used in the proposed scheme.

Symbol	Description
RC	Registration center
S_j	j^{th} server
SID_j	Identity of server S_j
x_j	Master secret key of S_j
U_i	i^{th} user
ID_{U_i}	Identity of the i^{th} user, U_i
$T_x(\cdot)$	A Chebyshev polynomial
$H(\cdot)$	A one way cryptographic hash function
$BH(\cdot)$	A secure biohashing function
$E_k(\cdot)/D_k(\cdot)$	Symmetric encryption/decryption using key k
n	Number of users in the system
m	Number of servers initially in the system
m'	Number of servers added later in the system ($m' << m$)
$\|\|, \oplus$	Concatenation and bitwise XOR operations, respectively
$A \rightarrow B : \langle M \rangle$	Entity A sends message M to entity B via open channel
$\triangle TS$	Maximum transmission delay

4.3 The proposed scheme

The notations listed in Table 4.1 are applied to describe the proposed scheme. The various phases related to the proposed scheme are given in subsequent sections.

- **Registration phase:** It is composed of user registration phase as well as server registration phase. In user registration phase, user U_i delivers his/her secret credentials to the RC. The RC further selects master key and other secret parameters for U_i and loads necessary information into his/her smart card. In server registration phase, each server S_j sends its identity to the RC. The RC selects master secret key for S_j,

computes Chebyshev polynomials, and finally loads necessary information in S_j. All communications are executed only once in offline mode.

- **Login phase:** This phase takes identity, password and biometric from U_i and exploits user smart card data to checks if he/she is a registered user or not. If verification is successful, U_i sends login request message to the intended server S_j via a public channel.
- **Authentication and key establishment phase:** In this phase, U_i and S_j mutually authenticate each other and establish a shared session key for future message communication over a public channel.
- **Password and biometric change phase:** In this phase, U_i can update his/her existing password PW_i^{old} to a new password PW_i^{new} freely and completely locally without any involvement of the RC.
- **Dynamic server addition phase:** This phase describes the method for adding a new server into the existing network. An efficient remote multi-server authentication scheme should be highly scalable, that is, it should be able to add new servers into the existing network.
- **User revication and re-registration phase:** In case a user's smart-card is lost or stolen, the proposed scheme keeps the provision for revocation and re-registration of U_i with the same user identity. For this purpose, U_i has to revoke his/her account and re-register without changing identity. This phase describes the revocation and re-registration of a legal user.

4.3.1 Registration phase

Thus, a genuine user needs to submit physically all the registration parameters to the trusted registration center (RC). Moreover, for any practical scenario, a genuine user first physically submits all his/her credential details, and only after verification of all those details, a successful registration happens.

1) Server registration

The RC performs the following steps in offline:

- **Step SR1:** The server S_j chooses its unique identity ID_{S_j} and sends it to the RC through a secure channel.

- **Step SR2:** The RC randomly selects a master key x_j for each server S_j. Next, RC selects K_s and K_u randomly in the interval $(-\infty, +\infty)$ for all registered servers and registered users, respectively. For each server S_j in the system, RC computes Chebyshev polynomials $T_{x_j}(K_s)$ and $T_{x_j}(K_u)$ using master key x_j. Note that the computed keys $T_{x_j}(K_s)$, $T_{x_j}(K_u)$ and x_j are secretly shared between the respective server S_j and the RC.

- **Step SR3:** Finally, the RC loads the following information into each server $\{S_j \mid j = 1, 2, \ldots, m\}$: (i) its own identifier ID_{S_j}, (ii) Chebyshev polynomial $T_{x_j}(K_s)$, (iii) Chebyshev polynomial $T_{x_j}(K_u)$ and (iv) master secret key x_j.

The server registration phase is summarized in Figure 4.1.

2) User registration

The RC executes the following steps in offline for each user U_i:

- **Step UR1:** U_i selects identity ID_{U_i}, password PW_i and imprints personal biometric impression B_i at the sensor of a particular terminal or mobile device. Moreover, U_i selects a random secret number R_i. Next, U_i computes the masked identifier $ID_i = H(ID_{U_i}||R_i||T_i)$, where T_i is the registration timestamp of U_i. Using a secure biohashing function $BH(\cdot)$ and biometrics B_i, U_i computes $b_i = BH(B_i)$. Finally, U_i computes a masked password $RPW_i = H(ID_i||PW_i||b_i||R_i)$, $K_i = H(b_i||R_i||ID_i)$ and $C_i = R_i \oplus H(b_i||ID_{U_i}||PW_i)$. U_i sends $\langle ID_i, T_i, K_i, C_i, RPW_i \rangle$ to the RC through a secure channel. It is worth noting that one can also apply the fuzzy extractor technique for biometric verification purpose as explained in Remark 4.4.

- **Step UR2:** On receiving user parameters, RC selects randomly a master key x_i for each user U_i. Next, for each user U_i, RC selects a variable $K_{u_i} \in (-\infty, +\infty)$, computes Chebyshev polynomials $T_{x_i}(K_{u_i})$ and $T_{x_i}(K_u)$, where $K_u \in (-\infty, +\infty)$ is common for all registered users. Moreover, RC generates parameters $SK_i = K_i \oplus x_i$, $P = K_s \oplus H(x_i||K_i)$ and $A_i = H(ID_{U_i}||\ RPW_i||\ T_i||\ T_{x_i}(K_{u_i})\ ||x_i||P)$.

- **Step UR3:** RC loads the following information into each U_i's smart card ($i = 1, 2, \ldots, n$): (i) user masked identity ID_i, (ii) registration timestamp T_i, (iii) hash value A_i, (iv) Chebyshev polynomial $T_{x_i}(K_{u_i})$, (v) C_i, (vi) SK_i, (vii) P, (viii) Chebyshev polynomial $T_{x_i}(K_u)$ and (ix) $m + m'$ server key-plus-id combinations $\{(ID_{S_j}, T_{x_j}(K_s)) \mid 1 \leq j \leq m + m'\}$.

Server S_j	Registration Center RC
Select server id ID_{S_j}.	Select x_j for server S_j, K_s.
$\xrightarrow[\textbf{(secure channel)}]{\{ID_{S_j}\}}$	For S_j, compute $T_{x_j}(K_s)$
	and $T_{x_j}(K_u)$.
Store received parameters	$\xleftarrow[\textbf{(secure channel)}]{\{ID_{S_j}, T_{x_j}(K_s), T_{x_j}(K_u), x_j\}}$
into server memory.	

User U_i	Registration Center RC
Input ID_{U_i}, PW_i and B_i.	
Select random secret R_i.	
Compute	
$ID_i = H(ID_{U_i}\|\|R_i\|\|T_i)$.	
Compute $b_i = BH(B_i)$,	
$RPW_i = H(ID_i\|\|PW_i\|\|b_i\|\|R_i)$,	Select x_i, K_{u_i} and compute
$K_i = H(b_i\|\|R_i\|\|ID_i)$,	$T_{x_i}(K_{u_i})$, $T_{x_i}(K_u)$,
$C_i = R_i \oplus H(b_i\|\|ID_{U_i}\|\|PW_i)$.	$SK_i = K_i \oplus x_i$,
Send $\langle Uh_i, Ur_i, ID_i\rangle$ to servers.	$P = K_s \oplus H(x_i\|\|K_i)$,
$\xrightarrow[\textbf{(secure channel)}]{\{ID_i, T_i, K_i, C_i, RPW_i\}}$	$A_i = H(ID_{U_i}\|\|RPW_i\|\|T_i\|\|$
	$T_{x_i}(K_{u_i})\|\|x_i\|\|P)$.
	$\{ID_i, T_i, A_i, T_{x_i}(K_{u_i}), SK_i, C_i,$
	$\xleftarrow[\textbf{(secure channel)}]{T_{x_i}(K_u), P, (ID_{S_j}, T_{x_j}(K_s))\}}$

Figure 4.1: Server and user registration phases of the proposed scheme.

- **Step UR4:** After successful registration of each user U_i, the RC computes $Uh_i = H(T_{x_i}(K_{u_i}) \,||Ur_i)$ and sends Uh_i, Ur_i and ID_i to each server S_j for each user U_i. Each server S_j also maintains a database for all registered users, where every record contains $\langle Uh_i, Ur_i, ID_i\rangle$ corresponding to each U_i.

After registration, RC keeps only user masked identifier ID_i and registration timestamp T_i for each user U_i with a unique user registration number Ur_i. This unique Ur_i is also given to U_i for future reference. All other information are deleted from its memory. The user

registration phase is summarized in Figure 4.1.

Remark 4.1. *In the user registration phase, a user's smart card needs to store various parameters as explained in Step UR3. We consider an experimental scenario that comprises of* 20 *initial servers and* 20 *additional servers. We find that the smart card storage overhead is* $(2112 + 42L)$ *bits, where* L *is the size required to store a Chebyshev polynomial (in bits). The standard smart cards, such as integrated circuit processor cards, have maximum data capacity of 8 Kbytes with 8-bit processor [8]. In this case, such smart cards can store* $8 * 8 * 2^{10}$ *bits* $= 65536$ *bits, which is much larger than space required by the proposed scheme. So, the required parameters can be easily stored in the user smart card's memory.*

Remark 4.2. *If an adversary* $\mathcal{A}$ *obtains or steals a genuine user's smart card, he/she may attempt to change smart card parameters, such as* SK_i *and* C_i *which store hash values of* ID_{U_i}, PW_i *and* B_i. *In this situation, the RC uniquely identifies the corresponding user by checking his/her credentials. Finally, the RC can provide a new smart card to the genuine user. Note that this case is also applicable when a genuine user forgets his/her password completely.*

4.3.2 Login phase

U_i performs the following steps in its local system for login purpose:

- **Step L1:** U_i gives his/her identity ID_{U_i}, password PW_i and personal biometric impression B_i, and computes $b_i = BH(B_i)$, where $BH(\cdot)$ is a secure biohashing function. Using the smart card parameter C_i, U_i further computes $R'_i = C_i \oplus H(ID_{U_i} \,||PW_i \,||b_i)$ and masked identifier $ID'_i = H(ID_{U_i}||\ R'_i\ ||T_i)$, where T_i is the registration timestamp collected from his/her own smart card.

- **Step L2:** Next, U_i (smart card) computes masked password $RPW'_i = H(ID'_i||\ PW_i||\ b_i||\ R'_i)$. Applying smart card parameter SK_i, U_i computes $x'_i = H(b_i||\ R'_i||\ ID'_i) \oplus SK_i$. Finally, using the smart card stored parameters $T_{x_i}(K_{u_i})$ and P, U_i calculates $A'_i = H(ID_{U_i}\,||RPW'_i\,||T_i||\ T_{x_i}(K_{u_i})\ ||x'_i||P)$, and then matches it with the stored A_i. If $A'_i = A_i$, we ensure that U_i has entered correct id, password and biometric information to successfully login to the system.

- **Step L3:** U_i selects any particular server S_j with which he/she wants to establish a session. Then U_i computes $K_s = P \oplus\ H(x_i||\ H(b_i||\ R_i||\ ID_i))$, $T_{K_1} = T_{x_i}(T_{x_j}(K_s))$, $T_{x_i}(K_s)$ and $K_1 = H(T_{x_j}(K_s)||ID_i||ID_{S_j}||TS_i)$ where TS_i is U_i' current time stamp.

- **Step L4:** U_i selects random nonce RN_i and prepares the message $M_1 = \{ID_i, ID_{S_j}, E_{K_1}(ID_i||\ ID_{S_j}\ ||T_{K_1}||\ T_{x_i}(K_s)||\ T_{x_i}(K_u)\ ||T_{x_i}(K_{u_i})\ ||RN_i||K_i),\ TS_i,\ H(K_i||\ TS_i||ID_i\ ||ID_{S_j}||\ RN_i||\ T_{x_i}(K_u)||\ T_{K_1})\}$ and sends the login request message $\langle M_1 \rangle$ to S_j.

The login phase is summarized in Figure 4.2.

4.3.3 Authentication and session key establishment phase

The following steps are executed in this phase:

- **Step AK1:** S_j receives message $\langle M_1 \rangle$ and verifies if $\mid TS_i - TS_i^* \mid < \triangle TS_i$. Here, TS_i^* is the current system timestamp of S_j and $\triangle TS_i$ is the maximum transmission delay. If this condition is satisfied, using stored parameters $(T_{x_j}(K_s),\ ID_{S_j})$ and received parameters $(TS_i,\ ID_i)$, S_j computes $K_1' = H(T_{x_j}(K_s)\ ||ID_i\ ||ID_{S_j}\ ||TS_i)$.

- **Step AK2:** Using the computed key K_1', S_j decrypts $E_{K_1}(ID_i\ ||ID_{S_j}\ ||T_{K_1}\ ||T_{x_i}(K_s)\ ||T_{x_i}\ (K_u)\ ||T_{x_i}(K_{u_i})\ ||RN_i\ ||K_i)$ and obtains parameters $ID_i,\ ID_{S_j},\ T_{K_1},\ T_{x_i}(K_s),\ T_{x_i}(K_u),\ T_{x_i}(K_{u_i}),\ RN_i$ and K_i.

- **Step AK3:** Using received parameter $T_{x_i}(K_{u_i})$, S_j finds out the corresponding stored parameter Ur_i for respective ID_i and computes $Uh_i = H(T_{x_i}(K_{u_i})\ ||Ur_i)$. If Uh_i is found in the database record, S_j proceeds further. Otherwise, S_j discards the user login request and sends a deny message to U_i.

- **Step AK4:** Next, S_j computes $H(K_i||\ TS_i||\ ID_i||\ ID_{S_j}||\ RN_i||\ T_{x_i}(K_u)\ ||T_{K_1})$ and compares it with the received hash value. If both values match, S_j computes $T_{K_1}' = T_{x_j}(T_{x_i}(K_s))$ and matches it with received T_{K_1}. If T_{K_1}' is equal to T_{K_1}, S_j authenticates U_i as a genuine user. Otherwise, S_j terminates the process.

- **Step AK5:** After successful authentication of U_i, S_j computes $T_{K_2} = T_{x_j}(T_{x_i}(K_{u_i}))$, $Y = K_i \oplus T_{K_2}$ and $K_2 = H(T_{x_i}(K_{u_i})\ ||ID_{S_j}\ ||ID_i||\ TS_i||\ TS_j||\ RN_i||\ T_{K_1}')$. Here, TS_j is the current time stamp of S_j. S_j then computes $T_{K_3} = T_{x_j}(T_{x_i}(K_u))$.

- **Step AK6:** S_j generates message $M_2 = \{ID_i,\ ID_{S_j},\ E_{K_2}(ID_i\ ||ID_{S_j}\ ||Y\ ||T_{x_j}(K_u)\ ||RN_j\ ||T_{K_3}),\ TS_j,\ H(TS_i\ ||TS_j\ ||RN_i\ ||RN_j\ ||Y\ ||T_{K_3}\ ||T_{x_j}(K_u))$ and sends message $\langle M_2 \rangle$ to U_i via a public channel. Finally, S_j generates the session key $SK_{ij} = H(ID_i||\ ID_{S_j}||\ TS_i||\ TS_j||\ RN_i||\ RN_j||\ T_{K_1}'||T_{K_2}\ ||T_{K_3})$.

User U_i	Server S_j
Insert smart card and input ID_{U_i}, PW_i and B_i.	
Compute (i) $b_i = BH(B_i)$.	
(ii) $R'_i = C_i \oplus H(ID_{U_i}\|\|PW_i\|\|b_i)$	
(iii) $ID'_i = H(ID_{U_i}\|\|R'_i\|\|T_i)$	
(iv) $RPW'_i = H(ID'_i\|\|PW_i\|\|b_i\|\|R'_i)$	
(v) $x'_i = H(b_i\|\|R'_i\|\|ID'_i) \oplus SK_i$	
(vi) $A'_i = H(ID_{U_i}\|\|RPW'_i\|\|T_i\|\|T_{x_i}(K_{u_i})\|\|x'_i\|\|P)$.	Receive message $\{M_1\}$.
Verify $A'_i \stackrel{?}{=} A'_i$	Verify if $\mid TS_i - TS^*_i < \triangle TS_i$?
[accept/reject]	**[accept/reject]**
Compute (i) $T_{K_1} = T_{x_i}(T_{x_j}(K_s))$,	Compute $K'_1 = H(T_{x_j}(K_s)\|\|ID_i\|\|ID_{S_j}\|\|TS_i)$.
(ii) $K_1 = H(T_{x_j}(K_s)\|\|ID_i\|\|ID_{S_j}\|\|TS_i)$,	Using K'_1, decrypt $ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u)$.
(iii) $K_s = P \oplus H(x_i\|\|H(b_i\|\|R_i\|\|ID_i))$,	Compute $H(H(b_i\|\|R_i\|\|ID_i)\|\|TS_i\|\|ID_i\|\|ID_{S_j}\|\|RN_i\|\|$
$M_1 = \{ID_i, ID_{S_j}, E_{K_1}(ID_i\|\|ID_{S_j}\|\|T_{K_1}\|\|T_{x_i}(K_s)$	$T_{x_i}(K_u)\|\|T_{K_1})$.
$\|\|T_{x_i}(K_u)\|\|T_{x_i}(K_{u_i})\|\|RN_i\|\|H(b_i\|\|R_i\|\|ID_i), TS_i, H(H(b_i\|\|$	Search (i) pair $\langle ID_i, Ur_i\rangle$ in server database,
$R_i\|\|ID_i)\|\|TS_i\|\|ID_i\|\|ID_{S_j}\|\|RN_i\|\|T_{x_i}(K_u)\|\|T_{K_1})\}$.	(ii) parameter $Uh_i = H(T_{x_i}(K_{u_i})\|\|Ur_i)$ in database.
$\xrightarrow[\textbf{(public channel)}]{\{M_1\}}$	Verify computed and received hash values.
	[accept/reject]
	Compute $T'_{K_1} = T_{x_j}(T_{x_i}(K_s))$.
	Verify if $T'_{K_1} = T_{K_1}$?
	If verification holds, S_j **authenticates** U_i.
	Compute
	(i) $T_{K_2} = T_{x_j}(T_{x_i}(K_{u_i}))$,
	(ii) $Y = H(b_i\|\|R_i\|\|ID_i) \oplus T_{K_2}$,
	(iii) $K_2 = H(T_{x_i}(K_{u_i})\|\|ID_{S_j}\|\|ID_i\|\|TS_i\|\|TS_j\|\|RN_i\|\|T'_{K_1})$,
	(iv) $T_{K_3} = T_{x_j}(T_{x_i}(K_u))$.
	Generate $M_2 = \{ID_i, ID_{S_j}, E_{K_2}(ID_i\|\|ID_{S_j}\|\|Y\|\|T_{x_j}(K_u)$
Receive message $\{M_2\}$.	$\|\|RN_j\|\|T_{K_3}), TS_j, H(RN_j\|\|TS_j\|\|Y\|\|T_{K_3}\|\|T_{x_j}(K_u))\}$.
Verify if $\mid TS_j - TS^*_j < \triangle TS_j$?	$\xleftarrow[\textbf{(public channel)}]{\{M_2\}}$
[accept/reject]	
Compute $K_2 = H(T_{x_i}(K_{u_i})\|\|ID_{S_j}\|\|ID_i\|\|TS_i\|\|TS_j\|\|RN_i)$.	
Using K_2, decrypt encrypted message.	Generate $SK_{ij} = H(ID_i\|\|ID_{S_j}\|\|TS_i\|\|TS_j\|\|RN_i\|\|RN_j\|\|$
Retrieve $T'_{K_2} = H(b_i\|\|R_i\|\|ID_i). \oplus Y$	$T'_{K_1}\|\|T_{K_2}\|\|T_{K_3})$.
Compute $T'_{K_3} = T_{x_i}(T_{x_j}(K_u))$.	
Verify if $T'_{K_3} = T_{K_3}$?	
If verification holds, U_i **authenticates server** S_j.	
Generate $SK_{ij} = H(ID_i\|\|ID_{S_j}\|\|TS_i\|\|TS_j\|\|RN_i\|\|RN_j\|\|$	
$T_{K_1}\|\|T'_{K_2}\|\|T'_{K_3})$.	

Figure 4.2: Login and authentication & key establishment phases of the proposed scheme.

- **Step AK7:** After receiving message M_2 from S_j, U_i first checks whether $\mid TS_j - TS_j^* \mid < \triangle TS_j$ holds or not. Here, TS_j^* is the current system timestamp of U_j. If this condition holds, U_i computes $K_2 = H(T_{x_i}(K_{u_i}) \mid\mid ID_{S_j} \mid\mid ID_i \mid\mid TS_i \mid\mid TS_j \mid\mid RN_i)$ using its own parameters and received server timestamp TS_j. Followed by this, U_i decrypts the encrypted message $E_{K_2}(ID_i \mid\mid ID_{S_j} \mid\mid Y \mid\mid T_{x_j}(K_u) \mid\mid RN_j \mid\mid T_{K_3})$ using the computed K_2, and retrieves the encrypted parameters. Moreover, U_i computes $T'_{K_2} = K_i \oplus Y$ and $T'_{K_3} = T_{x_i}(T_{x_j}\ (K_u))$. If the condition $T'_{K_3} = T_{K_3}$ holds, U_i authenticates S_j successfully. Finally, U_i generates the same session key $SK_{ij} = H(ID_i \mid\mid ID_{S_j} \mid\mid TS_i \mid\mid TS_j \mid\mid RN_i \mid\mid RN_j \mid\mid T_{K_1} \mid\mid T'_{K_2} \mid\mid T'_{K_3})$ for future message communication with S_j.

This authentication and session key establishment phase is summarized in Figure 4.2.

Remark 4.3. *In the proposed scheme, the registration center RC's involvement is only needed to load the necessary parameters in the servers and users' devices during the registration phase. Thus, the need to involve the RC during the login and authentication phases is not required in the proposed scheme. On the other hand, existing related schemes proposed in multi-server environment involve the RC in the login and authentication phases, where the communication between a server S_j and the RC, and between the RC and S_j take place. The proposed scheme requires only two messages communication between a user U_i and S_j, and between S_j and U_i, each of them needs only* 640 *bits (see Table 4.4). Furthermore, compared to other related existing schemes, the proposed scheme involves minimum communication and computation overheads (see Tables 4.4 and 4.6).*

4.3.4 Password and biometric change phase

In this phase, a user U_i can update his/her existing password PW_i^{old} to a new password PW_i^{new} freely and completely locally without any involvement of the RC. For this purpose, he/she performs the following steps:

- **Step PB1:** U_i inputs his/her smart card into the card reader of a specific terminal and inputs identity ID_{U_i}, password PW_i^{old} and personal biometrics B_i^{old}. The smart card computes $b_i = BH(B_i^{old})$, $R'_i = C_i \oplus H(ID_{U_i} \mid\mid PW_i \mid\mid b_i)$, masked identifier $ID'_i = H(ID_{U_i} \mid\mid R'_i \mid\mid T_i)$, $RPW'_i = H(ID'_i \mid\mid PW_i \mid\mid b_i \mid\mid R'_i)$, $x'_i = H(b_i \mid\mid R'_i \mid\mid ID'_i) \oplus SK_i$ and $A'_i = H(ID_{U_i} \mid\mid RPW'_i \mid\mid T_i \mid\mid T_{x_i}(K_{u_i}) \mid\mid x'_i \mid\mid P)$. If $A'_i = A_i$, login is successful.

- **Step PB2:** After successful login, U_i provides his/her new changed password PW_i^{new} and imprints new biometrics B_i^{new}. After that the smart card computes $b_i^{new} =$

$BH(B_i^{new})$, $C_i^{new} = R_i' \oplus H(b_i^{new}||ID_{U_i}||PW_i^{new})$, $RPW_i' = H(ID_i' ||PW_i^{new} ||b_i^{new} ||R_i')$, and $A_i^{new} = H(ID_{U_i} ||RPW_i' ||T_i ||T_{x_i}(K_{u_i}) ||x_i' ||P)$.

- **Step PB3:** Finally, A_i and C_i are replaced with A_i^{new} and C_i^{new} into smartcard's memory, respectively.

The password and biometric change phase is summarized in Figure 4.3.

User (U_i)	Smart card of U_i
Insert own smart card into card reader.	
Input ID_{U_i} and old password PW_i^{old}.	
Imprint personal biometrics B_i^{old}.	
	Calculate $b_i = BH(B_i^{old})$.
	$R_i' = C_i \oplus H(ID_{U_i} ||PW_i ||b_i)$.
	$ID_i' = H(ID_{U_i}|| R_i' ||T_i)$.
	$RPW_i' = H(ID_i'|| PW_i|| b_i|| R_i')$.
	$x_i' = H(b_i|| R_i'|| ID_i')\oplus SK_i$.
	$A_i' = H(ID_{U_i} ||RPW_i' ||T_i|| T_{x_i}(K_{u_i}) ||x_i'||P)$.
	Verify the condition $A_i' = A_i$,
	If verification holds
	Allow U_i to enter new password PW_i^{new}.
Input new changed password PW_i^{new}.	
Input new biometrics B_i^{new}.	
	Calculate new $b_i^{new} = BH(B_i^{new})$,
	$C_i^{new} = R_i' \oplus H(b_i^{new}||ID_{U_i}||PW_i^{new})$.
	$RPW_i' = H(ID_i' ||PW_i^{new} ||b_i^{new} ||R_i')$.
	$A_i^{new} = H(ID_{U_i} ||RPW_i' ||T_i ||T_{x_i}(K_{u_i}) ||x_i' ||P)$.
	Replace A_i with A_i^{new} in its memory.
	Replace C_i with C_i^{new} in its memory.

Figure 4.3: Password and biometric change phase of the proposed scheme.

Remark 4.4. *A fuzzy extractor contains the following two procedures:*

- ***Gen:*** *It is probabilistic fuzzy generator function which takes user's personal biometrics, say B_i as input and outputs a pair of biometric secret key b_i and a public reproduction parameter τ_i as $Gen(B_i) = (b_i, \tau_i)$.*

- ***Rep:*** *It is a deterministic fuzzy reproduction function which takes user's biometrics* B_i' *and* τ_i *as inputs and recovers the original biometric secret key* $b_i = Rep(B_i', \tau_i)$ *provided that the Hamming distance between the current biometrics* B_i' *and previously registered biometrics* B_i *is less than an error tolerance threshold value,* t.

If we use the fuzzy extractor in place of the biohashing function $BH(\cdot)$*, we need the following modifications in the proposed scheme:*

(i) *During the user registration phase (Step UR1),* U_i *needs to compute* $Gen(B_i) = (b_i, \tau_i)$. *In Step UR3, the RC also needs to load* $Gen(\cdot)$ *and* $Rep(\cdot)$, τ_i *and* t *in each user* U_i*'s smartcard.*

(ii) *During the login phase (Step L1),* U_i *needs to calculate* $b_i = Rep(B_i, \tau_i)$.

(iii) *During the password and biometric update phase (Step PB1), the smartcard needs to compute* $b_i = Rep(B_i^{old}, \tau_i)$. *In Step PB2, the smartcard also needs to compute* b_i^{new} *and* τ_i^{new} *as* $Gen(B_i^{new}) = (b_i^{new}, \tau_i^{new})$. *In addition,* $Gen(\cdot)$, $Rep(\cdot)$ *and* t *are required to store in the memory of the smart card. Furthermore,* τ_i *needs to be replaced by* τ_i^{new} *in the smartcard's memory.*

4.3.5 Dynamic server addition phase

An efficient remote multi-server authentication scheme should be highly scalable, that is, it should be able to add new servers into the existing network. In this phase, we describe the method for adding a new server into the existing network.

- **Step D1:** Whenever a new server S_j is deployed into the network, prior to deployment the RC assigns a unique identifier ID_{S_j} and master secret key x_j to it. Next, the RC computes the Chebyshev polynomial $T_{x_j}(K_s)$ and loads master secret key x_j and $T_{x_j}(K_s)$ in the memory of S_j.

- **Step D2:** After deployment of S_j, the RC informs the users U_i about addition of S_j. Thus, it is noted that no other information is required to store in the user's smart card regarding addition of servers.

The dynamic server addition phase is summarized in Figure 4.4.

Remark 4.5. *In many situations, revocation of an existing server may be essential. For revocation of a server S_j, the RC needs to simply inform the revoked server's identity (ID_{S_j}) to all users in the system, and also requests them to delete key-plus-id combination ($ID_{S_j}, T_{x_j}(K_s)$) from their smart cards.*

User (U_i)	Registration center(RC)	New server (S_j)
PRIOR TO SERVER DEPLOYMENT		
	Assign unique identifier ID_{S_j}, master secret key x_j. Compute Chebyshev polynomial $T_{x_j}(K_s)$. $\{x_j, T_{x_j}(K_s)\}$ $\xrightarrow[\textbf{(secure Channel)}]{}$	
AFTER SERVER DEPLOYMENT		
		Load $\{ID_{S_j}, x_j, T_{x_j}(K_s)\}$ into its own memory.
	$\xleftarrow[\textbf{(secure channel)}]{\{ID_{S_j}, T_{x_j}(K_s)\}}$	
Store $\{ID_{S_j}, T_{x_j}(K_s)\}$ into smart card.		

Figure 4.4: Dynamic server addition phase of the proposed scheme.

4.3.6 User revocation and re-registration phase

As the user smart card may be lost or stolen any where any time, user revocation and re-registration is essential for any smart card based authentication scheme. In case a user's smart-card is lost or stolen, the proposed scheme keeps the provision for user revocation and re-registration of that user U_i with the same user identity. For that, user has to revoke his/her account and re-register without changing identity ID_{U_i}. For revocation, user needs to provide his/her all authorized documents like PAN card, date of birth, passport etc. Here, we describe the revocation and re-registration phase as follows.

- **Step Re1:** For revocation, the RC securely sends a user revocation message containing

Uh_i, Ur_i and ID_i to each server for each revoked user U_i. After receiving this message, server simply puts a revocation flag for that user in its registered users database.

- **Step Re2:** At the time of authentication during Step **AK3**, if server finds the computed Uh_i has been notified by revocation flag in registered users' database, server discards the user authentication request message with a deny message to the corresponding user.

- **Step Re3:** If any genuine user wants to re-register with the same identity, the RC first verifies ID_i by computing the user identity ID_{U_i} and the old registration timestamp T_i for U_i. If it is valid, the RC executes the registration phase to reactivate the account of U_i.

The user revocation process as well as user re-registration process are summarized in Figure 4.5.

4.4 Security analysis

In Section 4.4.1, through the formal security using the Real-Or-Random (ROR) model, we prove the semantic security of the proposed protocol. The mutual authentication proof between a user and a server in the proposed protocol with the help of widely-used BAN logic [34] is done in Section 4.4.2. In Section 4.4.3, we provide the informal security analysis to show that the proposed protocol prevents other known attacks. Furthermore, in Section 4.5 we simulate the proposed scheme for the formal security verification using broadly-accepted AVISPA tool [22] to show that the proposed scheme is secure against replay and man-in-the-middle attacks.

Wang *et al.* [204] reviewed several anonymous two-factor authentication schemes and then pointed out that under the current widely accepted adversarial model, certain goals are beyond attainment. They further observed that the widely used formal methods including random oracle model and BAN logic can not capture some structural mistakes, and hence, guaranteeing the soundness of authentication protocols still remains an open issue. Due to such important observations in their analysis, it is necessary to have all the formal security analysis, BAN logic analysis, informal security analysis and formal security verification of the proposed scheme so that the designed schemes can achieve high level security.

Registration center(RC)	Server (S_j)	Revoked user (U_i)
USER REVOCATION PROCESS		
User revocation message $\xrightarrow[\textbf{(secure channel)}]{\{Uh_i, Ur_i, ID_i\}}$		
	Put revocation flag in server database against U_i.	
		User login request $\xleftarrow[\textbf{(public channel)}]{\{Uh_i, Ur_i, ID_i\}}$
	Verify if registration flag is on for U_i.	
	If verification holds, discard login request.	

User (U_i)	Registration center(RC)
USER RE-REGISTRATION PROCESS	
Re-registration request message $\xrightarrow[\textbf{(secure channel)}]{\{ID_{U_i}\}}$	
	Fetch stored registration timestamp T_i.
	Fetch stored random number R_i.
	Compute $ID_i^* = H(ID_{U_i}\|\|R_i\|\|T_i)$.
	Verify if stored $ID_i = ID_i^*$?
	If verification holds, initiate user registration as described in Section 4.3.1.

Figure 4.5: User revocation and re-registration phase of the proposed scheme.

4.4.1 Formal security analysis using ROR model

This section presents the formal security analysis of the proposed multi-server authentication protocol.

In order to break the security of the proposed multi-server authentication protocol $\mathscr{P}$, $\mathcal{A}$ executes different kinds of attacks. We simulate the attacks using various oracle queries as explained below.

- ***Execute***(U_i, S_j): Through this passive attack, $\mathcal{A}$ can eavesdrop or output a message m

communicated between U_i and S_j in an actual execution of the protocol $\mathscr{P}$.

- ***Send***$(U_i/S_j,m)$: This active attack enables $\mathcal{A}$ to receive an actual reply message from participant $\mathcal{P}^t$. $\mathcal{A}$ sends a request message m to $\mathcal{P}^t$, and $\mathcal{P}^t$ replies to $\mathcal{A}$ according to the rules of the protocol.
- ***Reveal***$(\mathcal{P}^t)$: *Reveal* simulation query reveals the current session key SK generated by $\mathcal{P}^t$ (and its partner) to $\mathcal{A}$.
- ***Corrupt***(U_i,a): *Corrupt* simulation query models the capability of $\mathcal{A}$ to obtain secret information of a user participant U_i, thereby corrupting the protocol.
  - If $a = 1$, query returns password of U_i to $\mathcal{A}$.
  - If $a = 2$, query returns secret biometric string of U_i to $\mathcal{A}$.
  - If $a = 3$, query returns user U_i's smart card stored parameters to $\mathcal{A}$.
- ***Test***$(\mathcal{P}^t)$: *Test* query can be invoked only once and is used to measure the strength of the semantic security of session key SK. $\mathcal{A}$ sends this query to participant $\mathcal{P}^t$. If no session key is generated for the current session, a *null* value is resulted. Otherwise, $\mathcal{P}^t$ takes decision according to the outcome of an unbiased flipped coin b. If $b = 1$, $\mathcal{P}^t$ returns current computed session key to $\mathcal{A}$. If $b = 0$, $\mathcal{P}^t$ returns a *random string* of same length to $\mathcal{A}$.

We now define the following definitions [26], [223] prior to proving Theorem 4.1.

Definition 4.1. *An instance $\mathcal{P}^t$ is known to be accepted, if upon receiving the last expected protocol message, it goes into an accept state. The ordered concatenation of all communicated (sent and received) messages by instance $\mathcal{P}^t$ forms the session identification (sid) of $\mathcal{P}^t$ for the current session.*

Definition 4.2. *Two instances ${U_i}^{t_1}$ and ${S_j}^{t_2}$ is said to be partnered if following three conditions are fulfilled simultaneously: 1) both ${U_i}^{t_1}$ and ${S_j}^{t_2}$ are in accept state; 2) both ${U_i}^{t_1}$ and ${S_j}^{t_2}$ mutually authenticate each other and share the same sid; and 3) ${U_i}^{t_1}$ and ${S_j}^{t_2}$ are mutual partners of each other.*

Definition 4.3 (Freshness)**.** *An instance $\mathcal{P}^t$ is known to be fresh, when the following conditions are met simultaneously: 1) $\mathcal{P}^t$ is in accept state; 2) Reveal($\mathcal{P}^t$) query has never been submitted to $\mathcal{P}^t$ or its partner; and 3) strictly less than two Corrupt($\mathcal{P}^t$,a) queries has been*

submitted to $\mathcal{P}^t$, if $\mathcal{P} \in U_i$. Otherwise, if $\mathcal{P} \in S_j$, then strictly less than two Corrupt($\mathcal{P}^t$,a) query has been submitted to $\mathcal{P}$'s partner.

Definition 4.4 (Semantic security)**.** *Let $Succ(\mathcal{A})$ refers to an event where an adversary $\mathcal{A}$ execute a single $Test$ ($\mathcal{P}^t$) query with chosen bit b directed a to a fresh instance $\mathcal{P}^t$ and the query outputs a guess bit b'. If $b' = b$, the adversary $\mathcal{A}$ is successful in breaking the semantic security of the multi-server authentication protocol ($MSAP$). The advantage function of adversary $\mathcal{A}$ in breaking the semantic security of our protocol $\mathscr{P}$ by guessing the correct bit b' is defined by*

$$Adv_{\mathscr{P}}^{MSAP}(\mathcal{A}) = |2Pr[Succ(\mathcal{A})] - 1| = |2.Pr[b = b'] - 1|.$$

Definition 4.5. *A biometrics-based password authentication protocol is semantically secure if the advantage function $Adv_{\mathscr{P}}^{MSAP}(\mathcal{A})$ 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, $|\mathcal{D}|$ is the size of password dictionary, l_b denotes the extracted string length of user biometrics and ε_{bm} is the probability of "false positive" [158].*

Definition 4.6. *The advantage probability of $CMDLP$ problem is negligible for any adversary $\mathcal{A}$, i.e., $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}) \leq \epsilon$, for any sufficiently small $\epsilon > 0$.*

- During the server registration phase, the server S_j stores the parameters $\{ID_{S_j}, T_{x_j}(K_s), T_{x_j}(K_u), x_j\}$. In the user registration phase, the registration server (RC) loads $\{ID_i, T_i, A_i, T_{x_i}(K_{u_i}), SK_i, C_i, T_{x_i}(K_u), P, ID_{S_j}, T_{x_j}(K_s)\}$ into the smart card of the user U_i. U_i selects a password PW_i from an evenly distributed finite dictionary $\mathcal{D}$ with size $|\mathcal{D}|$. Further, U_i owns identity ID_{U_i}, biometrics B_i and one-way cryptographic hash function $H(\cdot)$.

- We consider two participants of the proposed protocol: user U_i and server S_j. To remove ambiguity, we mention a common notation $\mathcal{P}$ for both participants U_i and S_j. An execution of the protocol $\mathscr{P}$ is termed as an instance. To attack the proposed protocol $\mathscr{P}$, let $\mathcal{A}$ be a probabilistic polynomial time adversary that execute the oracle queries and interact with the t^{th} instance of an executing participant $\mathcal{P}^t$. An oracle query has three possible outcomes: (a) oracle receives correct message (*accept*), (b) oracle receives incorrect message (*reject*), and (c) oracle does not receive any result, i.e., no conclusion is achieved ($\perp$).

- The malicious adversary $\mathcal{A}$ has absolute control over communication channel [70]. So, $\mathcal{A}$ has the potential of blocking, intercepting, modifying or removing a message m communicated between U_i and S_j.

- Once $\mathcal{A}$ achieves smart card of user U_i, it can extract the secret stored information using the power analysis attacks [119], [142].

Theorem 4.1. *Let $\mathcal{A}$ be a polynomial time bounded attacker (or adversary) running within time upper bound $t_{\mathcal{A}}$. Suppose in order to break the semantic security of the proposed multi-server authentication protocol $\mathscr{P}$, $\mathcal{A}$ makes at most q_s times Send queries, q_e times Execute queries, q_H and q_{BH} times H and BH hash oracle queries, respectively. Then,*

$$\begin{aligned} Adv_{\mathscr{P}}^{MSAP}(\mathcal{A}) \leq & \frac{q_h^2 + 10q_h}{2^{l_h}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} \\ & + \frac{(q_s+q_e)^2 + 4q_s}{2^{l_r}} + 4q_h(1+(q_s+q_e)^2)Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}), \end{aligned}$$

where l_h refers to the string length of hash results, l_r is the string length of random numbers, l_b denotes extracted string length of user biometrics, ε_{bm} is the probability of "false positive" [158], $\mathcal{D}$ is a finite dictionary with size $|\mathcal{D}|$, and $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$ is the advantage probability of $\mathcal{A}$ in breaking $CMDLP$ (see Definition 2.7).

Proof. We define a set of games G_i $(i = 0, 1, 2, 3, 4, 5)$ starting from G_0 and terminating at 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 an adversary $\mathcal{A}$.

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

$$Adv_{\mathscr{P}}^{MSAP}(\mathcal{A}) = |2Pr[Succ_0] - 1|. \tag{4.1}$$

- **Game G_1:** This game simulates all oracle queries including *Send*, *Reveal*, *Execute*, *Corrupt*, *Test* and *hash* queries except *Send* query. Working procedures of these queries as per the proposed protocol $\mathscr{P}$ are described in Table 4.2. *Send* query is simulated in Table 4.3. Moreover, G_1 creates three lists: (1) list L_h answers hash oracles H, (2) list L_a stores outputs of random oracle queries, and (3) list L_T records transcripts between U_i and S_j. Due to indistinguishability of simulation of G_1 and real protocol execution of G_0, we obtain the following:

$$Pr[Succ_1] = Pr[Succ_0]. \tag{4.2}$$

Table 4.2: 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 hash function 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_i,a) simulation query performs as follows: If $a = 1$, the query returns password (PW_i) of the user U_i. If $a = 2$, the query outputs biometrics (B_i) of U_i. If $a = 3$, the query returns the secret information stored in user smart card.
Simulation of ***Execute(U_i,S_j)*** query occurs in succession with simulation of *Send* queries as shown below. Let $D_1 = E_{K_1}(ID_i \| ID_{S_j} \| T_{K_1} \| T_{x_i}(K_s) \| T_{x_i}(K_u) \| T_{x_i}(K_{u_i}) \| RN_i \| H(b_i \| R_i \| ID_i))$ and $H_1 = H(H(b_i \| R_i \| ID_i) \| TS_i \| ID_i \| ID_{S_j} \| RN_i \| T_{x_i}(K_u) \| T_{K_1})$. U_i sends message M_1 to S_j, where $M_1 = \{ID_i, ID_{S_j}, D_1, TS_i, H_1\}$. Let $D_2 = E_{K_2}(ID_i \| ID_{S_j} \| Y \| T_{x_j}(K_u) \| RN_j \| T_{K_3}) \| TS_j$ and $H_2 = H(RN_j \| TS_j \| Y \| T_{K_3} \| T_{x_j}(K_u))$. S_j sends message M_2 to U_i, where $M_2 = \{ID_i, ID_{S_j}, D_2, H_2\}$. Note that $\langle ID_i, ID_{S_j}, D_1, TS_i, H_1\rangle \leftarrow Send(U_i, \textbf{start})$ and $\langle ID_i, ID_{S_j}, D_2, H_2\rangle \leftarrow Send(S_j, \langle ID_i,$ $\mathrm{ID}_{S_j}, D_1, TS_i, H_1\rangle)$. Finally, $M_1 = \langle ID_i, ID_{S_j}, D_1, TS_i, H_1\rangle$ and $M_2 = \langle ID_i, ID_{S_j}, D_2, H_2\rangle$ are returned.

- **Game G_2:** In this game, we consider a collision situation with hash results and random numbers in the transcripts of messages M_1 and M_2 of our protocol $\mathscr{P}$. Following the birthday paradox, the collision probability of oracle H query is at most $\frac{q_h^2}{2^{l_h+1}}$. Further, messages M_1 and M_2 contain random numbers RN_i and RN_j, and the probability of random numbers collision is at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. So, we have,

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}}. \tag{4.3}$$

- **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 authentication phase of our protocol $\mathscr{P}$ involves two messages communication, we consider following two cases in game Game G_3.

Table 4.3: Simulation of send oracle queries.

Send simulation query performs as follows.
(a) Let U_i be the destination target state. For a $Send(U_i, \textbf{start})$ query, U_i gives the following response.
Compute $T_{K_1} = T_{x_i}(T_{x_j}(K_s))$, $K_1 = H(T_{x_j}(K_s)||ID_i||ID_{S_j}||TS_i)$, $K_s = P \oplus H(x_i||H(b_i||R_i||$ $\text{ID}_i))$, D_1 and H_1 as in Table 4.2. Output $M_1 = \langle ID_i, ID_{S_j}, D_1, TS_i, H_1 \rangle$.
(b) Let S_j be the target state. For a $Send(S_j, \langle ID_i, ID_{S_j}, D_1, TS_i, H_1 \rangle)$ query, U_i gives the following response.
Verify whether $\mid TS_i - TS_i^* \mid < \triangle TS_i$. If it is so, compute K_1', decrypt the received encrypted D_1, and verify the computed and received hash values.
A mismatch rejects the session. Otherwise, compute $T'_{K_1} = T_{x_j}(T_{x_i}(K_s))$ and match with the received T_{K_1}.
If it is successful, compute T_{K_2}, Y, K_2, T_{K_3} as in Table 4.2.
Further, it creates the parameters D_2 and H_2 as shown in Table 4.2 and outputs $M_2 = \langle ID_i,$ $\text{ID}_{S_j}, D_2, H_2 \rangle$.
(c) We assume that U_i is destination target state. For a $Send(U_i, \langle ID_i, ID_{S_j}, D_2, H_2 \rangle)$ query, U_i answers as follows:
If $\mid TS_j - TS_j^* \mid$ is more than maximum transmission delay, abort the session.
Otherwise, compute K_2, retrieve T'_{K_2}, compute T'_{K_3} and verify authenticity of T'_{K_3}.
If it is incorrect, terminate the session. Otherwise, authenticate S_j and establish SK_{ij} as the session key.
Finally, both U_i and S_j accept the successful termination of the session.

- **Case 1:** Considering $Send(S_j, M_1)$ query, we must carefully watch message M_1. The hash value $H(H(b_i||\ R_i||\ ID_i)\ ||TS_i\ ||ID_i\ ||ID_{S_j}\ ||RN_i\ ||T_{x_i}(K_u)\ ||T_{K_1})$ $\in L_{\mathcal{A}}$ must hold, otherwise the session will be terminated. The maximum calculated probability is up to $\frac{q_h}{2^{l_h}}$. Again, it must be that $H(b_i||R_i||ID_i) \in L_{\mathcal{A}}$ and $(T_{x_j}(K_s)||ID_i||ID_{S_j}||TS_i, K_1) \in L_{\mathcal{A}}$, whose probabilities are at most $\frac{q_h}{2^{l_h}}$ and $\frac{q_h}{2^{l_h}}$, respectively. Finally, the message $M_1 \in L_T$ should hold, or the session will stop. For this, the probability is $\frac{q_s}{2^{l_r}}$.

- **Case 2:** To respond $Send(U_i, M_2)$ oracle query, $H(RN_j||\ TS_j||\ Y||\ T_{K_3}||\ T_{x_j}(K_u))$ $\in L_{\mathcal{A}}$ and $(T_{x_i}(K_{u_i})\ ||ID_{S_j}\ ||ID_i\ ||TS_i\ ||TS_j\ ||RN_i\ ||T'_{K_1}, K_2) \in L_{\mathcal{A}}$ must hold with the total maximum probability $\frac{2q_h}{2^{l_r}}$. Finally, for a transcript $M_2 \in L_T$, we get the maximum probability as $\frac{q_s}{2^{l_r}}$.

Considering both cases, we have,

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{5q_h}{2^{l_h}}. \quad (4.4)$$

- **Game G_4:** This game considers all online and offline attacks executed by the adversary $\mathcal{A}$. As our protocol $\mathscr{P}$ 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_i and biometrics B_i, $\mathcal{A}$ requires all information stored in smart card of U_i. For this purpose, $\mathcal{A}$ executes $Corrupt(U_i, 3)$ which is composed of the following two sub-cases.

    * **Case 1.1:** For online password guessing, $\mathcal{A}$ runs query $Corrupt(U_i, 1)$. Here, $\mathcal{A}$ selects a password on-the-fly from dictionary $\mathcal{D}$ and then runs at most q_s times *Send(S_j,M_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 $Corrupt(U_i, 2)$. 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, $\varepsilon_{bm} \approx 2^{-14}$ [158]. As a whole, the guessing probability under this case is at most $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$.

  - **Case 2:** For launching offline guessing attack, once $\mathcal{A}$ executes $Corrupt(U_i, 3)$ along with $Corrupt(U_i, 1)$ or $Corrupt(U_i, 2)$ query, he/she executes either pure $Execute(U_i, S_j)$ or successive $Send$ queries with hash oracles. As in our protocol $\mathscr{P}$, the encryption keys K_1 and K_2 are created with hash values of chaotic map parameters, $\mathcal{A}$ needs to solve $CMDLP$ with hash oracle simultaneously. So, for this case, the probability is at most $2q_h Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$.

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

$$|Pr[Succ_4] - Pr[Succ_3]| \leq \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} + 2q_h Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}). \quad (4.5)$$

- **Game G_5:** This final game G_5 considers strong forward security. Here, $\mathcal{A}$ simulates $Execute$, $Send$ and $Hash$ oracle queries on the old transcripts only. For this purpose, two indexes α, $\beta \in \{1, 2, \ldots, q_s + q_e\}$ are chosen. The game is terminated if $Test$

query can not return a valid session key for α^{th} instance of U_i and β^{th} instance of S_j, respectively. Following the analysis of G_4, we find that

$$|Pr[Succ_5] - Pr[Succ_4]| \leq 2q_h(q_s + q_e)^2 \times Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}). \tag{4.6}$$

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

$$Pr[Succ_5] = \frac{1}{2}. \tag{4.7}$$

Using the triangular inequality, we have the following:

$$\begin{aligned} |Pr[Succ_0] - \frac{1}{2}| &= |Pr[Succ_1] - Pr[Succ_5]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_5]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_3]| \\ &\quad + |Pr[Succ_3] - Pr[Succ_5]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_3]| \\ &\quad + |Pr[Succ_3] - Pr[Succ_4]| + |Pr[Succ_4] - Pr[Succ_5]|. \end{aligned} \tag{4.8}$$

Using Equations (4.1)-(4.8), we obtain,

$$\begin{aligned} \frac{1}{2}Adv_{\mathscr{P}}^{MSAP}(\mathcal{A}) &= |Pr[Succ_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}} + \frac{2q_s}{2^{l_r}} + \frac{5q_h}{2^{l_h}} \\ &\quad + \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} + 2q_h Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}) \\ &\quad + 2q_h(q_s + q_e)^2 Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}). \end{aligned} \tag{4.9}$$

Finally, multiplying both sides by 2 in Equation (4.9) and rearranging the terms, we obtain the required result. Hence, the theorem is proved. □

4.4.2 Mutual authentication proof based on BAN logic

Basic BAN logic notations and logical postulates are provided in Section 2.6. According to the analytic procedures of the BAN logic, the proposed protocol needs to satisfy the following goals:

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

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

The generic types of the proposed protocol are given below:

- **Message 1.** $U_i \rightarrow S_j : (ID_i,\, ID_{S_j},\, E_{K_1}(ID_i \,||ID_{S_j} \,||T_{K_1} \,||T_{x_i}(K_s) \,||T_{x_i}(K_u) \,||T_{x_i}(K_{u_i}) \,||RN_i||K_i||TS_i),\, TS_i,\, H(K_i \,||TS_i \,||ID_i \,||ID_{S_j}||\, RN_i||\, T_{x_i}(K_u) \,||T_{K_1}))$.

- **Message 2.** $S_j \rightarrow U_i : (ID_i,\, ID_{S_j},\, E_{K_2}\, (ID_i \,||ID_{S_j} \,||Y \,||T_{x_j}(K_u) \,||RN_j \,||T_{K_3}),\, TS_j,\, H(TS_i \,||TS_j \,||RN_i \,||RN_j \,||Y \,||T_{K_3} \,||T_{x_j}(K_u)))$.

The idealized form of the proposed protocol are given below:

- **Message 1.** $U_i \rightarrow S_j : (ID_i,\, ID_{S_j},\, \{ID_i,\, ID_{S_j},\, T_{K_1},\, T_{x_i}(K_s),\, T_{x_i}(K_u),\, T_{x_i}(K_{u_i}),\, RN_i,\, K_i,\, TS_i\}_{K_1},\, TS_i,\, \langle K_i,\, TS_i,\, ID_i,\, ID_{S_j},\, RN_i,\, T_{x_i}(K_u)\rangle_{T_{K_1}})$.

- **Message 2.** $U_i \rightarrow S_j : (ID_i,\, ID_{S_j},\, \{ID_i,\, ID_{S_j},\, \langle T_{K_2}\rangle_{K_i},\, T_{x_j}(K_u),\, RN_j,\, T_{K_3},\, TS_j\}_{K_2},\, TS_j, \langle\, TS_i,\, TS_j,\, RN_i,\, RN_j,\, Y,\, T_{x_j}(K_u)\rangle_{T_{K_3}})$.

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

- **A.1:** $U_i \mid\equiv \#(TS_j)$
- **A.2:** $S_j \mid\equiv \#(TS_i)$
- **A.3:** $U_i \mid\equiv S_j \Rightarrow (ID_{S_j}, TS_j, RN_j, T_{x_j}(K_u))$
- **A.4:** $S_j \mid\equiv U_i \Rightarrow (ID_i, ID_{S_j}, K_i, TS_i, RN_i, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), T_{x_j}(K_s))$
- **A.5:** $U_i \mid\equiv (x_i, TS_i, RN_i, T_{x_j}(K_s))$
- **A.6:** $S_j \mid\equiv (x_j, TS_j, RN_j, T_{x_j}(K_s), T_{x_j}(K_u))$
- **A.7:** $U_i \mid\equiv (U_i \overset{K_1}{\rightleftharpoons} S_j)$
- **A.8:** $S_j \mid\equiv (U_i \overset{K_1}{\rightleftharpoons} S_j)$
- **A.9:** $U_i \mid\equiv (U_i \overset{K_2}{\rightleftharpoons} S_j)$
- **A.10:** $S_j \mid\equiv (U_i \overset{K_2}{\rightleftharpoons} S_j)$

Based on the above-mentioned assumptions and the logical postulates of BAN logic, we analyze the idealized form of the proposed scheme and provide the main procedures of proof as follows. According to the message 1, we obtain:

- S_1: $S_j \lhd (ID_i, ID_{S_j}, \{ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i\}_{K_1}, TS_i, \langle K_i, TS_i, ID_i, ID_{S_j}, RN_i, T_{x_i}(K_u)\rangle_{T_{K_1}})$.
- S_2: According to the above mentioned inference rule (Rule 5), we obtain $S_j \lhd \{ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i\}_{K_1}$.
- S_3: According to the assumption A.8 and Rule 1, we obtain $S_j \mid\equiv U_i \mid\sim (ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i)$.
- S_4: According to the assumption A.2 and Rule 3, we obtain $S_j \mid\equiv \#(ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i)$.
- S_5: According to Rule 2, we obtain $S_j \mid\equiv U_i \mid\equiv (ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i)$.
- S_6: According to the assumption A.4 and Rule 4, we obtain $S_j \mid\equiv (ID_i, ID_{S_j}, T_{K_1}, T_{x_i}(K_s), T_{x_i}(K_u), T_{x_i}(K_{u_i}), RN_i, K_i, TS_i)$.
- S_7: According to S_6 and Rule 5, we obtain $S_j \mid\equiv ID_i$, $S_j \mid\equiv ID_{S_j}$, $S_j \mid\equiv T_{K_1}$, $S_j \mid\equiv T_{x_i}(K_s)$, $S_j \mid\equiv T_{x_i}(K_u)$, $S_j \mid\equiv T_{x_i}(K_{u_i})$, $S_j \mid\equiv RN_i$, $S_j \mid\equiv TS_i$.
- S_8: According to the assumption $A.6$, we get $S_j \mid\equiv x_j$, $S_j \mid\equiv TS_j$, $S_j \mid\equiv RN_j$, $S_j \mid\equiv T_{x_j}(K_s)$, $S_j \mid\equiv T_{x_j}(K_u)$.
- S_9: According to the proposed scheme, $SK = H(ID_i || ID_{S_j} || TS_i || TS_j || RN_i || RN_j || T_{K_1} || T_{K_2} || T_{K_3})$. Here, $T_{K_2} = T_{x_j}(T_{x_i}(K_{u_i}))$ and $T_{K_3} = T_{x_j}(T_{x_i}(K_u))$. So, according to S_7, we assume $S_j \mid\equiv T_{K_2}$ and $S_j \mid\equiv T_{K_3}$.

According to the results of S_7 and S_8, we obtain $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$. **(Goal 2)**

- S_{10}: $U_i \lhd (ID_i, ID_{S_j}, \{ID_i, ID_{S_j}, \langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u), RN_j, T_{K_3}, TS_j\}_{K_2}, TS_j\langle TS_i, TS_j, RN_i, RN_j\langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u)\rangle_{T_{K_3}})$.
- S_{11}: According to Rule 5, we obtain $U_i \lhd \{ID_i, ID_{S_j}, \langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u), RN_j, T_{K_3}, TS_j\}_{K_2}$.

- S_{12}: According to A.9 and Rule 1, we have
  $U_i \mid\equiv S_j \mid\sim (ID_i, ID_{S_j}, \langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u), RN_j, T_{K_3}, TS_j)$.

- S_{13}: According to A.1 and Rule 3, we get
  $U_i \mid\equiv \#(ID_i, ID_{S_j}, \langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u), RN_j, T_{K_3}, TS_j)$.

- S_{14}: According to Rule 2, we have
  $U_i \mid\equiv S_j \mid\equiv (ID_i, ID_{S_j}, \langle T_{K_3}\rangle_{K_i}, T_{x_j}(K_u), RN_j, T_{K_3}, TS_j)$.

- S_{15}: According to Rule 5, we have
  $U_i \mid\equiv S_j \mid\equiv (\langle T_{K_3}\ \rangle_{K_i}, RN_j, T_{x_j}(K_u), TS_j)$.

- S_{16}: According to A.4 and Rule 4, we get
  $U_i \mid\equiv (RN_j, T_{x_j}(K_u), TS_j)$.

- S_{17}: According to S_{16} and Rule 5, we get
  $U_i \mid\equiv RN_j$, $U_i \mid\equiv T_{x_j}(K_u)$, $U_i \mid\equiv TS_j$. As $T_{K_3} = T_{x_i}(T_{x_j}(K_u))$ and $U_i \mid\equiv x_i$, we assume $U_i \mid\equiv T_{K_3}$.

- S_{18}: According to A.5, we get
  $U_i \mid\equiv x_i$, $U_i \mid\equiv TS_i$, $U_i \mid\equiv RN_i$, $U_i \mid\equiv T_{x_j}(K_s)$. As $T_{K_1} = T_{x_i}(T_{x_j}(K_s))$, we assume $U_i \mid\equiv T_{K_1}$.

- S_{19}: According to the proposed scheme, $SK = H(ID_i||\ D_{S_j}\ ||TS_i\ ||TS_j\ ||RN_i\ ||RN_j\ ||T_{K_1}||T_{K_2}\ ||T_{K_3})$.

Finally, according to the results of S_{17} and S_{18}, we obtain $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$. **(Goal 1)**

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

4.4.3 Informal security analysis

This section also informally shows that the proposed scheme can withstand various other known attacks.

1) Privileged-insider attack

During the user registration process, U_i sends the information T_i, $K_i = H(b_i|| R_i|| ID_i)$, C_i and RPW_i to the RC through secure channel. In the proposed scheme, PW_i of U_i is concatenated with parameter R_i, hashed biometrics key $b_i = BH(B_i)$ and the resulting value is sent using a one-way cryptographic hash function $H(\cdot)$. Therefore, guessing password PW_i from RPW_i without knowing user biometric template B_i and R_i is a computationally infeasible task. Therefore, our protocol can resist this attack successfully.

2) Password-guessing attack

U_i masks PW_i of U_i as $C_i = R_i \oplus H(b_i|| ID_{U_i}|| PW_i)$, $RPW_i = H(ID_i|| PW_i|| b_i|| R_i)$ and $A_i = H(ID_{U_i} ||RPW_i ||T_i ||T_{x_i}(K_{u_i}) ||x_i ||P)$. User smart card stores parameters C_i and A_i. Once this smart is lost or stolen by an adversary $\mathcal{A}$, by exploiting power analysis attack, the relevant information including parameters C_i and A_i are disclosed to him/her [119], [142]. However, as discussed in [55], [100] user biometrics keys are usually not susceptible to loss, theft or any forgery. Hence, as hash value of user biometrics key b_i and masked identity ID_i are unknown to $\mathcal{A}$, he/she cannot guess PW_i from C_i and A_i, which is considered to be a computationally infeasible problem. So, $\mathcal{A}$ is unable to derive PW_i from a lost or stolen smart card. The proposed scheme is then resistant to such an attack.

3) Strong user anonymity

ID_{U_i} of U_i is embedded in masked identifier $ID_i = H(ID_{U_i} ||R_i ||T_i)$, where T_i is the registration timestamp of U_i, and in future all communications happen using the masked identifier ID_i. In $C_i = R_i \oplus H(b_i|| ID_{U_i} ||PW_i)$ the original identity of U_i is used but without knowing PW_i or b_i, it is computationally infeasible to retrieve ID_{U_i} of U_i. An adversary is unable to compute ID_{U_i} even if he/she knows C_i or A_i from the smart card. Moreover, ID_{U_i} is not revealed to S_j. As a result, the proposed scheme is able to maintain strong user anonymity property.

4) Mutual authentication

U_i and S_j mutually authenticate each other without help of the RC. When S_j receives the user login message, it authenticates U_j by calculating $T'_{K_1} = T_{x_j}(T_{x_i}(K_s))$ and then checking it with the received T_{K_1}. If $T'_{K_1} = T_{K_1}$, S_j authenticates U_i as a genuine user. Similarly, when U_j receives the message from S_j, he/she also authenticates S_j by calculating $T'_{K_3} = T_{x_i}(T_{x_j}(K_u))$

and then checking it with the received T_{K_3}. If both are equal, U_i authenticates S_j. Thus, both U_i and S_j mutually authenticate each other.

5) Server spoofing attack

In a server spoofing attack, an attacker can act like a registered server and try to impersonate. The malicious server can use the message $\langle M_2 \rangle$ from previous conversation and try to authenticate user as another server. But, in the proposed scheme, a server S_j sends $E_{K_2}(ID_i || ID_{S_j} || Y || T_{x_j}(K_u) ||RN_j ||T_{K_3}) ||TS_j ||H(RN_j ||TS_j ||Y ||T_{K_3} ||T_{x_j}(K_u))$ as part of $\langle M_2 \rangle$, where $K_2 = H(T_{x_i}(K_{u_i}) ||ID_{S_j} ||ID_i ||TS_i ||TS_j ||RN_i ||T'_{K_1})$. So, to generate K_2, the attacker needs to know $T_{x_i}(K_{u_i})$ and T'_{K_1}, which is not feasible as x_i is secret to U_i and x_j is secret known to S_j. As a result, the proposed scheme is resistant to server spoofing attack.

6) Stolen-verifier attack

The RC only stores the masked identifier ID_i and respective T_i for each user U_i, where $ID_i = H(ID_{U_i} || R_i || T_i)$ and T_i is the registration timestamp of U_i. It does not store any password information for verification. Moreover, the identity of U_i is masked by $H(\cdot)$ using T_i and random number R_i. No adversary can then steal user's password and identity. Hence, the proposed scheme is resilient against such an attack.

7) Stolen smartcard attack

In the proposed scheme, using the stolen smart card an adversary $\mathcal{A}$ can masquerade as a legitimate user. $\mathcal{A}$ is unable to login to the system as user U_i needs to give his/her identity ID_{U_i}, password PW_i and his/her personal biometrics B_i. As the smart card is not tamper resistant, so $\mathcal{A}$ may extract all the stored information ID_i, T_i, A_i, $T_{x_i}(K_{u_i})$, C_i, SK_i, P, $T_{x_i}(K_u)$ and $m + m'$ server key-plus-id combinations $\{(ID_{S_j}, T_{x_j}(K_s)) \mid 1 \leq j \leq m + m'\}$. But, due to one way property of $H(\cdot)$, and difficulty of solving DLP and DHP in Chebyshev chaotic map, $\mathcal{A}$ is unable to retrieve user password, biometrics or secrets x_i and x_j. So, $\mathcal{A}$ can not misuse a stolen smart card.

8) User and server secret key leakage attack

Bergamo *et al.* [29] described an attack, where an attacker can find out an integer solution u from the equation $y = T_u(x)$ if both x and y are known with $x \in [-1, +1]$. The proposed scheme can resist such an attack in two ways. First, all the four parameters $T_{x_i}(K_s)$, $T_{x_i}(K_u)$,

$T_{x_i}(K_{u_i}$ and $T_{x_j}(K_u)$ are sent securely through symmetric key encryption $E_k(\cdot)$ or through one-way hash function $H(\cdot)$. So, an eavesdropper cannot access these parameters directly. Second, we have used the extended Chaotic maps and extended Chebyshev polynomials, where K_u, K_{u_i} and K_s are randomly chosen in the range $(-\infty, +\infty)$. According to the method explained in [29], Bergamo *et al.*'s attack can be launched only if $T_u(x)$ is known and $x \in [-1, +1]$. It is then clear that the proposed scheme can resist the Bergamo *et al.*'s attack successfully.

4.5 Simulation for formal security verification using AVISPA tool

In this section, we simulate the proposed scheme using broadly-accepted AVISPA tool [22]. We provide the implementation details of the proposed scheme in high-level protocol specification language (HLPSL) [200] and then the simulation results to show the proposed scheme is secure against replay and man-in-the-middle attacks.

4.5.1 Overview of AVISPA tool

AVISPA is considered as a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques [3], [18]. The architecture of the AVISPA tool is shown in Figure 4.6. We have used the widely-accepted AVISPA back-ends for our formal security verification [37], [62], [64], [153], [155], [165], [167], [169], [211], [212], [216].

AVISPA currently implements four back-ends and abstraction-based methods, which are integrated through the high level protocol specification language, called HLPSL [200]. A static analysis is performed to check the executability of the protocol, and then the protocol and the intruder actions are compiled into an intermediate format (IF). The intermediate format is the start point for the four automated protocol analysis techniques. IF is a lower-level language than HLPSL and is read directly by the back-ends to the AVISPA tool. The first back-end, called the On-the-fly Model-Checker (OFMC), does several symbolic techniques to explore the state space in a demand-driven way [28]. The second back-end, which is known as the CL-AtSe (Constraint-Logic-based Attack Searcher), provides a translation from any security protocol specification written as transition relation in intermediate format into a set of constraints

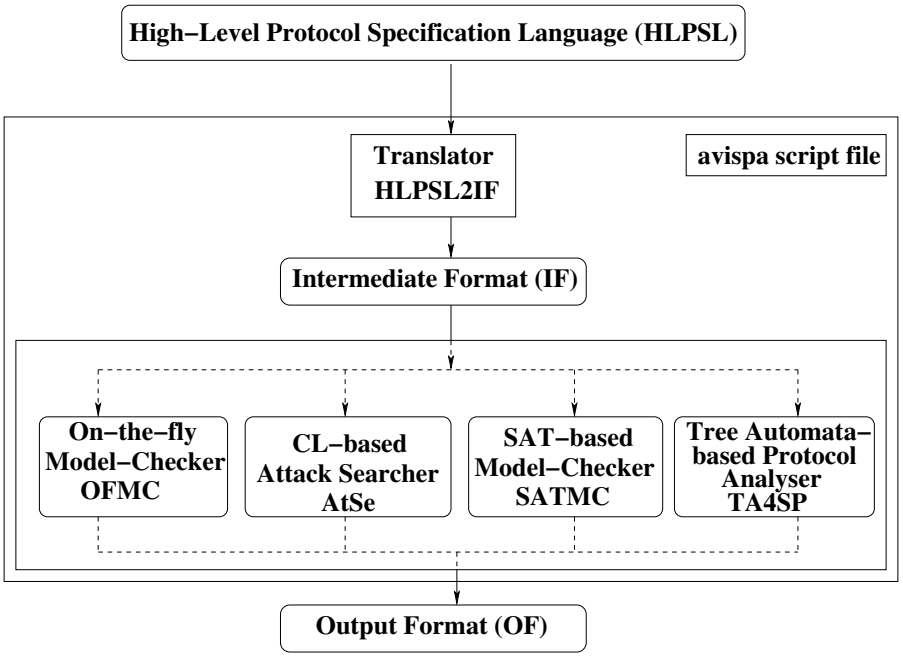


Figure 4.6: Architecture of the AVISPA tool (Source: [22]).

which are effectively used to find whether there are attacks on protocols. The third back-end, called the SAT-based Model-Checker (SATMC), builds a propositional formula which is then fed to a state-of-the-art SAT solver and any model found is translated back into an attack. The fourth back-end, known as TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), approximates the intruder knowledge by using regular tree languages.

Protocols to be implemented by the AVISPA tool have to be specified in HLPSL (High Level Protocols Specification Language) [200], and written in a file with extension hlpsl. This language is based on roles: basic roles for representing each participant role, and composition roles for representing scenarios of basic roles. Each role is independent from the others, getting some initial information by parameters, communicating with the other roles by channels. The intruder is modeled using the Dolev-Yao model [70] (as described in our threat model in Section 1.1) with the possibility for the intruder to assume a legitimate role in a protocol run. The role system also defines the number of sessions, the number of principals and the roles.

The output format (OF) of AVISPA is generated by using one of the four back-ends explained above. When the analysis of a protocol has been successful (by finding an attack or not), the output describes precisely what is the result, and under what conditions it has been obtained. In OF, there are the following sections.

- The first printed section SUMMARY indicates that whether the tested protocol is safe, unsafe, or whether the analysis is inconclusive.
- The second section, called DETAILS either explains under what condition the tested protocol is declared safe, or what conditions have been used for finding an attack, or finally why the analysis was inconclusive.
- Other sections such as PROTOCOL, GOAL and BACKEND are the name of the protocol, the goal of the analysis and the name of the back-end used, respectively.
- Finally, after some comments and statistics, the trace of an attack (if any) is also printed in the standard Alice-Bob format.

Some of the basic types supported by HLPSL are as follows:

- *agent:* Values of type *agent* represent principal names. The intruder is always assumed to have the special identifier i.
- *public_key:* Variables of this type represent agents' public keys in a public-key cryptosystem. For example, given a public (respectively private) key pk, its inverse private (respectively public) key is obtained by inv_pk.
- *symmetric_key:* Variables of this type represent keys for a symmetric-key cryptosystem.
- *text: text* values are often used as nonces. These values can be used for messages. If Na is of type *text (fresh)*, then Na' will be a fresh value which the intruder cannot guess.
- *nat: nat* type represents the natural numbers in non-message contexts.
- *const:* It represents constants.
- *hash_func:* The base type *hash_func* represents cryptographic hash functions. The base type function also represents functions on the space of messages. It is assumed that the intruder cannot invert hash functions (in essence, that they are one-way collision-resistant functions).

The space of legal messages are defined as the closure of the basic types. For a given message M and encryption key K, $\{M\}\_K$ refers to as the symmetric/public-key encryption. The associative "." operator is always used for concatenations.

```
role alice (Ui, Sj, RC   : agent,
   SKij  : symmetric_key, MKj :symmetric_key,
          MKi :symmetric_key, H : hash_func,
          Snd, Rcv: channel(dy))
played_by Ui
def=
 local State  : nat,
 IDi, IDui, IDsj, Ci,RPWi,TSj,TSi, Ti,Bi, Ri, PWi,
 Bbi, Kui, Ku, Xj, Xi, Ks, TXiKui, TXiKu, SKi, P, Ai,TXjKs,
 TXjKu,TXiKs, Ki,Tk1, Tk2, Tk3, RNi, RNj, K1, K2, Y: text,
 BH : hash_func
const alice_bob, bob_alice, sub1, sub2, sub3, sub4, sub5, sub6: protocol_id
init  State := 0
transition
1.State = 0  /\ Rcv(start)=|>
State' := 1  /\ IDi':=H(IDui.Ti) /\ Bbi' := BH(Bi)
           /\ RPWi' :=H(IDi'.PWi.Bbi.Ri) /\ Ki':= H(IDi'.Bi.Ri)
           /\ Ci':= xor(Ri, H(Bi.IDui.PWi))
           /\ Snd({IDi'.Ki'.xor(Ri, H(Bbi.IDui.PWi)).Ti.RPWi'}_MKi)
           /\ secret({PWi}, sub1, Ui) /\ secret({MKi}, sub2, {RC,Ui})
           /\ secret({IDui}, sub3, {RC,Ui}) /\ secret({Bi}, sub4, {RC,Ui})
           /\ secret({Ri}, sub5, Ui)
2. State = 1 /\ Rcv({H(IDui.Ti).Ti.H(IDui.H(IDi'.PWi.BH(Bi).Ri).
              Ti.H(Xi'.Ku').Xi'.xor(Ks, H(Xi'.H(IDi'.BH(Bi).Ri)))).
              H(Xi'.Kui').xor(Ri, H(BH(Bi).IDui.PWi)).
              xor(H(IDi'.BH(Bi).Ri), Xi').
              xor(Ks, H(Xi'.H(IDi'.BH(Bi).Ri))).H(Xi'.Ku').
              IDsj.H(Xj'.Ks')}_MKi) =|>
State' := 2 /\ Tk1':= H(Xi'.H(Xj'.Ks'))
           /\ Ks':= xor(H(Xi'.H(IDi'.BH(Bi).Ri)),
                 xor(Ks, H(Xi'.H(IDi'.BH(Bi).Ri))))
           /\ TXiKs':= H(Xi'. Ks') /\ TSi':= new() /\ RNi':= new()
           /\ K1':= H(H(Xj'.Ks').IDi'.IDsj.TSi')
           /\ Snd(IDi'.IDsj.{IDi'.IDsj.Tk1'.H(Xi'.Ks').H(Xi'.Kui').
              H(Xi'.Ku').RNi'.H(IDi'.BH(Bi).Ri)}_K1'.TSi'.
              H(H(IDi'.BH(Bi).Ri).TSi'.IDi'))
           /\ witness(Ui, Sj, alice_bob, TSi')
3. State = 2 /\ Rcv(H(IDui.Ti).IDsj.{H(IDui.Ti).IDsj.xor(H(H(IDui.Ti).
              BH(Bi).Ri),H(Xj'.H(Xi'.Kui'))).H(Xj'.H(Xi'.Ku')).H(Xj'.Ku').
              RNj'}_H(H(Xi'.Kui').H(IDui.Ti).IDsj.TSi'.TSj'.RNi'.
              H(Xi'.H(Xj',Ks'))).TSj'.H(RNj'.RNi'.TSi'.TSj'.
              xor( H(H(IDui.Ti).BH(Bi).Ri),
              H(Xj'.H(Xi'.Kui'))).H(Xj'.H(Xi'.Ku')).H(Xj'.Ku'))=|>
State' := 3 /\ SKij':= H(H(IDui.Ti).IDsj.TSi.TSj.RNi.RNj.H(Xi'.H(Xj'.Ks')).
                  H(Xj'.H(Xi'.Kui')).H(Xj'.H(Xi'.Ku')))
   /\ secret({SKij'}, sub6, {Ui,Sj}) /\ request(Sj, Ui, bob_alice, TSj')
 end role
```

Figure 4.7: Role specification in HLPSL for U_i.

The "*played_by A*" declaration indicates that the agent named in variable A will play in a specific role. A knowledge declaration (generally in the top-level *Environment* role) is used to specify the intruder's initial knowledge. Immediate reaction transitions have the form $X = | > Y$, which relate an event X and an action Y, and it indicates that whenever we take a transition that is labeled in such a way as to make the event predicate X true, we must immediately (that is, simultaneously) execute action Y. If a variable V needs to be permanently secret, it is expressed by the goal *secrecy_of* V. Therefore, if V is ever obtained or derived by the intruder, a security violation will result.

```
role bob (Ui, Sj, RC   : agent,
SKij  : symmetric_key,
        MKj :symmetric_key,
        MKi :symmetric_key,
H : hash_func,
        Snd, Rcv: channel(dy))
played_by Sj
def=
local State  : nat,
 IDi, IDui, IDsj, Ci, RPWi, TSj, TSi, Ti, Bi, Ri,
 PWi, Kui, Ku, Xj, Xi, Ks, TXiKui, TXiKu, SKi, P,
 Ai, TXjKs, TXjKu, TXiKs, Ki, Tk1, Tk2, Tk3,
 RNi, RNj, K1, K2, Y:  text,
 BH: hash_func
const alice_bob, bob_alice, sub1, sub2,
      sub3, sub4, sub5, sub6: protocol_id
init  State := 0
transition
1. State   = 0 /\ Rcv({IDsj.H(Xj'.Ks').H(Xj'.Ku').Xj'}_MKj) =|>
   State' := 1 /\ RNj':= new()
2. State   = 1 /\ Rcv(H(IDui.Ti).IDsj.{H(IDui.Ti).IDsj.
                H(Xi'.H(Xj'.Ks')).H(Xi'.Ks').H(Xi'.Kui').
                H(Xi'.Ku').RNi'.H(H(IDui.Ti).BH(Bi).Ri)}_(H(H(Xj'.Ks').
                H(IDui.Ti).IDsj.TSi')).TSi'.H(H(H(IDui.Ti).BH(Bi).Ri).
                TSi'.H(IDui.Ti))) =|>
State' := 2 /\ RNj':= new()
          /\ K1':= H(H(Xj'.Ks').H(IDui.Ti).IDsj.TSi')
          /\ Tk2':= H(Xj'.H(Xi'.Kui'))
          /\ Y':= xor(H(H(IDui.Ti).BH(Bi).Ri),Tk2')
          /\ TSj':= new()
          /\ K2':= H(H(Xi'.Kui').H(IDui.Ti).IDsj.TSi'.TSj'.
                RNi'.H(Xi'.H(Xj'.Ks')))
          /\ Tk3':= H(Xj'.H(Xi'.Ku'))
          /\ Snd(H(IDui.Ti).IDsj.{H(IDui.Ti).IDsj.Y'.Tk3'.
                H(Xj'.Ku').RNj'}_K2'.TSj'.H(RNj'.RNi'.TSi'.
                TSj'.Y'.Tk3'.H(Xj'.Ku')))
          /\ witness(Sj, Ui, bob_alice, TSj')
          /\ SKij':= H(H(IDui.Ti).IDsj.TSi.TSj.RNi.RNj.
                   H(Xi'.H(Xj'.Ks')).H(Xj'.H(Xi'.Kui')).
                   H(Xj'.H(Xi'.Ku')))
   /\ secret({PWi}, sub1, Ui) /\ secret({MKi}, sub2, {RC,Ui})
          /\ secret({IDui},sub3,{RC,Ui}) /\ secret({Bi}, sub4, {RC,Ui})
          /\ secret({Ri}, sub5, Ui) /\ secret({SKij}, sub6, {Ui,Sj})
          /\ request(Ui, Sj, alice_bob, TSi')
end role
```

Figure 4.8: Role specification in HLPSL for S_j.

4.5.2 HLPSL specification of the proposed scheme

We consider the basic roles: alice, rc and bob, which correspond to the participants: user U_i, RC and serverS_j, respectively. Besides these roles, we have also the roles for the session and environment, which are mandatory in AVISPA. More details on AVISPA tool and HLPSL specifications can be found in [22], [40], [56], [57], [58], [152], [200].

```
role rc (Ui, Sj, RC : agent,
SKij  : symmetric_key,
       MKj :symmetric_key,
       MKi :symmetric_key,
       H : hash_func,
       Snd, Rcv: channel(dy))
played_by RC
def=
local State  : nat,
     IDi, IDui, IDsj, Ci, RPWi, TSj, TSi, Ti, Bi,
     Ri, PWi, Kui, Ku, Xj, Xi, Ks, TXiKui, TXiKu,
     TXiKs, SKi, P, Ai, TXjKs, TXjKu, Ki, Tk1, Tk2, Tk3,
     RNi, RNj, K1, K2, Y:  text,
     BH: hash_func
const alice_bob, bob_alice, sub1, sub2, sub3,
     sub4, sub5, sub6: protocol_id
 init  State := 0
transition
1. State  = 0 ∧ Rcv({H(IDui.Ti).H(H(IDui.Ti).BH(Bi).Ri).
                    xor(Ri, H(BH(Bi).IDui.PWi)).Ti.
                    H(H(IDui.Ti).PWi.BH(Bi).Ri)}_MKi) =|>
State'= 1 ∧ Xi'= new() ∧ Kui'= new()
         ∧ Ku'= new()
% Parameters for user Ui
         ∧ Xj'= new() ∧ Ks'= new()
% Parameters for server Sj
         ∧ TXiKui'= H(Xi'.Kui') ∧ TXiKu'= H(Xi'.Ku')
         ∧ SKi'= xor(H(H(IDui.Ti).BH(Bi).Ri), Xi')
         ∧ P'= xor(Ks, H(Xi'.Ki))
         ∧ Ai'= H(IDui.H(H(IDui.Ti).PWi.BH(Bi).Ri).Ti.TXiKu'.Xi'.P')
         ∧ Snd({H(IDui.Ti).Ti.Ai'.TXiKui'.
            xor(Ri, H(BH(Bi).IDui.PWi)).SKi'.P'.TXiKu'.IDsj.TXjKs'}_MKi)
         ∧ TXjKs'= H(Xj'.Ks')
% Compute chebyshev polynomials TXj(Ks) and TXj(Ku)
         ∧ TXjKu'= H(Xj'.Ku')
         ∧ Snd({IDsj.TXjKs'.TXjKu'.Xj'}_MKj)
         ∧ secret({PWi}, sub1, Ui)
         ∧ secret({MKi}, sub2, {RC,Ui})
         ∧ secret({IDui}, sub3, {RC,Ui})
         ∧ secret({Bi}, sub4, {RC,Ui})
         ∧ secret({Ri}, sub5, Ui)
end role
```

Figure 4.9: Role specification in HLPSL for RC.

In Figure 4.7, we have implemented the role for U_i in HLPSL. In user registration phase, U_i first initiates the communication by sending the registration request $\langle ID_i, T_i, K_i, C_i, RPW_i \rangle$ securely to the RC using the $Snd(\,)$ channel. The type declaration $channel\,(dy)$ denotes a Dolev-Yao threat model channel in which the attacker can read, modify or delete the message content. U_i waits for the smart card containing the information $\{ID_i, T_i, A_i, T_{x_i}(K_{u_i}), T_{x_i}(K_u), C_i, SK_i, P, (ID_{S_j}, T_{x_j}(K_{s_j}))\}$, $1 \leq j \leq m+m'\}$ securely from the RC using the $Rcv(\,)$ channel. After successful login, U_i sends the login request $\langle M_1 \rangle$ to the respective server S_j. U_i then waits

```
role session(Ui, Sj, RC : agent,
SKij : symmetric_key,
       MKj :symmetric_key,
       MKi :symmetric_key,
H : hash_func)
def=
  local  SI, SJ, RI, RJ,BI,BJ: channel (dy)
composition
    alice(Ui, Sj, RC,SKij,MKi,MKj, H, SI, RI)
/\  rc(Ui, Sj, RC,SKij,MKi,MKj, H, BI, BJ)
/\  bob(Ui, Sj, RC,SKij,MKi,MKj, H, SJ, RJ)
end role

role environment()
def=
  const ui,sj,rc : agent, skij: symmetric_key,
      mkj: symmetric_key, mki:symmetric_key,
      h, bh   : hash_func,
      idi,idui,idsj, ci,rpwi,tsj,tsi, ti,bi,
      ri, pwi, kui,ku,xj,xi,ks,txikui,txiku,ski,p,
      ai,txjks,txjku,txiks,ki,tk1,tk2,tk3,rni,rnj,
      k1,k2,y : text,
      alice_server, rc_bob, bob_alice, sub1, sub2,
      sub3, sub4, sub5, sub6 : protocol_id
intruder_knowledge = {ui, sj, idi, h, bh}
  composition
session(ui,sj,rc,skij,mkj,mki,h) /\
session(i,sj,rc,skij,mkj,mki,h) /\
session(ui,i,rc,skij,mkj,mki,h) /\
       session(ui,sj,i,skij,mkj,mki,h)
end role

goal
  secrecy_of sub1, sub2, sub3, sub4, sub5, sub6
  authentication_on alice_bob
  authentication_on bob_alice
end goal
environment()
```

Figure 4.10: Role specification in HLPSL for the session, goal and environment.

for successful authentication replay message $\langle M_2 \rangle$ from S_j via open channel $Rcv(\,)$. Similarly, we have defined the roles for the S_j and RC in HLPSL in Figures 4.8 and 4.9, respectively.

We have finally given the specifications in HLPSL for the roles of session, goal and environ-

ment in Figure 4.10. In the session segment, the basic roles: alice, rc and bob are treated as instances with concrete arguments. The top-level role (environment) defines in the specification of HLPSL. The intruder (i) can also participate in the execution of protocol as a concrete session. Six secrecy goals and three authentication goals are verified (shown in Figure 4.10).

4.5.3 Analysis of results

We choose the back-end OFMC [28] for an execution test and a bounded number of sessions model checking. Executability check for non-trivial HLPSL specifications: It is often the case that, due to some modeling mistakes, the protocol model cannot execute to completion. The back-ends might not find any attack if the protocol model cannot reach the state where the attack can happen. Therefore, an executability check is very important [200].

- **Replay attack check:** OFMC will first check whether the honest agents can execute the protocol by performing a search of a passive intruder, and then give the intruder the knowledge of some "normal" sessions between honest agents [22]. The test results show that the proposed multi-server authentication protocol can resist against replay attack.

- **DelovYao model check:** At last, we choose the depth for the search is seven and output of model checking results are shown in Figure 4.11. As shown in the figure, there are totally 8 nodes have been searched in 0.12s. From the results, we can conclude that the proposed protocol can fulfill the design properties and it is secure under the test of AVISPA using the OFMC back-end with bounded number of sessions.

We have chosen the broadly-used the On-the-fly Model-Checker (OFMC) [28] and Constraint Logic based Attack Searcher (CL-AtSe) backends for the execution test in order to find whether there are any attacks on the proposed scheme. For the replay attack checking, the back-ends check whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After that the back-ends supply the intruder the knowledge of some normal sessions between the legitimate agents. All public parameters are known to the intruder. For the Dolev-Yao model checking, the back-ends verify whether there is any man-in-the-middle attack possible by the intruder.

It is worth noticing that the proposed scheme uses the bitwise XOR operations. At present, other backends, namely SATMC and TA4SP do not support this feature to implement bitwise XOR operations in the defined roles. As a result, the simulation results of the proposed scheme under both the SATMC and TA4SP backends will come as "inconclusive". Due to

this reason, we have ignored these simulation results for formal security verification in this chapter.

We have simulated the proposed scheme using SPAN, the Security Protocol ANimator for AVISPA [23], for both OFMC and CL-AtSe backends. The simulation results of the analysis using OFMC and CL-AtSe backends are shown in Figure 4.11. Under OFMC backend, the parse time is 0.01 seconds and the search time is 0.12 seconds for visiting eight nodes with a depth of three plies. On the other hand, under CL-AtSe backend, three states are analyzed with the translation time of 0.17 seconds. It is evident that the proposed scheme is safe against replay and man-in-the-middle attacks.

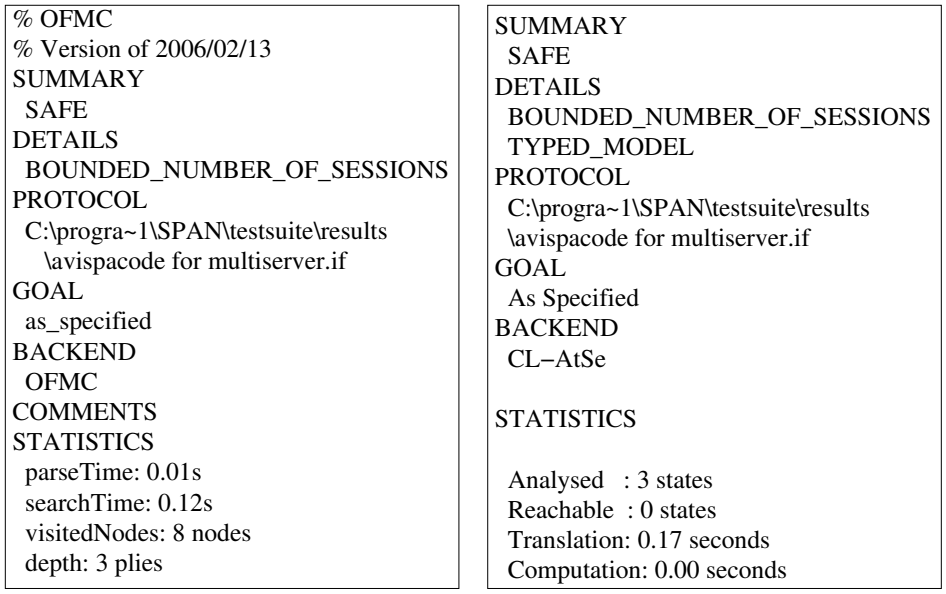


Figure 4.11: Analysis of simulation results using OFMC and CL-AtSe backends.

4.6 Performance comparison

Generally for any authentication protocol, the registration phase is only one-time process. Therefore, we consider the complexity of the login and authentication phases in terms of communication and computation.

4.6.1 Communication costs comparison

We assume that the identity is 32 bits, the timestamp is 32 bits, the output size of hash function $H(\cdot)$ is 160 bits (if we use SHA-1 hash function [6]) and the block size of symmet-

ric encryption/decryption (for example, *AES* [2]) is 128 bits. Since registration phases are executed only once, we concentrate on login and authentication phases for calculation of communication and computation costs. During these two phases, in the proposed scheme only two messages are involved in communication. U_i sends login request $M_1 = \{ID_i, ID_{S_j}, E_{K_1}(ID_i \ ||ID_{S_j} \ ||T_{K_1} \ ||T_{x_i}(K_s) \ ||T_{x_i}(K_u) \ ||T_{x_i}(K_{u_i}) \ ||RN_i \ ||K_i), TS_i, H(K_i \ ||TS_i \ ||ID_i \ ||ID_{S_j} \ ||RN_i \ ||T_{x_i}(K_u) \ ||T_{K_1})\}$ to S_j, which needs $(32+ \ 32+ \ \lceil 384/128 \rceil \times 128 \ +32+160) = 640$ bits. S_j sends the authentication reply $M_2 = \{ID_i, ID_{S_j}, E_{K_2}(ID_i \ ||ID_{S_j} \ ||Y \ ||T_{x_j}(K_u) \ ||RN_j \ ||T_{K_3}), TS_j, H(TS_i \ ||TS_j \ ||RN_i \ ||RN_j \ ||Y \ ||T_{K_3} \ ||T_{x_j}(K_u))\}$ to U_i, which again requires 640 bits. Therefore, during the login and authentication phases, the total communication cost required is $(640 + 640) = 1280$ bits. Note that the proposed scheme does not involve the *RC* in both the login and authentication phases. Only two messages communication are required. Table 4.4 shows the comparative study of communication costs among the proposed scheme and other related recently proposed schemes during the login and authentication phases. All the other related schemes except the proposed scheme need five messages communication. The communication costs of Odelu *et al.*'s scheme, He-Wang's scheme, Yoon *et al.*'s scheme and Kim *et al.*'s scheme are 2944 bits, 3520 bits, 2496 bits and 2496 bits, respectively. So, it is clear that compared to other related existing schemes, the proposed scheme consumes the minimum communication cost.

Table 4.4: Communication costs comparison among the proposed scheme and recent multi-server authentication schemes.

Scheme	I_1	I_2
Li *et al.* [131]	7232	7
Yoon *et al.* [230]	2496	5
Kim *et al.* [115]	2496	5
He-Wang [88]	3520	5
Odelu *et al.* [153]	2944	5
Our	1280	2

Note: I_1 : total number of bits transmission required during login, and authentication and session key establishment phases of the schemes; I_2 : total number of messages transmission required during login, and authentication and session key establishment phases of the schemes.

4.6.2 Computation costs comparison

Table 4.5 shows the execution times needed for various cryptographic operations. Let T_h, $T_{enc/dec}$, T_M, T_{Ch} and T_{bh} denote the time to execute a one-way hash function $H(\cdot)$, a symmetric encryption/decryption (using AES-128 symmetric cryptosystem), an elliptic curve point multiplication and $T_n(x) \pmod p$ in Chebyshev polynomial using the algorithm provided in [118], and a biohashing operation, respectively. The results shown in Table 4.5 are based on an experiment conducted on an Intel Pentium4 2600 MHz processor with 1024 MB RAM in [118]. We have $T_h \approx 0.0005s$, $T_M \approx 0.063075s$, $T_{enc/dec} \approx 0.0087s$ and $T_{Ch} \approx 0.02102s$. We assume that $T_{bh} \approx T_{Ch}$. We have ignored the computation cost of bitwise XOR operation as it is significantly low compared to other operations. We have compared the computation costs of the proposed scheme with other related existing schemes in Table 4.6 during the login and authentication phases.

Table 4.5: Execution timings of various cryptographic operations.

Term	Description of operation	Time taken (in seconds)
T_h	one-way cryptographic hash function	0.0005
$T_{enc/dec}$	symmetric key encryption/decryption	0.0087
T_M	elliptic curve point multiplication	0.063075
T_{Ch}	Chebyshev polynomial computation	0.02102
T_{bh}	biohashing	0.02102

Table 4.6: Comparison on computation cost among different schemes.

	Computation cost	Execution time (in milliseconds)
Li *et al.* [131]	$19T_h + 6T_M$	387.95
Kim *et al.* [115]	$15T_h + 4T_M$	259.80
Yoon *et al.* [230]	$15T_h + 4T_M$	259.80
He-Wang [88]	$21T_h + 8T_M$	515.10
Odelu *et al.* [153]	$6T_{enc/dec} + 24T_h + 6T_M$	442.65
Our	$8T_{enc/dec} + T_{bh} + 16T_h + 6T_{Ch}$	224.74

Note that the server and user registration phases are executed only once. Thus, we concentrate only on login and authentication phases for measuring the computation overhead.

Table 4.7: Comparison of functionality features among different schemes.

Scheme	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}
Li *et al.* [131]	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes	No	No	Yes	No	No	No
Yoon *et al.* [230]	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	No	Yes	No	No	No
Kim *et al.* [115]	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	No	Yes	No	No	No
He-Wang [88]	Yes	No	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No	Yes	No	No	No
Odelu *et al.* [153]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Our	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note : I_1 : whether provides mutual authentication or not; I_2 : whether provides flawless password change phase or not; I_3 : whether resists server spoofing attack or not; I_4 : whether resists man-in-the-middle attack/replay attack or not; I_5 : whether resists privileged-insider attack or not; I_6 : whether resists lost smart card attack or not; I_7 : whether provides strong user anonymity or not; I_8 : whether resists session-specific temporary information attack or not; I_9 : whether resists DoS attack or not; I_{10} : whether provides perfect forward secrecy or not; I_{11} : whether provides session key security or not; I_{12} : whether retains provision for revocation and re-registration or not; I_{13} : whether able to execute without identity-verification table or not; I_{14} : whether involves low computation cost or not; I_{15} : whether involves low communication cost or not; I_{16} : whether able to establish session key without involving the RC or not.
Yes: the scheme is secure or it supports a feature; **No:** the scheme is not secure or it does not support the feature;

During the login and authentication phases, the total computation cost per user in the proposed scheme is $123.88\,ms$, where it is $100.86\,ms$ per server. From this table, it is evident that our lightweight authentication scheme requires less computation overhead as compared to other related schemes. Finally, the total computation cost (including server side and user side) of the proposed scheme is $224.74\,ms$, which is minimum among other related existing schemes.

4.6.3 Security and functionality features comparison

Finally, in Table 4.7, we have tabulated an overall security and functionality features comparison among the proposed scheme and other schemes. It is observed that the proposed scheme outperforms other recently proposed existing schemes as the proposed scheme is secure and supports extra features. Furthermore, the proposed scheme has the lowest computation and

communication overheads. In addition, the proposed scheme does not involve the RC during the login, and authentication and key establishment phases.

4.7 Summary

We have designed a new Chebyshev chaotic map based lightweight multi-server authentication scheme. We have used the random oracle model and the BAN logic for formal security analysis, and also simulated the proposed scheme using the widely-accepted AVISPA tool for the formal security verification. The results show that the proposed scheme is secure from well-known possible attacks required in a multi-server environment. Moreover, the proposed scheme fulfills known functionality features applicable for a multi-server environment. The proposed scheme does not require involvement of the RC during the login, and authentication and key-establishment phase. As a consequence, the proposed scheme is efficient and more suitable for practical applications especially for mobile and battery powered devices as compared to other existing schemes.

Chapter 5

Biometric-Based Anonymous User Authentication for Crowdsourcing Internet of Things

The recent proliferation of mobile devices such as smartphones and wearable devices has given rise to crowdsourcing Internet of Things (IoT) applications. E-healthcare service is one of the important services for the crowdsourcing IoT applications that facilitates remote access or storage of medical server data to the authorized users (for example, doctors, patients and nurses) via wireless communication. In some situations, sharing of the patient information in a protected online environment with a group of medical professionals is very much essential, and for these types of treatments where multiple professionals are involved, crowdsourcing IoT in e-healthcare services is required. Unfortunately, ever-growing use of the Internet offers malicious users and attackers ample opportunity to gain unauthorized illegal access of medical data by exploiting various kinds of network and information attacks. To protect important and private medical information, design of a secure remote user authentication protocol for crowdsourcing in e-healthcare services needs more attention from the researchers.

In this chapter, we present a three-factor, extended chaotic map based secure and efficient remote user authentication scheme for crowdsourcing IoT environment. Prior to this, we state our basic research contribution and then describe the threat model in brief. We then describe different phases of the proposed scheme.

5.1 Research contributions

The main contributions of this chapter are listed below:

- We present a new robust, secure as well as efficient remote authentication scheme that uses the extended chaotic map, user biometrics, password and user smart card simultaneously.
- The proposed scheme has low computation and communication costs as compared to those for the existing related schemes.
- We also introduce an efficient mechanism for revocation of lost smart card of a legitimate user.
- Through the combined formal security analysis using the ROR model, BAN logic and formal security verification through simulation using the broadly-accepted ProVerif 1.93 tool, and also informal security analysis, we prove that the proposed scheme has the ability to resist various known attacks.

5.2 Threat model

We adopt the widely-accepted Dolev-Yao threat model (DY model) [70]. In the DY model, any two nodes communicate over an insecure channel. An adversary $\mathcal{A}$ has the ability to eavesdrop, modify or delete the messages transmitted between two participants. In addition, we assume that $\mathcal{A}$ can extract all the sensitive information stored in the lost/stolen smart card of a legal user U_i using the power analysis attacks [119], [142].

5.3 The proposed scheme

In this section, we present the biometrics and fuzzy extractor based user authentication protocol (BFE-UAP). The proposed scheme has five phases, namely 1) registration, 2) login, 3) authentication and key establishment, 4) password change and 5) revocation of lost smart card. For describing and analyzing the proposed scheme, we use the notations listed in Table 5.1.

The various phases related to the proposed scheme are given in subsequent sections.

Table 5.1: Notations used in this chapter.

Symbol	Description
S	Remote medical server in IoT environment
U_i	User of the system
ID_i	User identity
SC	Smart card of U_i
$H(\cdot)$	A one way cryptographic hash function
$T_x(\cdot)$	A Chebyshev polynomial
b	128-bit random number selected by U_i
mk	1024-bit master secret key of S
RN_u	128-bit random number of U_i
RN_s	128-bit random number of S
SN_i	Identity or serial number of a smart card SC
$\|\|, \oplus$	Concatenation, bitwise XOR operations
$A \xrightarrow{\langle M \rangle} B$	Entity A sends message M to entity B
TS_1, TS_2	Current timestamps
$\triangle T$	Maximum transmission delay
$Gen(\cdot)$	Fuzzy extractor probabilistic generation function
$Rep(\cdot)$	Fuzzy extractor deterministic reproduction function
τ	Permissible error tolerance value used in $Rep(\cdot)$ function.

- **Registration phase:** Through the registration phase, a legal user U_i obtains his/her smart card SC from the medical server S. In this phase, the communications between U_i and S take place over a secure channel (e.g. in person) as it a one-time process.

- **Login phase:** In order to access the services from S, U_i must login to the system. In this phase, user smart card accepts necessary credentials from U_i and verifies his/her authenticity. If verification is successful, U_i is allowed to send login request message to S_j via a public channel.

- **Authentication and key establishment phase:** In this phase, S receives the login request message from U_i and mutually authenticates each other. After successful mutual authentication, U_i and S establish a common secret session key which is used for future

secure communications between them.

- **Password and biometric change phase:** In this phase, a valid user U_i can update his/her old password PW_i with a new password PW_i' locally without involving the server S anymore:

- **Lost smartcard revocation phase:** In the threat model mentioned in Section 5.2 that if an attacker obtains a user's smart card SC, he/she can obtain all the smart card information by executing power analysis attacks [119], [157]. So, if any protocol is vulnerable to off-line password guessing attack or stolen smart card attack, the adversary may impersonate the legal user U_i to login to the server S using the lost or stolen smartcard and guess old password. In this situation, it is quite necessary to provide U_i with a new smart card and S should be able to discriminate between the old lost/stolen smartcard and the newly issued smartcard.

5.3.1 Registration phase

The following steps are involved in this phase:

- **Step 1.** U_i chooses his/her identity ID_i, password PW_i, personal biometrics $\mathcal{B}_i$ and 128-bit random number b.

- **Step 2.** Using fuzzy extractor generation procedure, U_i generates $(\alpha_i, \beta_i) = Gen(\mathcal{B}_i)$, and computes masked password $RPW_i = H(H(ID_i \,||PW_i) \,||\alpha_i)$ and $C = H(H(ID_i \,||PW_i \,||b) \,||\alpha_i)$. Here, α_i is the biometric secret key of U_i and β_i is the public reproduction parameter. U_i submits $\langle ID_i, C\rangle$ to S via a secure channel.

- **Step 3.** S receives the registration request $\langle ID_i, C\rangle$ from U_i and selects a 1024-bit number mk as its secret master key, which is known to this server only. Further, S selects 128-bit random number r and computes the following: $X = H(H(ID_i||mk) \oplus r)$ and $D_1 = X \oplus C = X \oplus H(H(ID_i \,||PW_i \,||b)||\, \alpha_i)$. S embeds the parameters $\{D_1, T_{mk}(X)\}$ in user smartcard SC and issues this smart card to U_i through a secure channel. S saves pair $\langle ID_i, SN_i, r\rangle$ into its database, where SN_i is the identity or serial number of the smart card SC.

- **Step 4.** U_i receives the smart card SC from S and computes $D_2 = RPW_i \oplus b$ and $f_i = H(RPW_i||b)$. Finally, U_i stores the information β_i, D_2, f_i, $H(\cdot)$, $Gen(\cdot)$, $Rep(\cdot)$ and τ into SC, where τ is the permissible error tolerance value used in $Rep(\cdot)$ function.

Figure 5.1 shows the steps of registration phase involved in the proposed scheme.

User (U_i)	Server (S)
Select ID_i, PW_i, biometrics $\mathcal{B}_i$ and random number b. Compute $(\alpha_i, \beta_i) = Gen(\mathcal{B}_i)$, $RPW_i = H(H(ID_i\|\|PW_i)\|\|\alpha_i)$, $C = H(H(ID_i\|\|PW_i\|\|b)\|\|\alpha_i)$. $\xrightarrow[\textbf{(secure channel)}]{\langle ID_i, C\rangle}$	
	Generate random number r and master secret key mk. Compute $X = H(H(ID_i\|\|mk) \oplus r)$, $T_{mk}(X)$, $D_1 = X \oplus C$. $\xleftarrow[\textbf{(secure channel)}]{\textbf{Smartcard}\{D_1, T_{mk}(X)\}}$
Compute $D_2 = RPW_i \oplus b$, $f_i = H(RPW_i\|\|b)$. Store β_i, D_2, f_i, $H(\cdot)$, $Gen(\cdot)$, $Rep(\cdot)$, and τ into smartcard SC.	Store $\langle ID_i, SN_i, r\rangle$ in server database.

Figure 5.1: User registration phase of the proposed scheme.

5.3.2 Login phase

For login purpose, the following steps need to be executed:

- **Step 1.** U_i inserts SC and inputs his/her identity ID_i, password PW_i and imprints personal biometrics $\mathcal{B}_i'$ at the sensor of a particular terminal. Using reproduction procedure and stored β_i, SC computes $\alpha_i = Rep(\mathcal{B}_i', \beta_i)$, $RPW_i = H(H(ID_i||PW_i)||\alpha_i)$ and generates $b' = D_2 \oplus RPW_i$.

- **Step 2.** Using the generated b', SC computes $f_i' = H(RPW_i||b')$ and checks if $f_i' = f_i$ holds or not. A mismatch results in immediate termination of the login phase. Otherwise, it is ensured that the user has entered correct identity, password and biometric information, and SC proceeds to compute $X = D_1 \oplus H(H(ID_i\ ||PW_i\ ||b')$

$||\alpha_i) = H(H(ID_i\ ||mk)\ \oplus r)\ \oplus H(H(ID_i\ ||PW_i\ ||b)\ ||\alpha_i)\ \oplus H(H(ID_i\ ||PW_i\ ||b')\ ||\alpha_i)$ $= H(H(ID_i\ ||mk)\ \oplus r)$, since $b = b'$.

- **Step 3.** SC generates a 128 bit random number u. Further, it generates $T_u(X)$ and using the smart card parameter $T_{mk}(X)$, it generates $KA = T_u(T_{mk}(X))$. SC then generates a 128-bit random number RN_u and computes $M_1 = X \oplus\ RN_u \oplus\ TS_1 \oplus\ T_u(X) = H(H(ID_i\ ||mk)\ \oplus r)\ \oplus RN_u\ \oplus TS_1\ \oplus T_u(X)$, $DID_i = ID_i\ \oplus H(KA)$ and $M_u = H(ID_i||\ X||\ KA||\ RN_u||\ TS_1)$, where TS_1 is the current timestamp of the user U_i's system. Finally, U_i sends the login request $\{DID_i,\ T_u(X),\ M_1,\ M_u,\ TS_1\}$ to S via a public channel.

5.3.3 Authentication and key establishment phase

The following steps are needed in this phase:

- **Step 1.** S receives the user login message at time ${TS_1}^*$ and verifies whether $|{TS_1}^*\ -TS_1| \leq \triangle T$, where $\triangle T$ is the maximum transmission delay. If the verification does not hold, S rejects this phase immediately. Otherwise, S computes $KA' = T_{mk}(T_u(X))$, $ID_i' = DID_i\ \oplus H(KA') = ID_i\ \oplus H(KA)\ \oplus H(KA') = ID_i$. If $KA' = KA$, it ensures that $ID_i' = ID_i$. S searches for the pair $\langle ID_i, r\rangle$ in its database. If this pair is found, using parameter r, computed identity ID_i' and master secret key mk, S generates $X' = H(H(ID_i'||mk) \oplus r)$, and further computes $M_2 = M_1 \oplus\ TS_1 \oplus\ X' \oplus\ T_u(X)$ $= (H(H(ID_i\ ||mk)\ \oplus r)\ \oplus RN_u\ \oplus TS_1\ \oplus T_u(X))\ \oplus TS_1\ \oplus H(H(ID_i\ ||mk)\ \oplus r)\ \oplus T_u(X)$ $= RN_u$. Using computed parameters (ID_i', X', KA', M_2) and received TS_1, S calculates $M_3 = H(ID_i'||X'||KA'||M_2||TS_1)$. S then verifies whether $M_3 \stackrel{?}{=} M_2$. If the condition is not satisfied, this user request is rejected for this session. Otherwise, S accepts the login request and considers the user U_i as authentic.

- **Step 2.** S selects 128-bit random number RN_s, generates the current timestamp TS_2 and computes $M_4 = X' \oplus\ RN_s \oplus\ TS_2 = H(H(ID_i\ ||mk)\ \oplus r)\ \oplus RN_s\ \oplus TS_2$, $SK_{su} = H(X'||\ KA'||\ TS_1||\ TS_2||\ M_2||\ RN_s)$, and $M_s = H(ID_i||\ SK_{su}||\ M_2||\ RN_s||\ TS_1||\ TS_2)$, where SK_{su} is the common secret key shared with U_i for the current session. Finally, S sends the authentication request $\{M_4, M_s, TS_2\}$ to the user U_i via a public channel.

- **Step 3.** Upon receiving the server message at time ${TS_2}^*$, SC of U_i verifies the condition $|TS_2 - {TS_2}^*| \leq \triangle T$. If it holds, SC computes $M_5 = X' \oplus\ M_4 \oplus\ TS_2 = H(H(ID_i'\ ||mk)\ \oplus r)\ \oplus H(H(ID_i\ ||mk)\ \oplus r)\ \oplus RN_s\ \oplus TS_2\ \oplus TS_2 = RN_s$.

- **Step 4.** Using received timestamp TS_2, computed M_5 and $KA = T_u(T_{mk}(x))$, U_i calculates current session key shared with S as $SK_{us} = H(X||\ KA||\ TS_1||\ TS_2||\ RN_u||\ M_5)$, which is same as SK_{su}. Using this session key, U_i verifies $M_s \stackrel{?}{=} H(ID_i||\ SK_{us}\ ||RN_u\ ||M_5\ ||TS_1||TS_2)$. If verification succeeds, U_i assumes that S is an authenticate server. Also, the current session key SK_{us} $(= SK_{su})$ is mutually verified and established.

Figure 5.2 shows the summary of the login and authentication & key establishment phases of the proposed scheme.

5.3.4 Password change phase

In this phase, a valid user U_i can update his/her old password PW_i with a new password PW_i' using the following steps locally without further contacting the server S anymore:

- **Step 1.** U_i inserts the smart card SC and inputs his/her identity ID_i, old original password PW_i and a new changed password PW_i'. U_i also imprints his/her biometrics $\mathcal{B}_i$ at the sensor of a particular terminal.

- **Step 2.** SC generates $(\alpha_i, \beta_i) = Gen(\mathcal{B}_i)$, and calculates $b = D_2 \oplus H(H(ID_i||\ PW_i)||\ \alpha_i)$, $D_1' = D_1 \oplus H(H(ID_i\ ||PW_i)\ ||\alpha_i) \oplus H(H(ID_i\ ||PW_i')\ ||\alpha_i)$, $D_2' = H(H(ID_i\ ||PW_i')\ ||\alpha_i) \oplus b$ and $RPW_i' = H(H(ID_i\ ||PW_i)\ ||\alpha_i)$. Also, f_i is updated with $f_i' = H(RPW_i'\ ||b)$ in the memory of SC.

- **Step 3.** Finally, SC replaces D_1 with D_1', D_2 with D_2' and f_i with f_i' in its memory.

Figure 5.3 shows the summary of the password change phase of the proposed scheme.

5.3.5 Smartcard revocation phase

To revoke a lost/stolen smartcard, the proposed scheme performs the following steps:

- **Step 1.** U_i selects his/her identity ID_i, password PW_i and also imprints his/her biometrics $\mathcal{B}_i$ at the sensor of a particular terminal. U_i computes $(\alpha_i^*, \beta_i^*) = Gen(\mathcal{B}_i)$.

- **Step 2.** U_i selects a new 128-bit random number b' and submits $\langle ID_i, H(H(ID_i\ ||PW_i\ ||b')\ ||\alpha_i^*)\rangle$ to the server S via a secure channel.

User (U_i)	Server (S)
Input ID_i, PW_i and $\mathcal{B}_i'$.	
Compute $\alpha_i = Rep(\mathcal{B}_i', \beta)$,	
$b' = D_2 \oplus H(H(ID_i \| \| PW_i) \| \| \alpha_i)$.	
Verify if $f_i \stackrel{?}{=} H(RPW_i \| \| b')$.	
If verification holds, compute	
$X = D_1 \oplus H(H(ID_i \| \| PW_i \| \| b') \| \| \alpha_i)$.	Verify if $\|TS_1{}^* - TS_1\| \leq \triangle T$?
Generate u and RN_u.	Compute $KA' = T_{mk}(T_u(X))$,
Compute $T_u(X)$, $KA = T_u(T_{mk}(X))$,	$ID_i' = DID_i \oplus H(KA')$.
$M_1 = X \oplus RN_u \oplus TS_1$,	Check if $\langle ID_i', r \rangle$ is found in database.
$DID_i = ID_i \oplus H(KA)$,	Compute $X' = H(H(ID_i \| \| mk) \oplus r)$,
$M_u = H(ID_i \| \| X \| \| KA \| \| RN_u \| \| TS_1)$.	$M_2 = M_1 \oplus TS_1 \oplus X' = RN_u$.
$\xrightarrow[\textbf{(public channel)}]{\{DID_i, T_u(X), M_1, M_u, TS_1\}}$	Verify if $H(ID_i \| \| X' \| \| KA$
	$\| \| M_2 \| \| TS_1) \stackrel{?}{=} M_u$.
	If verification holds, generate RN_s.
	Compute $M_4 = X' \oplus RN_s \oplus TS_2$,
	$SK_{su} = H(X' \| \| KA' \| \| TS_1 \| \| TS_2 \| \| M_2 \| \| RN_s)$,
	$M_s = H(ID_i \| \| SK_{su} \| \| M_2 \| \| RN_s \| \| TS_1 \| \| TS_2)$.
	$\xleftarrow[\textbf{(public channel)}]{\{M_4, M_s, TS_2\}}$
Verify if $\|TS_2 - TS_2{}^* \leq \triangle T\|$?	
Compute $M_5 = X \oplus M_4 \oplus TS_2$,	
$= RN_s$, $SK_{us} = H(X \| \| KA \| \| TS_1$	
$\| \| TS_2 \| \| RN_u \| \| M_5)$.	
Verify if $M_s \stackrel{?}{=} H(ID_i \| \| SK_{us}$	
$\| \| RN_u \| \| M_5 \| \| TS_1 \| \| TS_2)$.	
If verification holds,	
authentication is successful.	
Store session key SK_{us} $(= SK_{su})$	Store session key SK_{su} $(= SK_{us})$
shared with S.	shared with U_i.

Figure 5.2: Login and authentication phases of the proposed scheme.

- **Step 3.** S uniquely identifies the user U_i by checking its credentials such as his/her SSN, DOB, national card number, or some other relevant information. Further, S reads the new serial number SN_i' from the new smart card, and selects a new 1024-bit random number r' and computes as following: $X' = H(H(ID_i \| \| mk) \oplus r')$, $D_1' = H(H(ID_i \| \| mk) \oplus r') \oplus H(H(ID_i \| \| PW_i \| \| b') \| \| \alpha_i^*) = X' \oplus H(H(ID_i \| \| PW_i \| \| b') \| \| \alpha_i^*)$. S embeds

User (U_i)	Smart card (SC)
Input ID_i, PW_i, and $\mathcal{B}_i'$.	
Input new password PW_i'.	
	Generate $(\alpha_i, \beta_i) = Gen(\mathcal{B}_i)$.
	Calculate
	$b = D_2 \oplus H(H(ID_i \|\| PW_i) \|\| \alpha_i)$,
	$D_1' = D_1 \oplus H(H(ID_i \|\| PW_i) \|\| \alpha_i)$
	$\oplus H(H(ID_i \|\| PW_i') \|\| \alpha_i)$,
	$D_2' = H(H(ID_i \|\| PW_i') \|\| \alpha_i) \oplus b$,
	$RPW_i' = H(H(ID_i \|\| PW_i) \|\| \alpha_i)$.
	Update $f_i \leftarrow f_i' = H(RPW_i' \|\| b)$.

Figure 5.3: Password change phase of the proposed scheme.

the parameters $\{D_1', T_{mk}(X')\}$ into the new smart card SC_{new} and issues this smart card to U_i through a secure channel and updates (ID_i, SN_i, r) with (ID_i, SN_i', r') in its database.

- **Step 4.** U_i receives the smart card SC_{new} from S and computes $RPW_i = H(H(ID_i \|\| PW_i) \|\| \alpha_i^*)$, $D_2' = RPW_i \oplus b'$ and $f_i' = H(RPW_i \|\| b')$. Finally, U_i stores β_i^*, D_2', f_i', $H(\cdot)$, $Gen(\cdot)$, $Rep(\cdot)$, and τ into smart card SC_{new}.

The smartcard revocation phase the of proposed scheme is summarized in Figure 5.4.

5.4 Security analysis

In this section, we provide both formal and informal security analysis of our scheme. Wang *et al.* [204] reviewed several anonymous two-factor authentication schemes and then pointed out that under the current widely accepted adversarial model, certain goals are beyond attainment. They further observed that the widely used formal methods including random oracle model and BAN logic can not capture some structural mistakes, and hence, guaranteeing the soundness of authentication protocols still remains an open issue. Due to such important observations in their analysis, it is necessary to have all the formal security analysis, BAN logic analysis, informal security analysis and formal security verification of the proposed scheme so that the scheme can achieve high level security.

User (U_i)	Smart card (SC)
Input ID_i, PW_i, and $\mathcal{B}_i'$.	
Compute $(\alpha_i^*, \beta_i^*) = Gen(\mathcal{B}_i)$.	
Select 128-bit random number b'.	
$\{ID_i, H(H(ID_i\|\|PW_i\|\|b')\|\|\alpha_i^*)\}$ $\longrightarrow$ **(secure channel)**	
	Identify U_i from SSN, DOB, ID card etc.
	Issue new smart card SC_{new}.
	Read new serial number SN_i' from SC_{new}.
	Select 1024-bit random number r'.
	Compute
	$X' = H(H(ID_i \|\|mk) \oplus r')$,
	$D_1' = H(H(ID_i \|\|mk) \oplus r') \oplus H(H(ID_i\|\| PW_i \|\|b')\|\| \alpha_i^*)$
	$= X' \oplus H(H(ID_i \|\|PW_i \|\|b') \|\|\alpha_i^*)$.
	Load $\{D_1', T_{mk}(X')\}$ into SC_{new}.
	$\{D_1', T_{mk}(X')\}$ $\longleftarrow$ **(Smart card)**
Compute	
$RPW_i = H(H(ID_i \|\|PW_i) \|\|\alpha_i^*)$,	
$D_2' = RPW_i \oplus b'$,	
$f_i' = H(RPW_i \|\|b')$.	
	Update $(ID_i, SN_i, r) \leftarrow (ID_i, SN_i', r')$
	in server database.

Figure 5.4: Lost smartcard revocation phase of the proposed scheme.

5.4.1 Formal security analysis using ROR model

We present the formal security analysis of the proposed biometrics and fuzzy extractor based user authentication protocol (BFE-UAP) through the Real-Or-Random (ROR) model [14], [214]. An adversary $\mathcal{A}$ can make several oracle queries, which model the adversary capabilities in a real attack [26], [185].

To proof the formal security of the proposed scheme, we consider all possible oracle queries. We simulate various security attacks on the proposed protocol $\mathscr{P}$ through the following dif-

ferent oracle queries:

- ***Send***(U_i/S,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_i,S): This query enables $\mathcal{A}$ with a capability to eavesdrop message m communicated between U_i and S in an actual execution of the protocol.
- ***Corrupt***(U_i,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 $\mathcal{A}$ sends a request to $\mathcal{P}^t$ for the current session key SK and receives a *null* value if no session key is generated. Otherwise, $\mathcal{P}^t$ takes 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 session key SK.

We now define the following definitions [26], [223] prior to proving Theorem 5.1.

Definition 5.1. *Upon receiving last expected protocol message, if $\mathcal{P}^t$ goes to an accept state, $\mathcal{P}^t$ is said to be in accepted state. The session identification (sid) is formed by the ordered concatenation of all communicated messages by $\mathcal{P}^t$.*

Definition 5.2. *Two instances ${U_i}^{TS_1}$ and S^{TS_2} are known to be partnered on simultaneous fulfillment of conditions between ${U_i}^{TS_1}$ and S^{TS_2} if 1) both are in accepted state, (2) both mutually authenticate each other and share the same sid and 3) they are mutual partners of each other.*

Definition 5.3 (Freshness)**.** *$\mathcal{P}^t$ is said to be fresh, on simultaneous accomplishment of 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 5.4 (Semantic security)**.** *The advantage function of an adversary $\mathcal{A}$ in breaking the semantic security of the proposed biometrics and fuzzy extractor based user authentication protocol (BFE-UAP) $\mathscr{P}$ by guessing the correct bit b' is defined by $Adv_{\mathscr{P}}^{BFE-UAP} = |2Pr[b = b'] - 1|$.*

Definition 5.5. *A password authentication protocol with biometrics is semantically secure if the advantage function* $Adv_{\mathscr{P}}^{BFE-UAP}$ *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,* $|\mathcal{D}|$ *is the size of password dictionary,* l_b *denotes the extracted string length of user biometrics and* ε_{bm} *is the probability of false positive [158].*

Definition 5.6. *The advantage probability* $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$ *of the Chaotic map-based discrete logarithm problem (CMDLP) is negligible for any adversary* $\mathcal{A}$ *with execution time* $t_{\mathcal{A}}$*, that is,* $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}) \leq \epsilon$*, for a sufficiently small* $\epsilon > 0$*.*

Theorem 5.1. *Let* $\mathcal{A}$ *be a polynomial time bounded adversary running in time* $t_{\mathcal{A}}$*. Suppose* $\mathcal{A}$ *make* 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 scheme* $\mathscr{P}$*. Then,*

$$\begin{aligned} Adv_{\mathscr{P}}^{BFE-UAP} \leq\ & \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s+q_e)^2 + 4q_s}{2^{l_r}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} \\ & +4q_H(1+(q_s+q_e)^2)Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}), \end{aligned}$$

where l_H *refers to the string length of hash results,* l_r *is the string length of random numbers,* l_b*,* ε_{bm} *and* $|\mathcal{D}|$ *are defined in Definition 5.5, and* $Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$ *is the advantage probability of breaking* $CMDLP$ *problem by* $\mathcal{A}$ *defined in Definition 5.6.*

Proof. A set of six games are defined as G_i, $(i = 0, 1, 2, 3, 4, 5)$. Let S_i refer to an event of successfully guessing bit b in $Test$ query by an adversary $\mathcal{A}$ in the game G_i. The detailed descriptions of the games are given below.

- **Game** G_0**:** Assuming the real protocol in random oracles and the initial game are identical, we obtain,

$$Adv_{\mathscr{P}}^{BFE-UAP} = |2Pr[S_0] - 1|. \tag{5.1}$$

- **Game** G_1**:** Oracle queries like $Reveal$, $Execute$, $Corrupt$, $Test$ and H are simulated in the game G_1 and described in Table 5.2. The working procedure of $Send$ query is simulated in Table 5.3. We create three lists that record the output 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_i and S. As simulation of the games G_1 and G_0 (which is the real protocol under execution) are considered to be indistinguishable, we have,

$$Pr[S_1] = Pr[S_0]. \tag{5.2}$$

Table 5.2: Simulation of hash, reveal, test, corrupt and execute oracle queries.

Hash H simulation query performs as follows: If the record (q, H) is found in list L_H corresponding to hash query $H(q)$, return hash function 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_i,a) simulation query performs as follows: If $a = 1$, the query returns password (PW_i) of the user U_i. If $a = 2$, the query outputs biometrics key (α_i) corresponding to the biometrics $\mathcal{B}_\rangle$ of U_i. If $a = 3$, the query returns the secret information stored in user smart card.
Simulation of *Execute*(U_i, S) query occurs in succession with simulation of *Send* queries as shown below. Compute DID_i, M_1, M_u as given in Figure 5.2. U_i sends message M' to S, where $M' = \{DID_i, T_u(X), M_1, M_u, TS_1\}$. Compute M_4 and M_s as given in Figure 5.2. S sends authentication message M'' to U, where $M'' = \{M_4, M_s, TS_2\}$. Note that $\langle DID_i, T_u(X), M_1, M_u, TS_1\rangle \leftarrow Send(U, \textbf{start})$, $\langle M_4, M_s, TS_2\rangle \leftarrow Send(S, \langle DID_i, T_u(X), M_1, M_u, TS_1\rangle)$. Finally, $M' = \langle DID_i, T_u(X), M_1, M_u, TS_1\rangle$, $M'' = \langle M_4, M_s, TS_2\rangle$ are returned.

- **Game G_2:** This game considers the collision situations with hash results and random numbers in the transcripts of all communicated messages in login and authentication phases of our scheme $\mathscr{P}$. According to the birthday paradox, the H query has at most collision probability as $\frac{q_H^2}{2^{l_H+1}}$. In the login and authentication messages

$\{DID_i, T_u(X), M_1, M_u, TS_1\}$ and $\{M_4, M_s, TS_2\}$, M_1 and M_2 contain session specific random numbers RN_u and RN_s, respectively. Hence, the probability of collision for these numbers are at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. So, we have,

$$|Pr[S_2] - Pr[S_1]| \leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_H^2}{2^{l_H+1}}. \tag{5.3}$$

Table 5.3: Simulation of send oracle queries.

Send simulation query performs as follows. **(a)** For a $Send(U_i, \textbf{start})$ query, U_i gives the following response. Compute X, M_1, DID_i, M_u as in Figure 5.2. Output $M' = \langle DID_i, T_u(X), M_1, M_u, TS_1 \rangle$.
(b) Let S be the target state. For a $Send(S, \langle DID_i, T_u(X), M_1, M_u, TS_1 \rangle)$ query, S gives the following response. Verify whether $\lvert T_1 - T_1^* \rvert \leq \triangle T$ and compute KA', ID_i', X' and M_2 and verifies value of M_u. A mismatch rejects the session. Further, S computes M_4 and SK_{su}, and output $M'' = \langle M_4, M_s, TS_2 \rangle$.
(c) U_i answers $Send(U_i, \langle M_4, M_s, TS_2 \rangle)$ query as follows. Verify whether $\lvert TS_2 - T_2^* \rvert \leq \triangle T$, and compute M_5 and SK_{us}, and then verify M_s. A mismatch leads to termination of the session. Otherwise, establish SK_{us} as the session key as given in Figure 5.2. Finally, both U_i and S accept the successful termination of the session.

- **Game G_3:** In this game, without involving hash oracles H directly, $\mathcal{A}$ tries to guess the correct message from other oracle queries. As per two message communications of login and authentication phases, we have the following two cases:
  - **Case 1:** We consider login message $Send(S, M_1)$ query and try to respond it. Hence, the hash value $M_u = H(ID_i||X||KA||RN_u||TS_1) \in L_{\mathcal{A}}$ has probability up to $\frac{q_H}{2^{l_H}}$; otherwise, the session will be terminated. Again, to launch an attack, $\mathcal{A}$

must find out the parameters b', f_i, and $H(H(ID_i||PW_i||b')||\alpha_i)$ as given in Figure 5.1. Hence, the total calculated probability is at most $\frac{4q_H}{2^{l_H}}$. Finally, for a transcript message with parameter RN_u, $M_1 \in L_T$, and we get the maximum probability for this as $\frac{q_s}{2^{l_r}}$.

– **Case 2:** We consider M_2 which is included in the authentication message sent by S to the user U_i. To respond $Send(U_i, M_2)$ oracle query, $M_s \in L_{\mathcal{A}}$ must hold with probability $\frac{q_H}{2^{l_H}}$. Further, S computes and verifies ID_i', X', $H(ID_i||X||KA||M_2||TS_1)$ and SK_{su} with total probability of $\frac{5q_H}{2^{l_H}}$. Finally, for a transcript message (with parameter RN_s) $M_2 \in L_T$ and we get the maximum probability for this as $\frac{q_s}{2^{l_r}}$.

Considering both cases, we have,

$$|Pr[S_3] - Pr[S_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}}. \tag{5.4}$$

- **Game G_4:** In G_4, we consider mainly guessing attacks in both online and offline conditions. We have the following two cases:

  – **Case 1:** $\mathcal{A}$ executes $Corrupt(U_i, 3)$ to guess PW and α_i. In this case, we consider the following two sub-cases:

    * **Case 1.1:** $\mathcal{A}$ can guess password in online from a dictionary $\mathcal{D}$ and runs $Send(S, M_1)$ query q_s times having probability $\frac{q_s}{|\mathcal{D}|}$.
    * **Case 1.2:** We consider the intentional or accidental guessing of user biometrics key α_i online, which requires to execute query $Corrupt(U_i, 2)$. The guessing probability under this case is at most $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$.

  – **Case 2:** We consider guessing of the session key SK by $\mathcal{A}$ without active involvement of oracle H. We find that SK is created with hash values of two chaotic map parameters $T_u(T_{mk}(X))$ and $T_{mk}(T_u(X))$. So, for this case, the probability is at most $2q_H Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}})$.

  In absence of these guessing attacks, the games G_4 and G_3 are indistinguishable, and hence, we have,

$$|Pr[S_4] - Pr[S_3]| \leq \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} + 2q_H Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}). \tag{5.5}$$

- **Game G_5:** In the terminating game G_5, $\mathcal{A}$ executes H, *Send* and *Execute* oracle queries on old transcripts only to break forward security. To avoid termination of the game, the *Test* query should return the actual session key in respective instances of U_i and S. Considering the same analysis as mentioned in G_4, we obtain,

$$|Pr[S_5] - Pr[S_4]| \leq 2q_H(q_s + q_e)^2 \times Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}).$$

  Considering all above games and since $\mathcal{A}$ gains no advantage to guess the correct bit b, we get, $Pr[S_5] = 1/2$.

Using the triangular inequality, we have the following:

$$\begin{aligned} |Pr[S_0] - \frac{1}{2}| &= |Pr[S_1] - Pr[S_5]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_5]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| + |Pr[S_3] - Pr[S_5]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| + |Pr[S_3] - Pr[S_4]| \\ &\quad + |Pr[S_4] - Pr[S_5]|. \end{aligned} \tag{5.6}$$

Using Equations (5.1)-(5.6), we obtain,

$$\begin{aligned} \frac{1}{2}Adv_{\mathscr{P}}^{BFE-UAP} &= |Pr[S_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_H^2}{2^{l_H+1}} + \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}} + \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} \\ &\quad + 2q_H Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}) + 2q_H(q_s + q_e)^2 Adv_{\mathcal{A}}^{CMDLP}(t_{\mathcal{A}}). \end{aligned} \tag{5.7}$$

Finally, multiplying both sides by 2 in Equation (5.7) and rearranging the terms, we obtain the required result. Hence, the theorem is proved. □

5.4.2 Mutual authentication proof using BAN logic

Basic BAN logic notations and logical postulates are provided in Section 2.6. According to the analytic procedures of the BAN logic, the proposed protocol needs to satisfy the following goals:

- **Goal 1**. $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- **Goal 2**. $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.

The generic types of the messages in the proposed protocol are as follows.

- **Message 1.** $U_i \rightarrow S_j$: $\{DID_i, T_u(X), X \oplus RN_u \oplus TS_1 \oplus T_u(X), H(ID_i \,||X\, ||KA\, ||RN_u\, ||TS_1), TS_1\}$.
- **Message 2.** $S_j \rightarrow U_i$: $\{X \oplus RN_s \oplus TS_2 \oplus T_{mk}(X), H(ID_i|| \; H(X \; ||T_{mk}(T_u(X)) \; ||TS_1 \; ||TS_2 \; ||M_2 \; ||RN_s) \; ||M_2 \; ||RN_s \; ||TS_1 \; ||TS_2), TS_2\}$.

The idealized forms of the messages in the proposed protocol are given below.

- **Message 1.** $U_i \rightarrow S_j$: $\{DID_i,\ T_u(X),\ TS_1,\ \langle RN_u,\ TS_1,\ T_u(X)\rangle_X,\ \langle ID_i,\ RN_u,\ TS_1,\ KA\rangle_X\}$.
- **Message 2.** $S_j \rightarrow A$: $\{TS_2, \langle RN_s, TS_2, T_{mk}(X)\rangle_X, \langle ID_i, TS_1, TS_2, KA, RN_u, RN_s\rangle_X, TS_2\}$.

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

- **A.1:** $U_i \mid\equiv \#(TS_2)$
- **A.2:** $S_j \mid\equiv \#(TS_1)$
- **A.3:** $U_i \mid\equiv (U_i \overset{X}{\rightleftharpoons} S_j)$
- **A.4:** $S_j \mid\equiv (U_i \overset{X}{\rightleftharpoons} S_j)$
- **A.5:** $U_i \mid\equiv S_j \Rightarrow (RN_s, TS_2, T_{mk}(X))$
- **A.6:** $S_j \mid\equiv U_i \Rightarrow (RN_u, TS_1, T_u(X))$
- **A.7:** $U_i \mid\equiv u$
- **A.8:** $U_i \mid\equiv TS_1$
- **A.9:** $U_i \mid\equiv RN_u$
- **A.10:** $S_j \mid\equiv T_{mk}(X)$
- **A.11:** $S_j \mid\equiv mk$
- **A.12:** $S_j \mid\equiv TS_2$

- **A.13:** $S_j \mid\equiv RN_s$

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

According to the message 1, we obtain,

- S_1: $S_j \lhd \{DID_i, T_u(X), TS_1, \langle RN_u, TS_1, T_u(X)\rangle_X, \langle ID_i, RN_u, TS_1, KA\rangle_X\}$.
- S_2: According to the rule AR, we obtain, $S_j \lhd \langle RN_u, TS_1, T_u(X)\rangle_X$.
- S_3: According to A.4 and MMR, we obtain, $S_j \mid\equiv U_i \mid\sim (RN_u, TS_1, T_u(X))$.
- S_4: According to A.2 and FCR, we get, $S_j \mid\equiv \#(RN_u, TS_1, T_u(X))$.
- S_5: According to NVR, we have, $S_j \mid\equiv U_i \mid\equiv (RN_u, TS_1, T_u(X))$.
- S_6: Using A.6 and JR, we get, $S_j \mid\equiv (RN_u, TS_1, T_u(X))$.
- S_7: From S_6 and AR, we obtain, $S_j \mid\equiv RN_u$, $S_j \mid\equiv TS_1$, $S_j \mid\equiv T_u(X)$.
- S_8: According to A.11, A.12 and A.13, we get, $S_j \mid\equiv mk$, $S_j \mid\equiv TS_2$ and $S_j \mid\equiv RN_s$.
- S_9: From $SK = H(X \mid\mid T_{mk}(T_u(X)) \mid\mid TS_1 \mid\mid TS_2 \mid\mid M_2 \mid\mid RN_s)$ and the results obtained in Steps S_7 and S_8, we obtain, $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$. **(Goal 2)**
- S_{10}: Using the message 2 and AR, we obtain, $U_i \lhd \langle RN_s, TS_2\rangle_X$.
- S_{11}: According to A.3 and MMR, we get, $U_i \mid\equiv S_j \mid\sim (RN_s, TS_2)$.
- S_{12}: Using A.1 and FCR, we obtain, $U_i \mid\equiv \#(RN_s, TS_2)$.
- S_{13}: With NVR, we obtain, $U_i \mid\equiv S_j \mid\equiv (RN_s, TS_2)$.
- S_{14}: A.5 and JR give $U_i \mid\equiv (RN_s, TS_2)$.
- S_{15}: According to S_{14} and AR, we have, $U_i \mid\equiv RN_s$, $U_i \mid\equiv TS_2$.
- S_{16}: According to A.7-A.10, we obtain, $U_i \mid\equiv u$, $U_i \mid\equiv TS_1$, $U_i \mid\equiv RN_u$ and $S_j \mid\equiv T_{mk}(X)$.
- S_{17}: The results of Steps S_{15} and S_{16} give $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$. **(Goal 1)**

As a result, the goals **Goal 1** and **Goal 2** ensure that both U_i and S mutually authenticate each other.

5.4.3 Discussion on other attacks

In this section, we show that the proposed can also withstand the following security attacks.

1) Replay and impersonation attacks

Replay attack is considered to be one of the most common attacks in authentication process. In the proposed scheme, an attacker cannot replay the login message $\{DID_i, T_u(X), M_1, M_u, TS_1\}$. S ignores the message if $|TS_1{}^* - TS_1| > \triangle T$, where $\triangle T$ is the maximum transmission delay. To protect strong replay attack, S may also decide to store the pair $(ID_i, T_u(X))$. In case of a replayed message, $T_u(X)'$ will be same as the previous value $T_u(X)$. So, S considers this as a replayed message and discards the request. Further, an attacker can not modify any of the sent parameters as the message contains a hash value $M_u = H(ID_i||X||KA||RN_u||TS_1)$. In the authentication phase, any modification of the previous parameters leads to a mismatch of the sent parameters and received hash value, and the authentication request fails. Following the same logic explained above, an attacker can not also replay or modify the server authentication message. Hence, our scheme is secure against replay attack as well as user impersonation attack.

2) Man-in-the-middle attack

Through man-in-the-middle attack, an adversary may try to modify login or authentication messages. Also, the attacker may try to establish independent connections with U_i and S. However, as we already discussed in Section 5.4.3, an adversary will not be able to modify or regenerate login and authentication messages. Thus, the proposed scheme resists man-in-the middle attack.

3) Stolen smart card attack

Suppose the user U_i's smart card SC is lost or stolen, and an adversary $\mathcal{A}$ obtains all stored parameters from its memory by power analysis attacks [142]. Note that U_i's identity ID_i, password PW_i and biometric secret key α_i are not stored directly in SC. From stored $D_1 = H(H(ID_i||\ mk)\ \oplus r)\ \oplus H(H(ID_i\ ||PW_i\ ||b)\ ||\alpha_i)$, it is computationally infeasible to obtain ID_i, PW_i or biometrics α_i as $\mathcal{A}$ has to guess all these parameters simultaneously. Moreover, from $D_2 = RPW_i \oplus b$, $\mathcal{A}$ cannot obtain ID_i, PW_i or α_i. Also, RPW_i is masked with random number b. In the same logic, $\mathcal{A}$ can not obtain ID_i, PW_i or α_i from stored $f_i = H(RPW_i\ ||b)$. As a result, our scheme prevents stolen smart card attack (smart card breach attack).

4) Offline password guessing attack

Suppose an attacker $\mathcal{A}$ extracts all the stored information from the memory of a lost or stolen smart card of a legal user U_i by the power analysis attacks. To obtain U_i's identity ID_i, password PW_i or biometric key α_i, $\mathcal{A}$ needs to guess them all simultaneously from stored D_2 and f_i. The collision resistant property of the one-way hash function $H(\cdot)$ ensures that our scheme resists offline password guessing attack.

5) Known key secrecy/forward secrecy

In the proposed scheme, even if a particular session key is compromised, it does not help an adversary to reveal the other session keys. According to our scheme, the session key is computed as follows: $SK_{su} = H(ID_i|| X|| T_{mk}(T_u(x)) ||TS_1 ||TS_2 ||M_2 ||RN_s) = H(ID_i ||X ||T_{mk.u}(x) ||TS_1 ||TS_2 ||RN_u ||RN_s) = H(ID_i ||X ||T_{u.mk}(x) ||TS_1 ||TS_2 ||RN_u ||M_5) = H(ID_i ||X ||T_u(T_{mk}(x)) ||TS_1 ||TS_2 ||RN_u ||M_5) = SK_{us}$, where $KA = T_{mk}(T_u(X)) = T_u(T_{mk}(X))$, and u, RN_u, TS_1, RN_s, and TS_2 are generated randomly and uniquely for every new login session. So, this session key is fresh and unique for every session, and it can not be reused after the expiration of session. Thus, an adversary can not obtain any secret information from a compromised session key in order to compute the future session keys. In addition, before establishing the session key, both U_i and S mutually authenticate each other. Hence, the established session key and all communicated messages encrypted through this session key is secure against different attacks. Furthermore, the construction of the session key is based on both the temporal secrets, such as u, RN_u, TS_1, RN_s, and TS_2 as well as permanent secrets, such as mk and ID_i. The leakage of the temporal secrets do not lead to compromise the secrecy of the session key. As a result, the proposed scheme also provides the session key security.

6) User anonymity

Our scheme provides user anonymity property as an adversary $\mathcal{A}$ can not obtain user identity ID_i from any eavesdropped login or authentication message. Suppose $\mathcal{A}$ intercepts the login message $\{DID_i, T_u(X), M_1, M_u, TS_1\}$ during the login phase, where $DID_i = ID_i \oplus H(KA)$, $M_1 = X \oplus RN_u \oplus TS_1 = D_1 \oplus H(H(ID_i ||PW_i ||b') ||\alpha_i) \oplus RN_u \oplus TS_1$ and $M_u = H(ID_i ||X ||KA ||RN_u ||TS_1)$. Due to usage of random numbers r, RN_u and TS_1, and collision-resistant hash function $H(\cdot)$, it is computational infeasible task for $\mathcal{A}$ to derive ID_i from the eavesdropped login message. In a similar way, $\mathcal{A}$ can not also obtain ID_i from the intercepted

authentication message $\{M_4, M_s, TS_2\}$. Therefore, the proposed scheme achieves the user anonymity property.

```
(*————————channels————————*)
free pch: channel. (*public channel*)
free sch: channel [private]. (*private channel*)
(*————————shared keys————————*)
free SKu:bitstring [private].(*the session key of user*)
free SKs:bitstring [private]. (*the session key of server*)
(*————————Servers secret key————————*)
free mk:bitstring [private].
free r:bitstring [private].
(*————————constants————————*)
free ID:bitstring [private].
free PW:bitstring [private].
const Bi:bitstring [private].
const A:bitstring [private].
(*————————functions and equations————————*)
fun H(bitstring):bitstring. (*hash function*)
fun BH(bitstring):bitstring. (*Biohash function*)
fun xor(bitstring,bitstring):bitstring. (*XOR operation*)
fun con(bitstring,bitstring):bitstring. (*string concatenation*)
fun cmap(bitstring,bitstring):bitstring. (*Chaotic map operation*)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
equation forall u:bitstring,v:bitstring; cmap(u,cmap(v,A)) = cmap(v,cmap(u,A)).
(*———————— aims for verification————————*)
query attacker(SKu).
query attacker(SKs).
query id:bitstring; inj-event(UserAuth(id))==> inj-event(UserStart(id)).
(*————————event————————*)
event UserStart(bitstring). (*User starts authentication*)
event UserAuth(bitstring). (*User is authenticated*)
```

Figure 5.5: Declaration of channels, keys, constants, functions, equations, queries and events.

7) Parallel session and reflection attacks

In our scheme, from any one of the eavesdropped messages $\{DID_i, T_u(X), M_1, M_u, TS_1\}$ and $\{M_4, M_s, TS_2\}$, an attacker can neither obtain the correct identity ID_i, password PW_i nor the biometrics key α_i of a legal user U_i. Hence, from any eavesdropped message, an attacker can not a create a valid login request, and thus, he/she cannot start a new session with S by masquerading as a legal user. Hence, our scheme protects the parallel session and reflection attacks.

5.5 Formal security verification using ProVerif simulation tool

In this section, we present the formal security verification of the proposed scheme using the popular verification tool, called ProVerif [12]. This tool is based on applied pi calculus [13] and is used for proving session key secrecy and authentication. The details of implementation can be found in [5].

In Figure 5.5, we provide the code for declaration of channels, free variables, constants, functions, equations, queries and events required for the proof. The code for process of user in both registration phase and authentication phase is modeled in Figure 5.6.

The process of sever S can be modeled as parallel composition of process of registration (SReg) and process of authentication (SAuth). Figure 5.7 shows the program code for the processes of S.

Finally, we execute the codes of previous three tables in ProVerif latest version, i.e., ProVerif 1.93. The complete obtained result/output of session key secrecy (from both user and server side) and authentication is tabulated in Figure 5.8. This output can be verified via the official reference "http://proverif.rocq.inria.fr/index.php". The result shows that:

- RESULT inj-event(UserAuth(id)) ==> inj-event (UserStart(id)) is true.
- RESULT not attacker(SKs[ ]) is true.
- RESULT not attacker(SKu[ ]) is true.

In summary, the proposed scheme passes the ProVerif 1.93 security verification.

```
let User=
new b:bitstring;
let bi = BH(Bi) in
let RPW = H(con(H(con(ID,PW)),bi)) in
let Msg = H(con(H(con(ID,con(PW,b))),bi)) in
out(sch,(ID,Msg));
in(sch,(rD1:bitstring,rP:bitstring));
let D2 = xor(RPW,b) in
let fi = H(con(RPW,b)) in
```

```
!
(
event UserStart(ID);
let b1 = xor(D2,H(con(H(con(ID,PW)),bi))) in
let RPW1 = H(con(H(con(ID,PW)),bi)) in
let fi1 = H(con(RPW1,b1)) in
if fi = fi1 then
let X1 = xor(rD1,H(con(H(con(ID,con(PW,b1))),bi))) in
new u:bitstring;
new RNu:bitstring;
new T1:bitstring;
let Q = cmap(u,X1) in
let KA = cmap(u,rP) in
let M1 = xor(X1,xor(RNu,T1)) in
let DID = xor(ID,H(KA)) in
let Mu = H(con(ID,con(X1,con(KA,con(RNu,T1))))) in
out(pch,(DID,Q,M1,Mu,T1));
in(pch,(rM4:bitstring,rMs:bitstring,rT2:bitstring));
let M5 = xor(X1,xor(rM4,rT2)) in
let SKu = H(con(X1,con(KA,con(T1,con(rT2,con(RNu,M5)))))) in
if rMs = H(con(ID,con(SKu,con(RNu,con(M5,con(T1,rT2)))))) then
0
).
```

Figure 5.6: ProVerif code for the process of user.

```
let SReg =
in(sch,(sID:bitstring,sMsg:bitstring));
let X = H(xor(H(con(sID,mk)),r)) in
let P = cmap(mk,X) in
let D1 = xor(X,sMsg) in
out(sch,(D1,P)).
```

```
let SAuth =
in(pch,(xDID:bitstring,xQ:bitstring,xM1:bitstring,xMu:bitstring,xT1:bitstring));
let KA1 = cmap(mk,xQ) in
let ID1 = xor(xDID,H(KA1)) in
let X2 = H(xor(H(con(ID1,mk)),r)) in
let M2 = xor(xM1,xor(xT1,X2)) in
let Mu1 = H(con(ID1,con(X2,con(KA1,con(M2,xT1))))) in
if xMu = Mu1 then
event UserAuth(ID1);
new RNs:bitstring;
new T2:bitstring;
let M4 = xor(X2,xor(RNs,T2)) in
let SKs = H(con(X2,con(KA1,con(xT1,con(M2,xT1))))) in
let Ms = H(con(ID1,con(SKs,con(M2,con(RNs,con(xT1,T2)))))) in
out(pch,(M4,Ms,T2)).
```

```
let S = SReg | SAuth.
```

```
process !User | !S
```

Figure 5.7: ProVerif code for the process of server.

5.6 Performance comparison

In this section, we compare the performance of the proposed scheme with the recent authentication schemes for e-health care systems, such as the schemes of Lee [126], Li *et al.* [127], Xie *et al.* [222] and Xu *et al.* [225].

```
File "./tmpfiles/29940818/inpProt.pv", line 60, character 5 - line 60, character 8:
Warning: identifier SKu rebound
File "./tmpfiles/29940818/inpProt.pv", line 83, character 5 - line 83, character 8:
Warning: identifier SKs rebound
Completing equations...
Completing equations...
– Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
nounif mess(sch[],(sID_488,sMsg_489))/-5000
Completing...
200 rules inserted. The rule base contains 200 rules. 64 rules in the queue.
400 rules inserted. The rule base contains 384 rules. 60 rules in the queue.
600 rules inserted. The rule base contains 515 rules. 54 rules in the queue.
800 rules inserted. The rule base contains 614 rules. 50 rules in the queue.
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.
– Query not attacker(SKs[ ])
nounif mess(sch[],(sID_10359,sMsg_10360))/-5000
Completing...
200 rules inserted. The rule base contains 200 rules. 48 rules in the queue.
400 rules inserted. The rule base contains 352 rules. 48 rules in the queue.
600 rules inserted. The rule base contains 489 rules. 48 rules in the queue.
Starting query not attacker(SKs[ ])
RESULT not attacker(SKs[]) is true.
– Query not attacker(SKu[ ])
nounif mess(sch[],(sID_19433,sMsg_19434))/-5000
Completing...
200 rules inserted. The rule base contains 200 rules. 48 rules in the queue.
400 rules inserted. The rule base contains 352 rules. 48 rules in the queue.
600 rules inserted. The rule base contains 489 rules. 48 rules in the queue.
Starting query not attacker(SKu[ ])
RESULT not attacker(SKu[]) is true.
```

Figure 5.8: Analysis of results.

5.6.1 Communication cost analysis

From the tabulated data in Table 5.4, it is clear that the proposed scheme has low communication cost as compared to other related authentication schemes for e-health care systems during the the login and authentication phases. Since the registration process is one-time, we have not considered the costs involved in the schemes. Bit transmission overhead of our scheme is quite moderate. Our scheme requires only two message exchanges of sizes 640 bits and 352 bits only. The communication cost of Xu *et al.*'s scheme [225] is slightly less than that for the proposed scheme. However, their scheme suffers from several security attacks and functionality features (see Table 5.7).

Table 5.4: Comparison of communication costs.

Scheme	No. of messages	No. of bits
Lee [126]	2	1280
Xu *et al.* [225]	2	864
Xie *et al.* [222]	3	1096
Li *et al.* [127]	3	1472
Our	2	992

Table 5.5: Notations used and their time complexity.

Symbol	Description	Execution time (in milliseconds)
T_H	One-way hash function	0.50
T_{sym}	symmetric key encryption/decryption	8.70
T_M	Elliptic curve point multiplication	63.08
T_{CH}	Chebyshev map operation	21.02
T_{FE}	Fuzzy extractor operation	$\approx T_M$

5.6.2 Computation cost analysis

Table 5.5 shows different notations and their execution time executed on an Intel Pentium4 2600 MHz processor with 1024 MB RAM as performed in [118], [177]. In Table 5.6, we

Table 5.6: Comparison of computation costs.

Scheme	User side	Server side	Execution cost (user)	Execution cost (server)
Lee [126]	$9T_H + 2T_{CH}$	$10T_H + 2T_{CH}$	46.54 ms	47.04 ms
Xu *et al.* [225]	$5T_H + 3T_M$	$5T_H + 3T_M$	191.74 ms	191.74 ms
Xie *et al.* [222]	$7T_H + 2T_M + T_{sym} + 2T_{sym}$	$6T_H + 2T_M$	138.36 ms	146.56 ms
Li *et al.* [127]	$8T_H + 2T_{CH}$	$9T_H + 2T_{CH}$	46.04 ms	46.54 ms
Our	$9T_H + 1T_{FE} + 2T_{CH}$	$5T_H + 1T_{CH}$	109.62 ms	23.52 ms

tabulate and compare the computation overhead of our scheme with the relevant chaotic map based schemes for e-health care systems [126], [127], [222], [225] during the login and authentication phases. For all the given schemes, we separately tabulated the user side and server side computation costs. Since the bitwise XOR operation is negligible, we have ignored it. During the login phase, the computation overhead required for a user in our scheme is $9T_H + 2T_{CH} + T_{FE}$, whereas during the authentication phase, the computation overhead becomes $5T_H + T_{CH}$. Thus, the execution time for user and server are 109.62 ms and 23.52 ms, respectively. We have applied the fuzzy extractor functions for biometric verification. Due to the fuzzy extractor $Rep(\cdot)$ function for extracting the biometric key α_i, we require $T_{FE} \approx T_M$ [87]. However, the cost for the server side in the proposed scheme is relatively less as compared to all other scheme. In addition, though the computation cost for the user side in the proposed scheme is higher than that for the schemes of Lee [126] and Li *et al.* [127], it is justified because their schemes suffer from several security attacks and functionality features (see Table 5.7).

5.6.3 Security and functionality analysis

A detailed comparison on different security attacks and functionality features are tabulated in Table 5.7. Most of the schemes shown in Table 5.7 fail to provide efficiency in login phase and password change phase, and also they do not provide revocation of lost smart card phase. It is clear from Table 5.7 that the proposed scheme overcomes such security and functionality weaknesses of the existing schemes.

Table 5.7: Security and functionality comparison.

Security attribute /Scheme	Lee [126]	Xu *et al.* [225]	Xie [222]	Li *et al.* [127]	Our
Stolen smart card attack	X	✓	✓	✓	✓
Off-line password guessing attack	✓	✓	✓	✓	✓
On-line password guessing attack	✓	✓	✓	✓	✓
Strong replay attack	✓	X	✓	✓	✓
Privileged insider attack	X	✓	✓	✓	✓
User impersonation attack	✓	✓	✓	✓	✓
Server impersonation attack	✓	✓	✓	✓	✓
Denial of service attack	X	X	✓	X	✓
Known session key secrecy	✓	✓	✓	✓	✓
User anonymity provision	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓	✓
Session key security	X	✓	✓	✓	✓
Efficient password change	X	X	✓	X	✓
Login phase efficiency	X	X	✓	X	✓
Mutual authentication	✓	✓	✓	✓	✓
Revocation of smart card	X	X	X	X	✓
Low computation overhead	✓	X	X	✓	✓
Low communication overhead	X	✓	X	X	✓
Formal security analysis	X	X	X	X	✓

✓: the scheme is secure or it supports the feature; X: the scheme is not secure or it does not support the feature.

5.7 Summary

We proposed a secure, lightweight and efficient chaotic-map based user authentication scheme for e-healthcare systems. Most of the existing chaotic map based schemes are insecure against several known attacks. The proposed scheme can effectively withstand the attacks outlined in the existing schemes. Through the formal security analysis using the widely-accepted ROR model, it was shown that our scheme provides the session key security. Furthermore, the

BAN logic analysis shows that the proposed scheme provides the secure mutual authentication between a user and the medical server. In addition, the formal security verification using the ProVerif tool ensures that the proposed scheme is also secure. Finally, through extensive performance comparison, it was also shown that the proposed scheme considerably reduces total computation and communication costs as compared to other existing related schemes. As a result, the proposed scheme is very suitable for e-healthcare systems.

Chapter 6

Biometric-Based Anonymous User Authentication for Mobile Cloud Computing Services

Mobile Cloud Computing (MCC) provides cloud resources through on-demand basis by integrating cloud computing into mobile environment. Nowadays, both in industry as well as academia, mobile cloud computing has drawn much attention. Before providing any access of cloud service to a mobile user, mutual authentication of a mobile user and the cloud service provider is necessary. Authentication scheme should be lightweight with respect to resource constrained user mobile device. To access a mobile cloud computing service, a mobile user, say MU_i requests the cloud service through an installed mobile App or web browser. After that, a mutual authentication between MU_i and the cloud service provider CS_j is done by the user mobile App or web browser. Both MU_i and CS_j need to go through a secure mutual authentication process that should support some basic requirements, such as computation efficiency, user anonymity, and session key security. Intrinsically, mobile cloud computing services are quite distributed and heterogeneous in nature. Thus, registering separately for each cloud service provider by maintaining respective user account is a difficult task. To be precise, a mobile user requires to access several cloud services from cloud servers with the help of single registered user account.

In this chapter, we propose a new secure and lightweight mobile user authentication scheme for mobile cloud computing based on cryptographic hash, bitwise XOR and fuzzy extractor functions. The proposed scheme provides mobile user authentication in distributed mobile cloud computing environment, which supports secure key exchange, and user anonymity and

untraceability properties.

6.1 Research contributions

The following contributions are made in this chapter:

- The proposed scheme provides mobile user authentication in distributed mobile cloud computing environment, which supports secure key exchange, and user anonymity and untraceability properties.
- Compared with the related existing authentication schemes proposed in the mobile cloud computing environment, the proposed scheme has the lowest computation and storage requirements. This is primarily due to usage of efficient one way cryptographic hash function, bitwise XOR operation and fuzzy extractor operation only. It is worth noting that the fuzzy extractor method is applied for biometric verification.
- No trusted third party, like Identity Provider (IdP), Smart Card Generator (SCG) or Registration Center (RC) is involved in user login and authentication phases. This reduces overall communication and computation time of the proposed scheme.
- The proposed scheme is lightweight in nature, and meanwhile, it also removes the security and functionality drawbacks of the earlier existing schemes.
- The proposed scheme has the ability to resist various known attacks. These are evident through the rigorous formal security analysis using the widely-accepted ROR model and BAN logic. Further, we provide simulation of security security verification using the broadly-accepted ProVerif 1.93 simulation tool.

6.2 Threat model

In this chapter, we follow the widely-accepted Dolev-Yao threat model (DY model) [70], which accepts the following basic assumptions:

- The login and authentication messages between mobile users and cloud servers are communicated over a public insecure channel.
- The pubic channel messages are susceptible to eavesdropping, deletion or modification, which are executed by an adversary $\mathcal{A}$.

- If, by any means, the adversary $\mathcal{A}$ obtains legal user's smart card or mobile device, he/she can execute power analysis attack and also extract all stored information from the device [119], [142].

More details on Dolev-Yao threat model and power analysis attack are provided in Section 4.2.

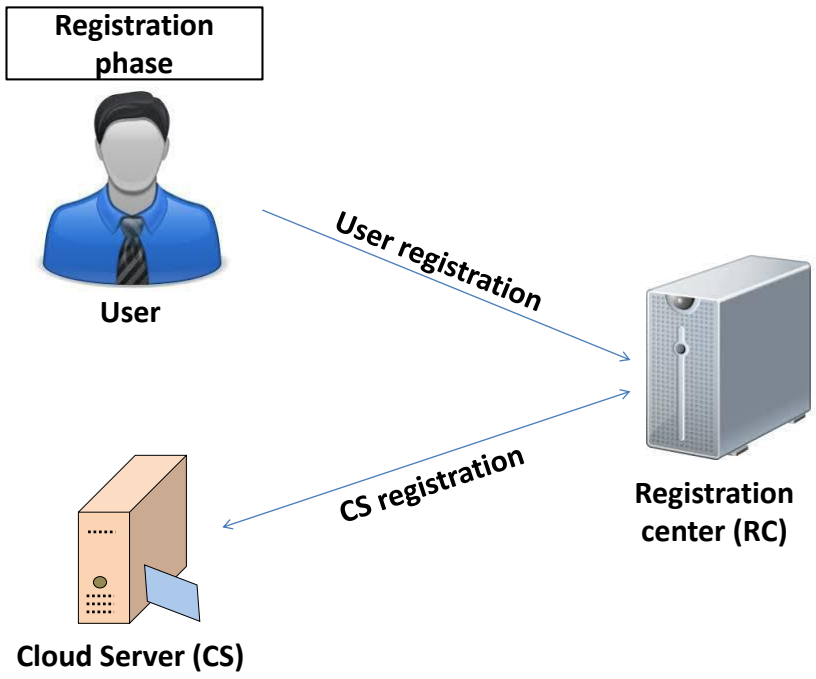


Figure 6.1: Framework of the proposed scheme (registration phase).

6.3 Network model

The proposed scheme is based on the basic assumption that the distributed mobile cloud computing environment has three basic entities: 1) mobile users, 2) cloud server or cloud service provider, and 3) trusted registration center (RC). The system contains a set of m legal mobile users, $M = \{MU_i | i = 1, 2, \cdots, m\}$, a set of n cloud servers, $N = \{CS_j | j = 1, \ldots, n\}$ and the trusted RC. A legal user or an unregistered external person may execute malicious activities in the system, called as an adversary $\mathcal{A}$. From different cloud service providers, a mobile user can access multiple mobile cloud computing services. The RC needs not to

involve in the login and authentication processes. Figures 6.1 and 6.2 present the framework of the proposed scheme.

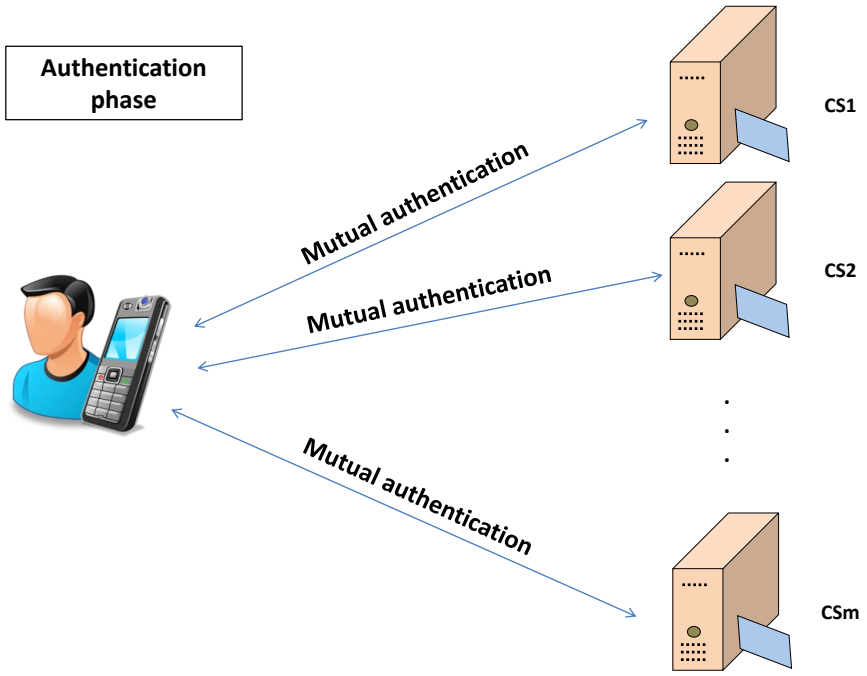


Figure 6.2: Framework of the proposed scheme (authentication phase).

6.4 The proposed scheme

In this section, we describe various phases related to the proposed scheme. The proposed scheme is composed of five phases, namely, 1) registration, 2) login, 3) authentication and key establishment, 4) password change, and 5) mobile device revocation phase. The basic purposes of different phases of the scheme are outlined below.

- **Registration phase:** The registration phase is composed of mobile user registration phase as well as cloud server registration phase. During the registration phase, the mobile users and cloud servers register to the RC independently. The RC generates the master secret keys randomly for the registered servers, and also generates mutual secret keys between respective mobile users and cloud servers.

- **Login phase:** The login phase receives the user's credentials and verifies his/her authenticity. This phase describes how a legal mobile user MU_i logins into the CS_j.

- **Authentication and key establishment phase:** During the authentication and key establishment phase, the mobile user and cloud server authenticate each other and mutually generate the secret shared session key. After successful authentication, MU_i and CS_j establish a secret session key for data communication in the current session.
- **Password change phase:** This phase gives the flexibility to MU_i in order to locally update old password into new password at any time for security reasons.
- **Mobile device revocation phase:** If a legal mobile user MU_i's device is lost or stolen, it is necessary to ensure that in-spite of accessing the stored information, an adversary $\mathcal{A}$ can not make a login to the cloud server. It is required to revoke the lost mobile device and allow MU_i to login using new mobile device.

Table 6.1 contains basic notations of various parameters that are used to design the proposed scheme. The proposed scheme makes use of the current system timestamps along with the random nonces to protect strong replay attacks. To achieve this goal, we assume that all the entities (mobile users, cloud service providers and the RC) in the network are synchronized with their clocks. This is a reasonable assumption as it is also applied in designing many authentication protocols proposed recently [36], [45], [64], [156], [170], [213].

6.4.1 Registration phase

In this phase, both mobile users and cloud servers register to the registration center independently. This phase is composed of two sub-phases: 1) mobile user registration phase and 2) cloud server registration phase. Both the phases are executed only once and messages are communicated through secure channel (for example, in person).

1) Mobile user registration phase

A mobile user MU_i registers to the RC through the following steps:

- **Step MUR1:** MU_i chooses his/her own identity ID_i, password PW_i, biometrics $\mathcal{B}_i$, two 128-bit random numbers b and k.
- **Step MUR2:** MU_i produces $(\theta_i, \phi_i) = Gen(\mathcal{B}_i)$ using the fuzzy extractor probabilistic generation function $Gen(\cdot)$ and computes the masked password $RPWB_i = H(ID_i||\ H(PW_i||\ \theta_i\ ||b))$. MU_i then submits the registration request message $\{ID_i, (RPWB_i \oplus k)\}$ to the RC via secure channel.

Table 6.1: Notations used in the proposed scheme.

Symbol	Description
RC	Registration center
MU_i	i^{th} mobile user
CS_j	j^{th} cloud service provider
ID_i	Identity of MU_i
ID_{S_j}	Identity of CS_j
r_{ij}	1024-bit random number selected by RC for MU_i & CS_j
b	128-bit random number chosen by MU_i
X_j	1024-bit master secret key of server CS_j
SN_i	Serial number of MU_i's mobile device
$H(\cdot)$	One-way cryptographic hash function
$\|\|, \oplus$	Concatenation, bitwise XOR operations
TS_i	Timestamp generated by MU_i
TS_j	Timestamp generated by CS_j
RN_i	128-bit MU_i's random number
RN_j	128-bit CS_j's random number
$A \xrightarrow{\langle M \rangle}$ B	Message (M) transmission from entity A to entity B
$\triangle T$	Maximum transmission delay
$Gen(\cdot)$	Fuzzy extractor probabilistic generation function
$Rep(\cdot)$	Fuzzy extractor deterministic reproduction function
τ	Permissible error tolerance value used in $Rep(\cdot)$ function.

- **Step MUR3:** The RC selects an 1024-bit master secret key X_j for server CS_j. RC also selects an 1024-bit random number r_{ij} for each MU_i and CS_j pair. Further, RC computes $A_{ij} = H(H(ID_i \oplus r_{ij}) \,||X_j)$, $V_{ij} = A_{ij} \oplus RPWB_i$ and the pseudo-identity of CS_j as $RID_{S_j} = H(ID_{S_j}||X_j)$.

- **Step MUR4:** In order to maintain user anonymity, instead of actual identity ID_i, RC selects a unique and random temporary identity TID_i for MU_i.

- **Step MUR5:** The RC saves n server key-plus-id combinations $\{TID_i, (ID_{S_j}, V_{ij}, RID_{S_j}) \mid 1 \leq j \leq n\}$ in mobile device of MU_i and delivers the mobile device to MU_i securely.

- **Step MUR6:** MU_i computes $D_i^1 = H(PW_i||\ \theta_i) \oplus b$ and $D_i^2 = H(ID_i\ ||PW_i\ ||\theta_i\ ||b)$, and $V'_{ij} = V_{ij} \oplus k = A_{ij} \oplus\ H(ID_i||\ H(PW_i||\ \theta_i\ ||b))$, $RID_{ij} = TID_i \oplus\ H(ID_i\ ||V'_{ij})$ and $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i||b)$ for $1 \leq j \leq n$. Finally, MU_i also stores ϕ_i, D_i^1, D_i^2, V'_{ij}s, RID_{ij}s and RID'_{S_j}s into his/her own mobile device, and deletes V_{ij}s, TID_i and RID_{S_j}s from the mobile device.

2) Cloud server registration phase

In order to register to the RC, a new cloud server must execute the following steps:

- **Step CSR1:** The cloud server (cloud service provider) CS_j sends its identity ID_{S_j} to the RC through a secure channel.
- **Step CSR2:** The RC provides the master secret key X_j to each CS_j.
- **Step CSR3:** For all MU_is, the RC saves the credentials $\{TID_i, (ID_i, r_{ij})\}$ in database of CS_j.
- **Step CSR4:** The RC also stores $\{ID_{S_j}, X_j\}$ in the database of CS_j.

Finally, the RC also saves pair (ID_i, SN_i) in its own database, where SN_i is the serial number of MU_i's mobile device. Figure 6.3 shows the fundamental steps of user and server registration phases.

Remark 6.1. *In the proposed scheme, a mobile user MU_i needs to store all the credentials in his/her mobile device. For example, if the mobile user MU_i wants to access* 100 *cloud service providers CS_js, his/her mobile device needs to store* 100 *credentials (i.e., keys). Note that in the proposed scheme, instead of using a smart card, a mobile device is used. Since the mobile device is resource-rich device as compared to resource-constrained smart device, the storage space in MU_i's mobile device is not an issue. Hence, storing more credentials in MU_i's mobile device is not a problem in the proposed scheme.*

6.4.2 Login phase

This phase describes how a legal mobile user MU_i logins into the CS_j. Figure 6.4 shows the basic steps of login and authentication-key establishment phases. The following steps are essential to complete the login phase:

Mobile user (MU_i)	Registration center (RC)
Input ID_i, PW_i and $\mathcal{B}_i$. Select random numbers b and k. Compute $(\theta_i, \phi_i) = Gen(\mathcal{B}_i)$, $RPWB_i = H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b))$. $\xrightarrow[\textbf{(secure channel)}]{\{ID_i, (RPWB_i \oplus k)\}}$	Select secret key X_j for each CS_j. Select r_{ij} for each (MU_i, CS_j) pair. Compute $A_{ij} = H(H(ID_i \oplus r_{ij}) \|\| X_j)$, $V_{ij} = A_{ij} \oplus RPWB_i$, $RID_{S_j} = H(ID_{S_j} \|\| X_j)$. Select temporary identity TID_i for MU_i. For all CS_js, store $\{ID_{S_j}, V_{ij}, TID_i, RID_{S_j}\}$ in memory of MU_i. $\xleftarrow[\textbf{(secure channel)}]{\{Mobile\ Device\}}$
Compute $D_i^1 = H(PW_i \|\| \theta_i) \oplus b$, $D_i^2 = H(ID_i \|\| PW_i \|\| \theta_i \|\| b)$, $V'_{ij} = V_{ij} \oplus k$ $= A_{ij} \oplus H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b))$. For all cloud servers CS_j, compute $RID_{ij} = TID_i \oplus H(ID_i \|\| V'_{ij})$, $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i \|\| b)$. Store $\{\phi_i, D_i^1, D_i^2, V'_{ij}, RID_{ij}$, $RID'_{S_j}\}$ into the memory of MU_i. Delete V_{ij}s, TID_i and RID_{S_j} from its memory.	
Cloud service provider (CS_j)	Registration center (RC)
Select server identity ID_{S_j}. $\xrightarrow[\textbf{Secure channel}]{\{ID_{S_j}\}}$	Select X_j for each CS_j. Save $\{ID_i, TID_i, r_{ij}\}$ in CS_j's memory. Save also $\{ID_{S_j}, X_j\}$ in CS_j's memory. Save pair (ID_i, SN_i) in RC's memory.

Figure 6.3: User and server registration phases of the proposed scheme.

Mobile user (MU_i)	Cloud service provider (CS_j)
Login phase	
Input ID_i, PW_i and $\mathcal{B}'_i$.	
Compute $\theta_i = Rep(\mathcal{B}'_i, \phi_i)$,	
$b' = D_i^1 \oplus H(PW_i \| \theta_i)$.	
Verifies if $D_i^2 = H(ID_i \| PW_i \| \theta_i \| b')$?	
If verification holds, compute	
$RPWB_i = H(ID_i \| \ H(PW_i \| \ \theta_i \ \| b'))$,	
$A_{ij} = V'_{ij} \oplus RPBW_i$.	
Generate RN_i.	
Compute $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$,	
$H_1 = H(ID_i \| C_1 \| RN_i \| TS_i)$,	
$TID_i = RID_{ij} \oplus H(ID_i \| V'_{ij})$,	
$RID_{S_j} = RID'_{S_j} \oplus H(\theta_i \| b')$,	
$TID_i^* = TID_i \oplus H(RID_{S_j} \| TS_i)$.	
$\xrightarrow[\textbf{(public channel)}]{\{TID_i^*, C_1, H_1, TS_i\}}$	
Authentication phase	
	Verify if $\|TS_i^* - TS_i\| \leq \triangle T$?
	Compute $RID_{S_j} = H(ID_{S_j} \| X_j)$,
	$TID_i = TID_i^* \oplus H(RID_{S_j} \| TS_i)$.
	Find $\langle ID_i, TID_i, r_{ij} \rangle$ from database.
	Compute
	$B_{ji} = H(H(ID_i \oplus r_{ij}) \| X_j)$.
	$M_1 = C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji}$.
	$= RN_i$, as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) \| X_j)$,
	$H_2 = H(ID_i \ \| C_1 \ \| M_1 \ \| TS_i)$.
	Verify if $H_2 = H_1$?
	If verification holds, generate RN_j.
	Compute $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$,
	$SK_{CS_j, MU_i} = H(ID_i \| ID_{S_j} \| B_{ji} \| M_1 \| RN_j$,
Verify if $\|TS_j^* - TS_j\| \leq \triangle T$?	$\| TS_i \| TS_j)$
Compute $M_2 = C_2 \oplus TS_j \oplus ID_i \oplus A_{ij}$,	$H_3 = H(ID_i \| M_1 \| RN_j \| TS_i \| TS_j \| SK_{CS_j, MU_i})$.
$= RN_j$, as $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij}) \| X_j)$.	$\xleftarrow[\textbf{(public channel)}]{\{C_2, H_3, TS_j\}}$
$SK_{MU_i, CS_j} = H(ID_i \| ID_{S_j} \| A_{ij} \| RN_i \| M_2 \| TS_i \| TS_j)$,	
$H_4 = H(ID_i \ \| RN_i \ \| M_2 \ \| TS_i \ \| TS_j \ \| SK_{MU_i, CS_j})$.	
Verify if $H_4 = H_3$? If verification holds,	
store session key $SK_{MU_i, CS_j} (= SK_{CS_j, MU_i})$.	Store session key $SK_{CS_j, MU_i} (= SK_{MU_i, CS_j})$.

Figure 6.4: Login and authentication phases of the proposed scheme.

- **Step L1:** MU_i inputs his/her identity ID_i, password PW_i and personal biometrics $\mathcal{B}_i'$ into his/her own mobile device. Using the fuzzy extractor reproduction procedure and stored ϕ_i, MU_i computes $\theta_i = Rep(\mathcal{B}_i', \phi_i)$ provided that the Hamming distance between registered $\mathcal{B}_i$ and currenly entered $\mathcal{B}_i'$ is less or equal to the error tolerance threshold value τ. Moreover, using the stored parameter D_i^1, MU_i generates $b' = D_i^1 \oplus H(PW_i \, ||\theta_i)$.

- **Step L2:** MU_i then computes $H(ID_i \, ||PW_i \, ||\theta_i \, ||b')$ and checks if $D_i^2 = H(ID_i|| \, PW_i|| \, \theta_i|| \, b')$ is true or not. MU_i proceeds to the next step only if this verification holds.

- **Step L3:** MU_i calculates $RPWB_i = H(ID_i|| \, H(PW_i|| \, \theta_i \, ||b'))$. Using the mobile device parameter V_{ij}', MU_i also generates $A_{ij} = V_{ij}' \oplus RPBW_i$. In addition, MU_i selects an 128-bit random number RN_i, generates the current timestamp TS_i, and then computes

$$\begin{aligned} C_1 &= A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j}) \\ &= H(H(ID_i \oplus r_{ij})||X_j) \oplus RN_i \\ &\quad \oplus TS_i \oplus H(ID_{S_j}), \\ H_1 &= H(ID_i||C_1||RN_i||TS_i), \\ TID_i &= RID_{ij} \oplus H(ID_i||V_{ij}'), \\ RID_{S_j} &= RID_{S_j}' \oplus H(\theta_i||b'), \\ TID_i^* &= TID_i \oplus H(RID_{S_j}||TS_i). \end{aligned}$$

- **Step L4:** Finally, MU_i sends login request $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j via a public channel.

Remark 6.2. *The current timestamp TS_i in a particular session is used to make TID_i^* as dynamic, because $TID_i^* = TID_i \oplus H(RID_{S_j}||TS_i)$. In addition, even if an adversary $\mathcal{A}$ eavesdrops $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and gets TID_i^*, it is computationally infeasible problem for $\mathcal{A}$ to know TID_i from TID_i^* without having the permanent secret RID_{S_j} $(= H(ID_{S_j}||X_j))$ as it involves the secret key X_j of CS_j. Suppose the same mobile user MU_i sends the login message $Msg_1' = \{TID_i^{**}, C_1', H_1', TS_i'\}$ to CS_j in another session, where $TID_i^{**} = TID_i \oplus H(RID_{S_j}||TS_i')$ and TS_i' is the current timestamp generated by MU_i in that session. In this case, $\mathcal{A}$ can not also derive TID_i from TID_i^{**} without having the permanent secret RID_{S_j} of CS_j. It is also observed that both TID_i^* and TID_i^{**} are distinct due to involvement of RID_{S_j} and current timestamps. This clearly shows that the user anonymity is*

completely preserved in the proposed scheme as the real identity ID_i as well as the temporary identity TID_i of MU_i are not revealed to the adversary $\mathcal{A}$.

6.4.3 Authentication and key establishment phase

In this phase, CS_j and MU_i mutually authenticate each other. After successful authentication, MU_i and CS_j establish a secret session key for data communication in the current session. This phase involves the following steps:

- **Step AKE1:** After receiving the login request message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ from MU_i, CS_j verifies if $|TS_i^* - TS_i| \leq \triangle T$, where TS_i^* is the actual received time of the message Msg_1 and $\triangle T$ is the maximum transmission delay. If this verification fails, CS_j rejects the login request immediately; otherwise, CS_j proceeds to the next step.

- **Step AKE2:** CS_j computes $RID_{S_j} = H(ID_{S_j}||X_j)$ and then extracts $TID_i = TID_i^* \oplus H(RID_{S_j}||TS_i)$ and then finds the record $\langle ID_i, r_{ij} \rangle$ from its database corresponding to TID_i. Using ID_{S_j} and the master key X_j, CS_j computes

$$\begin{aligned} B_{ji} &= H(H(ID_i \oplus r_{ij})||X_j), \\ M_1 &= C_1 \oplus TS_i \oplus H(ID_{S_j}) \oplus B_{ji} \\ &= A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j}) \oplus TS_i \\ &\quad \oplus H(ID_{S_j}) \oplus B_{ji} \\ &= RN_i, \text{ as } A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij})||X_j). \end{aligned}$$

- **Step AKE3:** Using computed parameter M_1 and received parameters $\{C_1, TS_i\}$, CS_j generates the hash value $H_2 = H(ID_i||C_1||M_1||TS_i)$. CS_j then verifies whether computed hash value $H_2 \stackrel{?}{=} H_1$. Failure of the condition terminates the current session. Otherwise, CS_j accepts the login request and proceeds to the next step. CS_j also saves record $\langle ID_i, RN_i, TS_i \rangle$ in its database to resist strong replay attack. For instance, if CS_j receives another login request message, say $Msg_1' = \{TID_i^*, C_1', H_1', TS_i'\}$ next time, it first checks the validity of TS_1'. If it is valid, CS_j further verifies if the extracted $RN_i' = C_1' \oplus TS_i' \oplus H(ID_{S_j}) \oplus B_{ji}$ matches with the stored RN_i in its database corresponding to ID_i. If it is present, Msg_1' is treated as a replay message. Thus, in the proposed scheme, to protect replay attack strongly we verify both timestamp as well as random nonce embedded in the login request message. Note that the schemes based on only random

nonces do not provide strong replay attack as demonstrated by several researchers in the literature [55], [130].

- **Step AKE4:** CS_j then selects an 128 bit random number RN_j and computes $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$, where TS_j is the current timestamp generated by CS_j. Further, CS_j computes the secret session key shared with MU_i as $SK_{S_j,MU_i} = H(ID_i \,||ID_{S_j}\, ||B_{ji}\, ||M_1\, ||RN_j\, ||TS_i\, ||TS_j)$, which is used for future message communication with MU_i. Finally, CS_j generates a hash value $H_3 = H(ID_i\, ||M_1\, ||RN_j\, ||TS_i\, ||TS_j\, ||SK_{CS_j,MU_i})$ and sends the authentication request message $Msg_2 = \{C_2,\, H_3,\, TS_j\}$ to MU_i via a public channel.

- **Step AKE5:** After MU_i receiving the authentication request message Msg_2 at time TS_j^* from CS_j, the condition $|TS_j^* - TS_j| \leq \triangle T$ is verified. If message transmission delay is within allowable limit, MU_i computes

$$\begin{aligned} M_2 &= C_2 \oplus TS_j \oplus ID_i \oplus A_{ij} \\ &= B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i \oplus TS_j \oplus ID_i \oplus A_{ij} \\ &= RN_j, \text{ as } A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij})||X_j). \end{aligned}$$

- **Step AKE6:** Using the received timestamp TS_j and computed parameter M_2, MU_i generates the session key shared with CS_j as $SK_{MU_i,CS_j} = H(ID_i\, ||ID_{S_j}\, ||A_{ij}\, ||RN_i\, ||M_2\, ||TS_i\, ||TS_j)$. Further, MU_i generates a hash value $H_4 = H(ID_i\, ||RN_i\, ||M_2\, ||TS_i\, ||TS_j\, ||SK_{MU_i,CS_j})$ and verifies whether $H_4 \stackrel{?}{=} H_3$. If the verification succeeds, MU_i assumes that the shared secret session key SK_{MU_i,CS_j} $(= SK_{CS_j,MU_i})$ is mutually verified and established. This is used for future message communication with CS_j in the current session.

The summary of the authentication phase of the proposed scheme is also provided in Figure 6.4.

Remark 6.3. *To speed up the searching of the credentials $\{ID_i, r_{ij}\}$ corresponding to the computed TID_i in the database of CS_j, the following procedure can be adapted. The credentials $\{TID_i, (ID_i, r_{ij})\}$ in the database of CS_j can be sorted in ascending order according to the key value TID_i by the RC during the cloud server registration phase in Section 6.4.1. Then, we can perform the binary search algorithm to find the credentials $\{ID_i, r_{ij}\}$ corresponding to TID_i, which is executed in $O(\log_2(n))$ time complexity where n is number of CS_js for each*

user MU_i. Hence, it is clear that the time for searching $\{ID_i, r_{ij}\}$ corresponding to TID_i by CS_j is not heavy even if n is large while implementing the proposed scheme for practical application, because CS_j is not resource-limited entity in the network.

6.4.4 Password change phase

The password change phase causes MU_i to update original password PW_i by the new password PW_i'. Note that this phase does not involve the RC or any CS_j and it is completely done locally. This phase requires the following steps:

- **Step PC1:** MU_i enters his/her old password PW_i along with identity ID_i and biometrics $\mathcal{B}_i'$.
- **Step PC2:** Using the fuzzy extractor reproduction procedure and stored ϕ_i, MU_i computes $\theta_i = Rep(\mathcal{B}_i', \phi_i)$. Moreover, using the stored parameter D_i^1, MU_i generates $b' = D_i^1 \oplus H(PW_i \,||\theta_i)$.
- **Step PC3:** MU_i then computes $H(ID_i \,||PW_i \,||\theta_i \,||b')$ and checks if $D_i^2 = H(ID_i||\ PW_i||\ \theta_i||\ b')$ is true or not. MU_i proceeds to the next step only if this verification holds.
- **Step PC4:** MU_i enters new password PW_i' and computes $D_i^{1*} = D_i^1 \oplus H(PW_i \,||\theta_i) \oplus H(PW_i' \,||\theta_i)$. Further, MU_i computes $D_i^{2*} = H(ID_i \,||PW_i' \,||\theta_i \,||b')$, $V_{ij}^* = V_{ij}' \oplus H(ID_i \,||H(PW_i \,||\theta_i \,||b')) \oplus H(ID_i \,||H(PW_i' \,||\theta_i \,||b'))$, $TID_i = RID_{ij} \oplus H(ID_i||\ V_{ij}')$ and $RID_{ij}^* = TID_i \oplus H(ID_i \,||V_{ij}^*)$ for $1 \leq j \leq n$.
- **Step PC5:** The user mobile device updates D_i^1 with D_i^{1*}, D_2 with D_i^{2*}, V_{ij}' with V_{ij}^* and RID_{ij} with RID_{ij}^*, in its memory.

The password change phase of the proposed scheme is summarized in Figure 6.5.

6.4.5 Mobile device revocation phase

If a legal mobile user MU_i's device is lost or stolen, it is necessary to ensure that in-spite of accessing the stored information, an adversary $\mathcal{A}$ can not make a login to the cloud server. It is required to revoke the lost mobile device and allow MU_i to login using new mobile device. For this purpose, the following steps are executed:

Mobile user (MU_i)	Mobile device
Input ID_i, PW_i and $\mathcal{B}_i'$. $\xrightarrow{\{ID_i, PW_i, \mathcal{B}_i'\}}$	
	Compute $\theta_i = Rep(\mathcal{B}_i', \phi_i)$, $b' = D_i^1 \oplus H(PW_i \|\| \theta_i)$. Verify if $D_i^2 = H(ID_i \|\| PW_i \|\| \theta_i \|\| b')$? If verification holds, request to supply new password.
Input new password PW_i'. $\xrightarrow{\{PW_i'\}}$	
	Compute $D_i^{1*} = D_i^1 \oplus H(PW_i \|\| \theta_i) \oplus H(PW_i' \|\| \theta_i)$, $D_i^{2*} = H(ID_i \|\| PW_i' \|\| \theta_i \|\| b')$, $V_{ij}^* = V_{ij}' \oplus H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b'))$ $\oplus H(ID_i \|\| H(PW_i' \|\| \theta_i \|\| b'))$, $TID_i = RID_{ij} \oplus H(ID_i \|\| V_{ij}')$. For all servers CS_js, compute $RID_{ij}^* = TID_i \oplus H(ID_i \|\| V_{ij}^*)$. Update $D_i^1 \leftarrow D_i^{1*}$, $D_2 \leftarrow D_i^{2*}$, $V_{ij}' \leftarrow V_{ij}^*$ and $RID_{ij} \leftarrow RID_{ij}^*$.

Figure 6.5: Password change phase of the proposed scheme.

- **Step MDR1:** MU_i enters ID_i, PW_i and imprints biometrics $\mathcal{B}_i$, and also generates two new 128-bit random numbers b' and k'.

- **Step MDR2:** MU_i produces $(\theta_i, \phi_i) = Gen(\mathcal{B}_i)$ and computes the masked password $RPWB_i = H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b'))$. MU_i then submits the registration request message $\langle ID_i, (RPWB_i \oplus k') \rangle$ to the RC via secure channel.

- **Step MDR3:** The RC verifies authenticity of MU_i by checking his/her other credentials, such as date of birth (DOB) and registered id number.The RC selects an 1024-bit random number r_{ij}' for each MU_i and CS_j pair. Further, RC computes $A_{ij} = H(H(ID_i \oplus r_{ij}') \|\| X_j)$ and $V_{ij} = A_{ij} \oplus RPWB_i$.

- **Step MDR4:** In order to maintain user anonymity, instead of actual identity ID_i, the RC selects a unique and random temporary identity TID'_i for MU_i.

- **Step MDR5:** The RC saves n server key-plus-id combinations $\{(ID_{S_j}, V_{ij}, TID'_i) \mid 1 \leq j \leq n\}$ in mobile device of MU_i and delivers the mobile device to MU_i securely.

- **Step MDR6:** MU_i computes $D_i^1 = H(PW_i|| \theta_i) \oplus b'$ and $D_i^2 = H(ID_i ||PW_i ||\theta_i ||b')$, and $V'_{ij} = V_{ij} \oplus k' = A_{ij} \oplus H(ID_i|| H(PW_i|| \theta_i ||b'))$ and $RID_{ij} = TID_i \oplus H(ID_i ||V'_{ij})$ for $1 \leq j \leq n$. Finally, MU_i stores ϕ_i, D_i^1, D_i^2, V'_{ij}s, RID_{ij}s and RID'_{S_j}s ($RID'_{S_j} = RID_{S_j} \oplus H(\theta_i||b')$) into his/her own mobile device, and deletes V_{ij}s and TID'_i from the mobile device.

- **Step MDR7:** All servers CS_js also update (ID_i, TID_i, r_{ij}) with (ID_i, TID'_i, r'_{ij}) in their databases after receiving the update request message securely from the RC.

The mobile device revocation phase the proposed scheme is summarized in Figure 6.6.

6.5 Security analysis

In this section, we provide both formal security and informal security analysis of the proposed scheme. In Section 6.5.1, formal security analysis is done using Real-Or-Random (ROR) Model. In Section 6.5.2, we provide formal authentication proof using BAN logic. In Section 6.5.3, through informal security analysis, we discuss on how the proposed scheme resists various other security threats and attacks.

6.5.1 Formal security using ROR model

In this section, formal security analysis of the proposed mobile user authentication protocol, say $\mathscr{P}$, is done using the widely-accepted Real-Or-Random (ROR) model [14], [214].

1) Outline of ROR model

An adversary $\mathcal{A}$ can make several oracle queries, which model the adversary capabilities in a real attack [26], [185]. Table 6.2 contains brief descriptions of various oracle queries that are used in this proof. We assume that $\mathcal{A}$ interacts with $\mathcal{P}^t$, the t^{th} instance of an executing participant (MU_i or CS_j).

Mobile user (MU_i)	Registration center (RC)
Input ID_i, PW_i and $\mathcal{B}_i$. Input random number b', and k'. Compute $(\theta_i, \phi_i) = Gen(\mathcal{B}_i)$. $RPWB_i = H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b'))$. $\{ID_i, (RPWB_i \oplus k')\}$ $\longrightarrow$ **(secure channel)**	
	Verify MU_i by SSN, DOB etc. $\forall$ CS_j, select r'_{ij} randomly. Compute $A_{ij} = H(H(ID_i \oplus r'_{ij}) \|\| X_j)$ $V_{ij} = A_{ij} \oplus RPWB_i$. Select temporary id TID'_i for MU_i. $\forall$ CS_j, load : $\{ID_{S_j}, V_{ij}, TID'_i\}$ in MU_i. $\{$mobile device$\}$ $\longleftarrow$ **(secure channel)**
Compute $D_i^1 = H(PW_i \|\| \theta_i) \oplus b'$, $D_i^2 = H(ID_i \|\| PW_i \|\| \theta_i \|\| b')$, $V'_{ij} = V_{ij} \oplus k' = A_{ij} \oplus$, $H(ID_i \|\| H(PW_i \|\| \theta_i \|\| b'))$. $\forall$ CS_j, compute $RID_{ij} = TID_i \oplus H(ID_i \|\| V'_{ij})$, $RID'_{S_j} = RID_{S_j} \oplus H(\theta_i \|\| b')$. Store $\{ \phi_i, D_i^1, D_i^2, V'_{ij}$s, RID_{ij}s RID'_{S_j}s$\}$. Delete V_{ij}s , TID'_i from memory.	
RC informs all servers CS_js to update their database. **All server CS_j update $(ID_i, TID_i, r_{ij}) \leftarrow (ID_i, TID'_i, r'_{ij})$.**	

Figure 6.6: Mobile device revocation phase the proposed scheme.

Definition 6.1 (Semantic security [26]). *Let $Adv_{\mathscr{P}}^{MUAP}$ denote the advantage of $\mathcal{A}$ running in polynomial time in breaking the semantic security of proposed mobile user authentication protocol (MUAP), referred as $\mathscr{P}$. Then, $Adv_{\mathscr{P}}^{MUAP} = |2Pr[b' = b] - 1|$, where b' is the guessed*

Table 6.2: Different oracle queries and their descriptions.

Query	**Description/purpose**
$Send(\mathcal{P}^t, m)$	It enables $\mathcal{A}$ to send request message m to $\mathcal{P}^t$ and $\mathcal{P}^t$ replies accordingly
$Corrupt(MU_i, a)$	Depending on a, $\mathcal{A}$ can obtain biometric and password of MU_i
$Test(\mathcal{P}^t)$	$\mathcal{A}$ requests $\mathcal{P}^t$ for the session key SK, $\mathcal{P}^t$ replies probabilistically on outcome of a flipped coin b
$Execute(MU_i, CS_j)$	It enables $\mathcal{A}$ to eavesdrop the messages communicated between MU_i and CS_j
$Reveal(\mathcal{P}^t)$	It enables $\mathcal{A}$ to obtain the session key SK generated between $\mathcal{P}^t$ and its partner

bit.

Definition 6.2. *The proposed protocol $\mathscr{P}$ is semantically secure if the advantage function $Adv_{\mathscr{P}}^{MUAP}$ is only negligibly larger than $\max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$, where q_s, l_b, C' and s' denote their usual meanings as tabulated in Table 6.3.*

2) Security proof

For the formal security proof, we use the notations listed in Table 6.3. Recent research has shown that user-chosen passwords (also termed as "weak secrets") follow the Zipf's law [202], which is a vastly different distribution from the uniform distribution. Actually, the size $|\mathcal{D}|$ of password dictionary $\mathcal{D}$ is generally much constrained in the sense that the users will not use the whole space of passwords, but rather a small space of the allowed characters space [202]. On the other hand, even if we consider only trawling guessing attacks, actually the advantage of an adversary will be over 0.5 when $q_s = 10^7$ or 10^8 [32], [202]. When further considering targeted guessing attacks in which the adversary can make use of the target user's personal information, the advantage of the adversary will be over 0.5 when $q_s \leq 10^6$ [210].

Theorem 6.1. *Suppose $Adv_{\mathscr{P}}^{MUAP}$ denotes the advantage function of an adversary $\mathcal{A}$ in breaking the semantic security of the proposed scheme $\mathscr{P}$ as defined in Definition 6.2. Then,*

$$Adv_{\mathscr{P}}^{MUAP} \leq \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} + 2\max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\},$$

where q_H, q_s, q_e, l_H, l_r, l_b, ε_{bm}, C' and s' have their usual meanings as tabulated in Table 6.3.

Table 6.3: Symbols used in the Real-Or-Random (ROR) model.

Symbol	Description
q_H	Total number of hash H oracle queries execution
q_s	Total number of *Send* oracle queries execution
q_e	Total number of *Execute* oracle queries execution
l_H	Length of hash output string
l_r	Length of random number string
l_b	Length of user biometric key
ε_{bm}	Probability of false positive in biometrics [158]
$\mathcal{D}$	Password space with its frequency distribution following Zipf's law [202]
C', s'	Zipf's parameters [202]
L_H	List that stores output of hash H oracle query
L_A	List that records random oracle outputs
L_T	List that records message transcripts between MU_i and CS_j

Proof. We follow the similar proof as in [45], [170]. The proof is composed of five games Gm_i $(i = 0, 1, 2, 3, 4)$. In a game Gm_i, an adversary $\mathcal{A}$ tries to guess a correct bit b through the $Test$ query. This event is defined as S_i and the corresponding probability is denoted by $Pr[S_i]$.

- **Game Gm_0:** The initial game Gm_0 is considered to be identical with the actual protocol executing under the ROR model. Hence, we have,

$$Adv_{\mathscr{P}}^{MUAP} = |2Pr[S_0] - 1|. \tag{6.1}$$

- **Game Gm_1:** This game considers simulation of *Send*, *Test*, *Execute*, *Reveal*, and *Corrupt* queries with respect to the proposed scheme. Table 6.4 describes the working procedure of *Execute* and *Send* query. Further, this game considers lists L_H, L_A, and L_T for storing results of various oracle queries. Due to indistinguishability of Gm_0 and Gm_1, we obtain,

$$Pr[S_1] = Pr[S_0]. \tag{6.2}$$

- **Game Gm_2:** The collision probability of random oracle query and hash (H) oracle query are considered in this game for all the communicated messages between MU_i and CS_j. In $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and $Msg_2 = \{C_2, H_3, TS_j\}$, MU_i and CS_j use random numbers RN_i and RN_j, and also the current timestamps TS_i and TS_j,

respectively. This causes collision probability at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. Moreover, based on the birthday paradox, use of H oracle query results in collision probability of $\frac{q_H^2}{2^{l_H+1}}$. Overall, we obtain,

$$|Pr[S_2] - Pr[S_1]| \leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_H^2}{2^{l_H+1}}. \tag{6.3}$$

Table 6.4: Simulation of Execute and Send oracle queries.

Simulation of *Execute*(MU_i, CS_j) query occurs in succession with simulation of *Send* queries as given below. Compute C_1 and H_1 as given in Figure 6.4. MU_i sends the message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j. Compute C_2, H_3 and SK_{CS_j,MU_i} as given in Figure 6.4. CS_j the sends authentication message $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i. Note that $\langle TID_i^*, C_1, H_1, TS_i\rangle \leftarrow Send(MU_i, \textbf{start})$, $\langle C_2, H_3, TS_j\rangle \leftarrow Send(CS_j, \langle TID_i^*, C_1, H_1, TS_i\rangle)$. Finally, $Msg_1 =$ and Msg_2 are returned.
Send query simulation is done as per the proposed scheme: **(a)** On $Send(MU_i, \textbf{start})$ query, MU_i responses as follows. Compute TID_i^*, C_1, H_1, TS_i as in Figure 6.4. Output $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$. **(b)** Over $Send(CS_j, \langle TID_i^*, C_1, H_1, TS_i\rangle)$ query, CS_j responds as follows. Test if $\lvert TS_i^* - TS_i\rvert \leq \triangle T$ and then generates B_{ji} and M_1. Also, verify the parameter H_1. Terminate the current session if verification fails. Moreover, CS_j computes C_2, SK_{CS_j,MU_i} and H_3, and output $Msg_2 = \{C_2, H_3, TS_j\}$. **(c)** MU_i answers $Send(MU_i, \langle C_2, H_3, TS_j\rangle)$ query mentioned below. Check if $\lvert TS_j^* - TS_j\rvert \leq \triangle T$. If verification passes, compute M_2 and SK_{MU_i,CS_j}. Finally, verify H_3. If verification fails, terminate the current session. Otherwise, compute and accept SK_{MU_i,CS_j} as the session key as depicted in Figure 6.4. On establishment of the shared session key, both MU_i & CS_j terminate the session.

- **Game** Gm_3**:** Since H hash oracle query is already considered in the game Gm_2, we need to calculate collision probability from all other remaining oracle queries. Now, we consider the following two cases:

– **Case 1:** After executing $Send(CS_j, Msg_1)$ query on $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$, it is noted that $H_1 = H(ID_i \,||C_1\, ||RN_i\, ||TS_i) \in L_{\mathcal{A}}$ has collision probability at most $\frac{q_H}{2^{l_H}}$. To launch attack successfully, $H(ID_i||\ \theta_i) \oplus b$ of D_i^1, $H(ID_i\ ||PW_i||\ \theta_i\ ||b)$ of D_i^2 and $A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ of C_1 should be revealed to $\mathcal{A}$. This results in the total collision probability up to $\frac{4q_H}{2^{l_H}}$. Moreover, as transcript message Msg_1 contains RN_i, $Msg_1 \in L_T$ must hold with probability up to $\frac{q_s}{2^{l_r}}$.

– **Case 2:** Considering $\mathcal{A}$ executes the query $Send(MU_i, Msg_2)$ and $H_3 \in L_{\mathcal{A}}$ holds, the calculated probability becomes $\frac{q_H}{2^{l_H}}$. Furthermore, CS_j computes $H(H(ID_i \oplus r_{ij})\ ||X_j)$ for B_{ji}, $H(ID_{S_j})$ in M_1, verifies $H(ID_i\ ||C_1\ ||M_1\ ||TS_i)$ with H_1, and finally, it computes $SK_{CS_j,MU_i} = H(ID_i||\ ID_{S_j}\ ||B_{ji}\ ||M_1\ ||RN_j\ ||TS_i\ ||TS_j)$. Hence the total probability for this part is $\frac{5q_H}{2^{l_H}}$. Due to the message transcript $Msg_2 \in L_T$, we obtain $\frac{q_s}{2^{l_r}}$ as the collision probability.

As a whole, we obtain,

$$|Pr[S_3] - Pr[S_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}}. \tag{6.4}$$

- **Game** Gm_4**:** In game Gm_4, by exploiting $Corrupt$ query, the adversary $\mathcal{A}$ tries to guess user's private credentials like password and biometric in online as well as offline modes. The guessing of biometric has maximum probability up to $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$ [45], [170] and that for password is $C' \cdot q_s^{s'}$ [202]. Since the games Gm_3 and Gm_4 are identical when these guessing attacks are absent, we have,

$$|Pr[S_4] - Pr[S_3]| \ \leq\ \max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \tag{6.5}$$

After executing all the games, $\mathcal{A}$ is only left in guessing the correct bit b. It is then clear that

$$Pr[S_4] = \frac{1}{2}. \tag{6.6}$$

Applying the law of triangular inequality, we have,

$$\begin{aligned} |Pr[S_0] - \frac{1}{2}| &= |Pr[S_1] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad +|Pr[S_3] - Pr[S_4]|. \end{aligned} \tag{6.7}$$

Using Equations (6.1)-(6.7), we obtain,

$$\begin{aligned}\frac{1}{2}Adv_{\mathscr{P}}^{MUAP} &= |Pr[S_0] - \frac{1}{2}| \\ &\leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_H^2}{2^{l_H+1}} + \frac{2q_s}{2^{l_r}} + \frac{9q_H}{2^{l_H}} \\ &\quad + \max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned} \tag{6.8}$$

Multiplying both sides of Equation (6.8) by a factor of 2 and then rearranging the terms, we finally obtain

$$Adv_{\mathscr{P}}^{MUAP} \leq \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s+q_e)^2 + 4q_s}{2^{l_r}} + 2\max\{C' \cdot q_s^{s'}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}.$$

Hence, the theorem is proved. □

6.5.2 Mutual authentication proof using BAN logic

BAN logic is used to mutual authentication between two communicating parties in a network [34]. Using the broadly-used BAN logic, we show that the proposed scheme achieves authentication goals discussed below. Basic BAN logic notations and logical postulates are provided in Section 2.6.

To complete the authentication proof, the proposed scheme must meet the following two goals:

- **Goal 1**. $MU_i \mid\equiv (MU_i \overset{SK}{\longleftrightarrow} CS_j)$.
- **Goal 2**. $CS_j \mid\equiv (MU_i \overset{SK}{\longleftrightarrow} CS_j)$.

The generic types of the messages in the proposed scheme are given below.

- **Message 1.** $MU_i \rightarrow CS_j$: $\{TID_i^*, H(H(ID_i \oplus r_{ij}) \,||X_j) \oplus RN_i \oplus TS_i \oplus H(ID_{S_j}), TS_i, H_1\}$.
- **Message 2.** $CS_j \rightarrow MU_i : \{B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i, TS_j, H_3\}$.

The the idealized forms of the above generic messages are provided mentioned below.

- **Message 1.** $MU_i \rightarrow CS_j$: $\{TID_i, TS_i, \langle ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})\rangle_{X_j}, H_1\}$.
- **Message 2.** $CS_j \rightarrow A$: $\{TS_j, \langle RN_j, TS_j, ID_i\rangle_{X_j}, H_3\}$.

The authentication proof of the proposed scheme starts with the following basic assumptions:

- **A.1:** $MU_i \mid\equiv \#(TS_j)$
- **A.2:** $CS_j \mid\equiv \#(TS_i)$
- **A.3:** $MU_i \mid\equiv (MU_i \overset{A_{ij}}{\rightleftharpoons} CS_j)$
- **A.4:** $CS_j \mid\equiv (MU_i \overset{A_{ij}}{\rightleftharpoons} CS_j)$
- **A.5:** $MU_i \mid\equiv CS_j \Rightarrow (ID_{S_j}, RN_j, TS_j)$
- **A.6:** $CS_j \mid\equiv MU_i \Rightarrow (ID_i, RN_i, TS_i)$
- **A.7:** $MU_i \mid\equiv TS_i$
- **A.8:** $MU_i \mid\equiv RN_i$
- **A.9:** $MU_i \mid\equiv ID_i$
- **A.10:** $MU_i \mid\equiv ID_{S_j}$
- **A.11:** $CS_j \mid\equiv TS_j$
- **A.12:** $CS_j \mid\equiv RN_j$
- **A.13:** $CS_j \mid\equiv ID_{S_j}$

Considering these basic assumptions, idealized forms and fundamental logical postulates, in the following we show the achievement of both the goals **Goal 1** and **Goal 2**.

According to the message 1, we obtain,

- S_1: $CS_j \lhd \{ID_i, TS_i, \langle ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})\rangle_{X_j}, H_1\}$.
- S_2: According to the law AL, we obtain, $CS_j \lhd \langle ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j})\rangle_{X_j}$.
- S_3: According to A.4 and MML, we obtain, $CS_j \mid\equiv MU_i \mid\sim (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}))$.
- S_4: According to A.2 and FCL, we get, $CS_j \mid\equiv \#(ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}))$.
- S_5: According to NVL, we have, $CS_j \mid\equiv MU_i \mid\equiv (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}))$.

- S_6: Using A.6 and JL, we get, $CS_j \mid\equiv (ID_i, r_{ij}, RN_i, TS_i, H(ID_{S_j}))$.
- S_7: From S_6 and AL, we obtain, $CS_j \mid\equiv RN_i$, $CS_j \mid\equiv TS_i$, $CS_j \mid\equiv ID_i$.
- S_8: According to A.11, A.12, A.13, we get, $CS_j \mid\equiv ID_{S_j}$, $CS_j \mid\equiv TS_j$ and $CS_j \mid\equiv RN_j$.
- S_9: Since $SK_{CS_j,MU_i} = H(ID_i \mid\mid ID_{S_j} \mid\mid B_{ji} \mid\mid M_1 \mid\mid RN_j \mid\mid TS_i \mid\mid TS_j)$ and the results in Steps S_7 and S_8 give $CS_j \mid\equiv (MU_i \overset{SK_{CS_j,MU_i}}{\longleftrightarrow} CS_j)$. **(Goal 2)**
- S_{10}: Using the message 2 and AL, we obtain, $MU_i \lhd \langle RN_j, TS_j \rangle_{X_j}$.
- S_{11}: According to A.3 and MML, we get, $MU_i \mid\equiv CS_j \mid\backsim (RN_j, TS_j)$.
- S_{12}: Using A.1 and FCL, we obtain, $MU_i \mid\equiv \#(RN_j, TS_j)$.
- S_{13}: Using NVL, we obtain, $MU_i \mid\equiv CS_j \mid\equiv (RN_j, TS_j)$.
- S_{14}: A.5 and JL give $MU_i \mid\equiv (RN_j, TS_j)$.
- S_{15}: According to S_{14} and AL, we have, $MU_i \mid\equiv RN_j$, $MU_i \mid\equiv TS_j$.
- S_{16}: According to A.7-A.10, we obtain, $MU_i \mid\equiv ID_i$, $MU_i \mid\equiv ID_{S_j}$, $MU_i \mid\equiv TS_i$, $MU_i \mid\equiv RN_j$.
- S_{17}: The results of Steps S_{15} and S_{16} give $MU_i \mid\equiv (MU_i \overset{SK_{CS_j,MU_i}}{\longleftrightarrow} CS_j)$. **(Goal 1)**

As a result, **Goal 1** and **Goal 2** ensure that both MU_i and CS_j mutually authenticate each other.

6.5.3 Discussion on other attacks

This section also informally analyzes the security of the proposed scheme in order to show that it can defend the following other known attacks.

1) Replay attack

According to the proposed scheme, the login and authentication phases require two message communications. In login phase, MU_i sends $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ to CS_j, whereas in authentication phase, CS_j sends $Msg_2 = \{C_2, H_3, TS_j\}$ to MU_i. CS_j does not accept Msg_1 if $|TS_i^* - TS_i| > \triangle T$. To resist replay attack, CS_j further computes $H(ID_i \mid\mid C_1 \mid\mid M_1 \mid\mid TS_i)$

and verifies it with received hash value as $H_1 \stackrel{?}{=} H(ID_i||\ C_1||\ M_1||\ TS_i)$. The cloud server CS_j rejects login request if this verification fails. As explained in Step 6 of authentication phase in Section 6.4.3, an attacker also fails to replay the authentication message Msg_2. In addition, CS_j also stores parameters $\langle ID_i, RN_i, TS_i\rangle$ in its database to resist strong replay attack. If CS_j receives another login request message, say $Msg_1' = \{TID_i^*,\ C_1',\ H_1',\ TS_i'\}$ next time, it first checks the validity of TS_1'. If it is valid, CS_j further verifies if the extracted $RN_i' = C_1' \oplus TS_i' \oplus H(ID_{S_j}) \oplus B_{ji}$ matches with the stored RN_i in its database corresponding to ID_i. If it is present, Msg_1' is treated as a replay message. As a whole, the proposed scheme protects strong replay attack because both the current timestamp and random nonce are applied.

2) Man-in-the-middle attack

An adversary $\mathcal{A}$ may try to launch man-in-the-middle attack in order to set up a third party independent connection with both MU_i and CS_j for a particular session. Moreover, $\mathcal{A}$ might intend to modify public message parameters to invalidate a login request of a legal user. The proposed scheme uses hash function, random nonce and bitwise XOR operation in both message MSg_1 and Msg_2. However, $\mathcal{A}$ can not modify any message as these need the credentials, such as A_{ij}, ID_{S_j}, V_{ij}' and RID_{ij}. This causes the proposed scheme to resist the man-in-the-middle attack.

3) Stolen/lost mobile device attack

The user mobile device contains $D_i^1 = H(PW_i\ ||\theta_i) \oplus b$ and $D_i^2 = H(ID_i\ ||PW_i\ ||\theta_i\ ||b)$. As guessing of ID_i, PW_i and biometrics $\mathcal{B}_i$ from D_i^1 and D_i^2 is computationally infeasible, $\mathcal{A}$ can not obtain these credentials from user mobile device. Further, user mobile device contains $V_{ij}' = A_{ij} \oplus RPWB_i$, where $A_{ij} = H(H(ID_i \oplus r_{ij})\ ||X_j)$ and $RPWB_i = H(ID_i\ ||H(PW_i\ ||\theta_i\ ||b))$. Since r_{ij} and b are random numbers and $H(\cdot)$ is a collision-resistant, it is computationally infeasible problem to obtain ID_i, PW_i and θ_i from V_{ij}' and A_{ij}' in polynomial time. Hence, the proposed scheme resists this attack.

4) Offline password guessing attack

According to user registration phase described in Section 6.4.1, mobile device of MU_i contains $\langle D_i^1,\ D_i^2,\ \phi_i,\ \{(ID_{S_j},\ V_{ij}',\ RID_{ij},\ RID_{S_j}')\ |\ 1 \le j \le n\}\rangle$. As discussed in Section 6.5.3, $\mathcal{A}$ can not guess password from any stored parameters like D_i^1, D_i^2 and V_{ij}' as PW_i is masked with

θ_i and random secret b. To obtain PW_i, $\mathcal{A}$ needs to guess these parameters simultaneously, which have negligible probability. So, this kind of attack is resisted by the proposed scheme.

5) Forward secrecy

Forward secrecy (also known as known key secrecy) ensures that a compromised session key does not help an adversary to compute past session keys. According to the proposed scheme, the session key is mutually computed as $SK_{MU_i,CS_j} = SK_{CS_j,MU_i} = H(ID_i \,||ID_{S_j}\, ||A_{ij}\, ||RN_i\, ||RN_j\, ||TS_i\, ||TS_j)$ where $A_{ij} = B_{ji} = H(H(ID_i \oplus r_{ij})\, ||X_j)$. Due to the use of RN_i, TS_i, RN_j, and TS_j, for every new login session, SK_{MU_i,CS_j} $(= SK_{CS_j,MU_i}$ is generated in random but in a unique way. As a consequence, compromise of the current session key provides no crucial information to the adversary that helps him/her to compute previous session keys.

6) Anonymity and untraceability

Generally, the mobile users intend to access cloud services in an anonymous way. As defined in the threat model in Section 6.2, $\mathcal{A}$ is able to eavesdrop public messages transmitted between MU_i and CS_j. During login time, MU_i sends $Msg_1 = \{TID_i^*,\, C_1,\, H_1,\, TS_i\}$ to CS_j. Note that MU_i does not send the original identity to login message. Rather, it sends the temporary identity TID_i embedded in $TID_i^* = TID_i \oplus H(RID_{S_j}||TS_i)$ (see Remark 6.2). Moreover, from $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ and $H_1 = H(ID_i\, ||C_1\, ||RN_i\, ||TS_i)$, it is not possible to obtain ID_i. During authentication phase, CS_j sends $Msg_2 = \{C_2,\, H_3,\, TS_j\}$ to MU_i, where $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$ and $H_3 = H(ID_i\, ||M_1\, ||RN_j\, ||TS_i\, ||TS_j\, ||SK_{CS_j,MU_i})$. As hash function $H(\cdot)$ is considered to be collision resistant, from these eavesdropped messages, it is computationally infeasible for an attacker to compute ID_i. Thus, the proposed scheme provides user anonymity property.

The messages $Msg_1 = \{TID_i^*,\, C_1,\, H_1,\, TS_i\}$ and $Msg_2 = \{C_2,\, H_3,\, TS_j\}$ are unique and dynamic in nature in each session, because each component in these messages use either timestamp or random nonce. Hence, the proposed scheme also preserves the untraceability property as an attacker can not trace the same user in different session.

7) Session key security

According to the proposed scheme, both MU_i and CS_j mutually establish a common session key SK_{MU_i,CS_j} $(= SK_{CS_j,MU_i})$ for future communication. Note that the session key is

calculated as

$$\begin{aligned} SK_{MU_i,CS_j} &= H(ID_i||ID_{S_j}||A_{ij}||RN_i||M_2||TS_i||TS_j) \\ &= H(ID_i||ID_{S_j}||B_{ji}||RN_i||M_2||TS_i||TS_j) \\ &= H(ID_i||ID_{S_j}||B_{ji}||M_1||M_2||TS_i||TS_j) \\ &= H(ID_i||ID_{S_j}||B_{ji}||M_1||RN_j||TS_i||TS_j) \\ &= SK_{CS_j,MU_i}. \end{aligned}$$

To establish the session key, both MU_i and CS_j mutually authenticate each other. Furthermore, to derive the session key an attacker needs to have the credentials ID_i, ID_{S_j}, and A_{ij} $(= B_{ji})$. Hence, the session key is secure.

8) Parallel session and reflection attacks

An adversary $\mathcal{A}$ can masquerade as a genuine user and then try to initiate a new parallel session with CS_j if the credentials belonging to a legal user are obtained. On the other side, $\mathcal{A}$ can not obtain mobile user credentials through offline guessing attack or with any eavesdropped messages. As a result, the proposed scheme resists parallel session as well as reflection attacks.

9) Ephemeral secret leakage attack

Under this attack, the exposer of ephemeral (temporary) secrets (e.g., random numbers) of a session may harm the secrecy of a session key. After execution of the protocol, if the random numbers are not properly deleted, an adversary $\mathcal{A}$ might obtain them from a compromised device and also can launch ephemeral secret leakage attack. An authentication protocol must be able to resist this attack.

In the proposed scheme, the session key is computed as $SK_{MU_i,CS_j} = H(ID_i\ ||ID_{S_j}\ ||A_{ij}\ ||RN_i\ ||M_2\ ||TS_i\ ||TS_j)$, where $A_{ij} = H(H(ID_i \oplus r_{ij})\ ||X_j)$, r_{ij} is an 1024-bit random secret and X_j is the master secret key of the cloud server CS_j. Hence, even if the values of random numbers RN_i and RN_j (that is, M_2) are known, the adversary $\mathcal{A}$ cannot compute SK_{MU_i,CS_j} as it also depends on the long-term secret credentials, such as ID_i, ID_{S_j} and A_{ij}. As a result, SK_{MU_i,CS_j} can not derive other session keys established in other sessions between MU_i and CS_j using the ephemeral secret leakage attack.

10) User impersonation attack

Using user impersonation attack, an adversary $\mathcal{A}$ can masquerade as a legitimate user and try to login to CS_j. However, the proposed scheme can resist this attack due to the following argument. $\mathcal{A}$ needs to input correct inputs ID_i, PW_i and $\mathcal{B}_i'$ to prove its authenticity as a genuine user. As already discussed, an adversary has no computationally feasible way to guess these parameters.

$\mathcal{A}$ can also try to generate a replay login message $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and submit it to CS_j. But as explained in the replay attack protection, a duplicate value of the timestamp TS_i or random number RN_i will reveal that the message is a replayed one and it is not an original message. Note that $C_1 = A_{ij} \oplus RN_i \oplus TS_i \oplus H(ID_{S_j})$ and $A_{ij} = V_{ij}' \oplus RPBW_i$. As $\mathcal{A}$ does not know ID_i, PW_i and $\mathcal{B}_i'$, he/she is unable to guess correct value of A_{ij} and can not modify Msg_1. Hence, the proposed scheme resists user impersonation attack.

11) Server impersonation attack

The proposed scheme protects server impersonation attack where an adversary $\mathcal{A}$ can masquerade as a cloud server and try to respond with valid message to MU_i. When CS_j receives the user login message, it replies with an authorization message $Msg_2 = \{C_2, H_3, TS_j\}$. This message contains the hash value $H_3 = H(ID_i \,||M_1\, ||RN_j\, ||TS_i\, ||TS_j\, ||SK_{CS_j,MU_i})$. Moreover, Msg_2 also contains $C_2 = B_{ji} \oplus RN_j \oplus TS_j \oplus ID_i$. $\mathcal{A}$ can not obtain $B_{ji} = A_{ij} = H(H(ID_i \oplus r_{ij}) \,||X_j)$ as it requires the server secret key X_j and random number r_{ij}. As a consequence, the proposed scheme also resists server impersonation attack.

12) Privileged-insider attack

In this attack, we assume that the registration information $\{ID_i, (RPWB_i \oplus k)\}$ from the mobile user registration request message is known to a privileged-insider user of the RC, who acts an adversary $\mathcal{A}$. Later, after completing the mobile user registration process, it is also assumed that $\mathcal{A}$ also attains the stolen/lost mobile device, and then extract the information stored in the device using the power analysis attack [142]. It is computationally difficult task for $\mathcal{A}$ to obtain PW_i and the biometric key θ_i from V_{ij}' and A_{ij}' in polynomial time. Furthermore, without having the random secret k, $\mathcal{A}$ can not compute $RPWB_i$ from $RPWB_i \oplus k$. Therefore, $\mathcal{A}$ can not also obtain PW_i and θ_i from $RPWB_i$. Hence, the proposed scheme is resilient against privileged-insider attack.

6.6 Formal security verification using ProVerif tool

The formal security verification of the proposed scheme is presented in this section using the applied pi calculus based ProVerif simulation tool [12], [13]. This tool can be practically used for testing whether an attacker is able to attack (or compromise) the session key in a security protocol.

In Figure 6.7, we provide the code for declaration of channels, free variables, constants, functions, equations, queries and events required for the proposed scheme. The code for the process of the mobile user in the registration, login and authentication phases is modeled in Figure 6.8. The process of the cloud sever CS_j is modeled as parallel composition of the process of registration (SReg) and process of authentication (SAuth). Figure 6.9 shows the program code for the processes related to CS_j.

Finally, we execute the codes of the previous three tables in ProVerif latest version (i.e., ProVerif 1.93). The complete obtained results of session key secrecy (from both user and server side) and authentication are shown in Figure 6.10. The result shows the following observatios:

- RESULT inj-event(UserAuth(id)) ==> inj-event (UserStart(id)) is true.
- RESULT not attacker(SKus[]) is true.
- RESULT not attacker(SKus[]) is true.

Hence, the proposed scheme passes the security verification.

6.7 Performance comparison

In this section, we compare the security and functionality of the proposed scheme with the recently developed multi-server authentication schemes designed for mobile cloud computing services [88], [183], [194], [196], [230].

6.7.1 Security and functionality comparison

A detailed comparison on different security attacks is shown in Table 6.5. It is seen that a large number of the recent schemes suffer from denial of service attack and stolen mobile device attack. Further, most of the existing schemes fail to provide efficiency in login phase

```
(* ———— channels ————*)
free pch: channel. (* public channel *)
free sch: channel [private]. (* private channel *)
(* ———— shared keys ———— *)
free SKus:bitstring [private].(* the session key of user *)
free SKsu:bitstring [private]. (* the session key of server *)
(* ———— Servers secret key ———— *)
free Xj:bitstring [private].
free rij:bitstring [private].
(* ———— constants ———— *)
free IDSj:bitstring [private].
free ID:bitstring [private].
free TID:bitstring [private].
free PW:bitstring [private].
const Bi:bitstring [private].
(* ———— functions and equations ———— *)
fun h(bitstring):bitstring. (* hash function *)
fun FE(bitstring):bitstring. (* Fuzzy extractor function *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring. (* string concatenation *)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(* ———— aims for verification ———— *)
query attacker(SKus).
query attacker(SKsu).
query id:bitstring; inj-event(UserAuth(id)) ==> inj-event(UserStart(id)).
(* ———— event ———— *)
event UserStart(bitstring). (* User starts authentication *)
event UserAuth(bitstring). (* User is authenticated *)
```

Figure 6.7: Declaration of channels, keys, constants, functions, equations, queries and events.

```
let User=
new b:bitstring;
let alpha = FE(Bi) in
let RPWB = h(con(ID,h(con(PW,con(alpha,b))))) in
out(sch,(ID,RPWB));
in(sch,(rVij:bitstring));
let D1 = xor(h(con(PW,alpha)),b) in
let D2 = h(con(ID,con(PW,con(alpha,b)))) in
!
(
event UserStart(ID);
let b1 = xor(D1,h(con(PW,alpha))) in
let D21 = h(con(ID,con(PW,con(alpha,b1)))) in
if D2 = D21 then
new RNi:bitstring;
new TSi:bitstring;
let Aij1 = xor(rVij,RPWB) in
let C1 = xor(Aij1,xor(RNi,xor(TSi,h(IDSj)))) in
let H1 = h(con(ID,con(C1,con(RNi,TSi)))) in
out(pch,(TID,C1,TSi,H1));
in(pch,(rC2:bitstring,rTSj:bitstring,rH3:bitstring));
let M2 = xor(rC2,xor(rTSj,xor(ID,Aij1))) in
let SKus = h(con(ID,con(IDSj,con(Aij1,con(RNi,con(M2,con(TSi,rTSj))))))) in
let H4 = h(con(ID,con(RNi,con(M2,con(TSi,con(rTSj,SKus)))))) in
if H4 = rH3 then
0
).
```

Figure 6.8: ProVerif code for the process of mobile user MU_i.

and password change phase, and they do not provide revocation of lost mobile device phase. It is clear from Table 6.5 that the proposed scheme overcomes such security and functionality weaknesses of the existing schemes.

```
let SReg =
in(sch,(sID:bitstring,sRPWB:bitstring));
let Aij = h(con(h(xor(sID,rij)),Xj)) in
let Vij = xor(Aij,sRPWB) in
out(sch,(Vij)).
```

```
let SAuth =
in(pch,(xID:bitstring,xC1:bitstring,xTSi:bitstring,xH1:bitstring));
let Bji = h(con(h(xor(xID,rij)),Xj)) in
let M1 = xor(Bji,xor(xC1,xor(xTSi,h(IDSj)))) in
let H2 = h(con(xID,con(xC1,con(M1,xTSi)))) in
if H2 = xH1 then
event UserAuth(xID);
new RNj:bitstring;
new TSj:bitstring;
let C2 = xor(Bji,xor(RNj,xor(TSj,xID))) in
let SKsu = h(con(xID,con(IDSj,con(Bji,con(M1,con(RNj,con(xTSi,TSj))))))) in
let H3 = h(con(xID,con(M1,con(RNj,con(xTSi,con(TSj,SKsu)))))) in
out(pch,(C2,TSj,H3)).
let S = SReg — SAuth.
process !User — !S
```

Figure 6.9: ProVerif code for the process of the cloud server CS_j.

6.7.2 Computational costs comparison

As implemented by Scott *et al.* [179] and Tseng *et al.* [196], we have considered Philips HiPersmart card device and Pentium IV computer for user side and cloud server side computation, respectively. Philips HiPersmart card has a clock speed of 36MHz with 32-bit RISC MIPS processor. It has flash memory of 256 KB with 16KB RAM. On the other side, Pentium IV has maximum clock speed of 3GHz operating under Windows XP OS with 512 MB RAM [112]. Bilinear pairing and other cryptographic operations are implemented in C language under specific IDE and specific C/C++ Library (MIRACL). Table 6.6 shows the notations for different cryptographic operations along with their execution time in Philips HiPersmart card device and Pentium IV computer, respectively.

In Table 6.7, we tabulate and compare the computation overhead of the proposed scheme

```
File "./tmpfiles/40438219/inpProt.pv", line 54, character 5 - line 54, character 9:
Warning: identifier SKus rebound
File "./tmpfiles/40438219/inpProt.pv", line 75, character 5 - line 75, character 9:
Completing equations...
Completing equations...
– Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
Completing...
200 rules inserted. The rule base contains 200 rules. 28 rules in the queue.
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.
– Query not attacker(SKsu[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKsu[])
RESULT not attacker(SKsu[]) is true.
– Query not attacker(SKus[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKus[])
RESULT not attacker(SKus[]) is true.
```

Figure 6.10: Analysis of the simulation results.

with the relevant schemes [88], [183], [194], [196], [230]. For all the given schemes, we separately tabulated computation for MU_i and cloud service provider CS_j under Philips HiPersmart card device and Pentium IV computer, respectively. Also, we mention the underlying cryptographic operations for each relevant scheme in comparison. We study that the total user side computation overhead of the proposed scheme in login and authentication phases is $9*T_H+8*T_X+T_{FE}$. Considering the execution time needed for XOR operation is negligible, the total execution time of MU_i is then approximately $(9*1+130) = 139$ ms. On the other hand, the cloud service provider CS_j has a computation overhead of $7*T_H+7*T_X$. Hence, total execution time in Pentium IV server is less than $7*0.01 = 0.07$ ms.

Table 6.5: Security and functionality comparison with the recent authentication schemes.

Security attributes	He-Wang [88]	Yoon-Yoo [230]	Shen *et al.* [183]	Tsai-Lo [194]	Tseng *et al.* [196]	Our
Stolen mobile device/smart card attack	✓	X	✓	✓	NA	✓
Strong replay attack	X	✓	✓	✓	✓	✓
Password guessing attack (online)	✓	✓	✓	✓	✓	✓
Password guessing attack (offline)	✓	✓	✓	✓	✓	✓
Privileged insider attack	✓	X	✓	✓	✓	✓
DoS attack	X	✓	✓	✓	✓	✓
Known session key secrecy	✓	✓	✓	✓	✓	✓
Strong user anonymity provision	X	X	X	✓	X	✓
Forward secrecy	✓	✓	✓	✓	✓	✓
Session key security	X	X	X	X	X	✓
User impersonation attack	X	X	✓	X	✓	✓
Server impersonation attack	✓	✓	✓	X	✓	✓
Ephemeral secret key leakage attack	X	X	X	X	✓	✓
User anonymity provision	✓	✓	✓	✓	X	✓
Efficient password change	X	✓	✓	NA	NA	✓
Login phase efficiency	X	✓	✓	X	X	✓
Revocation of smart card	NA	✓	✓	NA	NA	✓
Secure mutual authentication	✓	✓	X	X	✓	✓
Low computation overhead	X	X	X	X	✓	✓
Low communication overhead	X	✓	X	X	✓	✓
Formal security proof	X	X	✓	✓	✓	✓
Simulation using AVISPA/ProVerif	X	X	X	X	X	✓

✓: the scheme is secure or it supports a feature; X: the scheme is not secure or it does not support the feature; **NA:** the security or functionality feature cannot be applicable for that respective scheme.

6.7.3 Communication costs comparison

Comparison on communication costs of the proposed scheme with related mobile user authentication schemes [88], [183], [194], [196], [230] is also tabulated in Table 6.8. Since the user and server registration phases, password change phase and lost mobile device revocation phase are executed only once, we consider only login and authentication phases for calculation of communication cost for the proposed scheme and other schemes. The proposed scheme needs two messages $Msg_1 = \{TID_i^*, C_1, H_1, TS_i\}$ and $Msg_2 = \{C_2, H_3, TS_j\}$, which require

Table 6.6: Actual execution time of different operations.

Symbol	Description	Execution Time (in ms)	
		HiPerSmart card (MU_i)	Pentium IV (CS_j)
T_P	Bilinear pairing operation	380	3.16
T_M	Elliptive curve point multiplication	130	1.17
T_{FE}	Fuzzy extractor operation	$\approx T_M$	$\approx T_M$
T_{sym}	Symmetric encryption/decryption	< 17.93	< 0.16
T_{G_H}	map-to-point hash function	< 100	< 1
T_A	elliptic curve point addition	< 10	< 0.1
T_H	One way hash function	< 1	< 0.01
T_X	Bitwise XOR function	negligible	negligible

Table 6.7: Comparison of computational costs among related schemes.

Entity	He-Wang [88]	Yoon-Yoo [230]	Shen *et al.* [183]	Tsai-Lo [194]	Tseng *et al.* [196]	Our
MU_i	$7T_H + 3T_M$ $\approx$ 397 ms	$5T_H + 2T_M$ $\approx$ 265 ms	$5T_H + 3T_M$ $\approx$ 395 ms	$3T_H + 3T_M$ $\approx$ 393 ms	$3T_H + T_M$ $\approx$ 133 ms	$9T_H + 8T_X + T_{FE}$ $\approx$ 139 ms
CS_j	$5T_H + 2T_M$ $\approx$ 2.39 ms	$5T_H + 2T_M$ $\approx$ 2.39 ms	$5T_H + 2T_M$ $\approx$ 2.39 ms	$2T_H + 4T_M$ $2T_P$ $\approx 11.02ms$	$3T_H + 2T_M +$ $T_P + 2T_{G_H} + 2T_A$ $\approx 7.63ms$	$7T_H + 7T_X$ $\approx$ 0.07 ms
RC	$9T_H + 2T_M$	$5T_H$	$7T_H + T_M$	–	–	–
CP	ECC	ECC	ECC	Pairing	Pairing	Hash

CP: Cryptographic primitive used in the respective scheme.

$(160+160+32+160) = 512$ bits and $(160+32+160) = 352$ bits, respectively. So, the overall communication cost of the proposed scheme is $(512+352) = 864$ bits. Note that the proposed scheme does not involve RC during login and authentication phases, which causes significant reduction of overall communication cost. In addition, the proposed scheme requires only two rounds of message communication, whereas other related schemes require five, four or three

Table 6.8: Comparison of communication costs.

Scheme	No. of rounds	No. of bits
He-Wang [88]	5	3520
Yoon-Yoo [230]	5	2496
Shen *et al.* [183]	5	1856
Tsai-Lo [194]	4	1696
Tseng *et al.* [196]	3	992
Our	2	864

rounds of message communication. It is observed from Table 6.7 that the user mobile device in He-Wang's scheme [88] and Yoon-Yoo's scheme [230] takes approximately 397 ms and 265 ms, respectively. Shen *et al.*'s scheme [183] and Tsai-Lo's scheme [194] take approximately 395 ms and 393 ms, respectively. Reason behind the high computation cost in the existing schemes is that they either use ECC based cryptosystem or bilinear pairing based cryptosystem. Quite clearly the user side computation cost of the proposed scheme is much less than that for these schemes. As a result, the proposed scheme comparatively more suited for the mobile users with low-power computing devices.

6.8 Summary

In this chapter, we proposed a mobile user authentication scheme on mobile cloud computing environment, which is designed through on one-way hash function, bitwise XOR operation and fuzzy extractor functions only. We provided the formal security proof through the ROR model and the formal security verification through the ProVerif 1.93 simulation tool. Moreover, authentication proof of the proposed scheme is provided by BAN logic. As the proposed scheme avoids any computation expensive cryptographic operations, it has the lowest computation cost as compared to the existing related schemes. Further, the proposed scheme does not involve the *RC* in the authentication process. Hence, it has very low communication cost. Overall, high security and efficiency make the proposed scheme very suitable for the practical applications in the mobile cloud computing domain.

Chapter 7

Fine-Gained Access Control with User Authentication for Telecare Medicine Information Systems

Telecare medicine information system (TMIS) for health-care delivery service requires information exchange among multiple 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 an efficient user access control mechanism. 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. 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.

In this chapter, we present a new fine-grained access control scheme with user authentication for TMIS. The proposed scheme provides group-based user authentication depending on the access rights provided for the genuine users. The proposed scheme supports user anonymity, forward secrecy, and efficient password change without contacting the remote server.

7.1 Research contributions

The following contributions made in this chapter are listed below:

- We introduce the concept of fine-grained data access control of server data with suitable authentication scheme in TMIS. We divide all users into several groups based on the access types.
- We used the Key-Policy Attribute-Based Encryption (KP-ABE) [77] in order to achieve the fine-grained data access control with full granularity for accessing right data by a right user.
- 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.

7.2 Adversary model

We follow the widely accepted security assumptions about the smart card security and capacity of adversary in three factor authentication schemes [33], [46], [70].

- Expect registation process, all message communications are done through a public channel. The adversary can control the channel with an ability to intercept, delete, modify, resend and reroute the transmitted messages [70].

- An authorized legitimate user can perform malicious activity and turn into an adversasry.
- An adversary has capability to execute power analysis attacks in order to extract the stored information from a user's smart card.
- Our three factor authentication scheme does not consider the threats that arise from the biometric factors [163].
- The user-chosen passwords are assumed to be drawn from a Zipf-distributed [202] dictionary $\mathcal{D}$ of small size $|\mathcal{D}|$, where $|\mathcal{D}|$ is a fixed constant which is independent of the system security parameter [202]. The passwords cannot assumed to be uniformly distributed.

7.3 The proposed fine-grained access control scheme

In this section, we describe how the proposed scheme exploits attribute based access control to achieve fine-grained data access control in TMIS. First, we tabulate the important notations that are useful to explain and analyze the proposed scheme. Next, we describe the proposed scheme in detail.

We use the notations listed in Table 7.1. We use the secure hash standard (SHA-1) [6] as one-way cryptographic hash function. For symmetric key encryption/decryption, we apply the Advanced Encryption Standard (AES-128) [2] in our proposed scheme. Note that for better security, one can also consider SHA-256 as one-way cryptographic hash function [6], [61].

The proposed scheme consists of five phases: 1) setup, 2) registration, 3) login, 4) authorization, and 5) password change phase. 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.

- **Setup phase:** This phase is used to pre-load keying materials to the medical application server (MAS) and user smart card prior to start working. For this purpose, the MAS chooses a set of *network parameters*.
- **Registration phase:** In the registration phase, a user U_j needs to register with the MAS in offline for accessing medical data. This phase has two sub-phases, namely 1) *access structure generation*, where the MAS selects an access structure P_j for each user U_j and 2) *smart card generation*, where the MAS generates a smart card with valid identity SC_{id_j} for U_j.

Table 7.1: Notations used in the proposed scheme.

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
G	Multiplicative cyclic group of prime order p
$\|G\|$	Order of group G
$H(\cdot)$	Secure one-way hash function
MK_s	Master key of server MAS
$A\|\|B$	Data A concatenates with data B
$E_K(\cdot)$	Symmetric key encryption using the key K
$D_K(\cdot)$	Symmetric key decryption using the key K
T_{U_j}, T_s	Current timestamps of U_j and MAS, respectively
$Gen(\cdot)$	Fuzzy extractor probabilistic generation function
$Rep(\cdot)$	Fuzzy extractor deterministic reproduction function
$\triangle T$	Maximum transmission delay

- **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.
- **Authorization phase:** In this phase, a mutual authentication between U_j and MAS takes place. At the end of this phase, both U_j and MAS establish a session key for their future secure communication.
- **Password change phase:** In this phase, any legal user U_j can change his/her password freely and completely locally without the help of the MAS.

7.3.1 Setup phase

In this phase, the MAS executes the following steps:

- **Step S1:** The 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 S2:** The 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 S3:** The 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 S4:** Finally, the MAS assigns a unique randomly generated master key, say MK_S for its own. In addition, the MAS selects a one-way cryptographic hash function $H(\cdot)$ (for example, SHA-1 [6]).

7.3.2 Registration phase

This phase consists of the following steps:

- **Step R1:** 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 R2:** 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 R3:** 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 R4:** 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, 1) *access structure generation* and, 2) *smart card generation*, which are discussed below.

1) 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.

For example, consider a scenario as explained in [43]. 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 7.1, 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 random polynomial q_x of degree $d_x - 1$ using the Lagrange interpolation [91] 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.

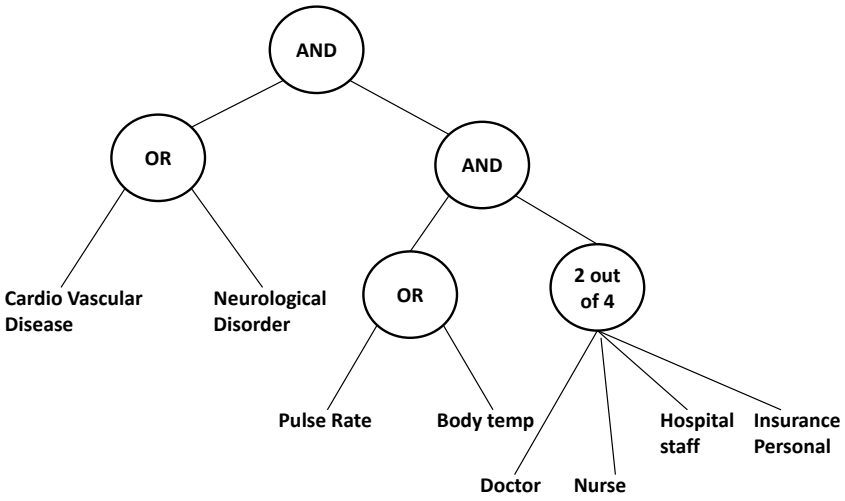


Figure 7.1: User access structure.

2) 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 and $H(\cdot)$. 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 7.2.

7.3.3 Login phase

U_j makes login into MAS using the following steps:

- **Step L1:** 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 L2:** Using the fuzzy extractor reproduction procedure $Rep(\cdot)$ and stored β_j, SC_j computes $\alpha_j = Rep(B'_j, \beta_j)$, 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

User (U_j)	Medical Application Server (MAS)
REGISTRATION PHASE	
Select ID_{U_j}, PW_j, B_j and r_j.	
Compute $(\alpha_j, \beta_j) = Gen(B_j)$,	
masked password $W_j = H(\alpha_j \| PW_j)$,	
and $A_{ID_j} = H(\alpha_j \| ID_{U_j} \| r_j)$.	Store A_{ID_j}, GID_j and server id S_{ID}.
Choose access group id GID_j.	Calculate $A_j = H(A_{ID_j} \| TS_{U_j})$,
$\xrightarrow[\textbf{(secure channel)}]{\langle A_{ID_j}, W_j, \alpha_j, GID_j \rangle}$	$R_{U_j} = H(W_j \| A_j \| GID_j)$,
	$X_j = H(\alpha_j \| S_{ID_j}) \oplus H(MK_S \| A_j)$.
	Generate access structure P_j for each U_j.
	Compute key $SK_j = \langle \{D_i = g^{\frac{q_i(0)}{t_i}}\}_{i \in \mathcal{L}} \rangle$.
Store β_j, $Gen(\cdot)$, $Rep(\cdot)$, τ in smart card.	$\xleftarrow[\textbf{(secure channel)}]{\textbf{Smartcard}=\langle A_{ID_j} \oplus SC_{id_j}, GID_j, R_{U_j}, r_j \oplus \alpha_j, TS_{U_j} \oplus SC_{id_j}, P_j, SK_j, e: G_1 \times G_2 \to G_T, X_j, H(\cdot) \rangle}$
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,	Delete R_{U_j} from records.
$H(\cdot)$, β_j, $Gen(\cdot)$, $Rep(\cdot)$, $\tau\}$.	Store $\{A_j, MK_s, TS_{U_j}\}$ in its database.

User (U_j)	Medical Application Server (MAS)
LOGIN PHASE	
Input PW_j, identity ID_j & imprint B'_j.	
Compute $\alpha_j = Rep(B'_j, \beta_j)$, $W'_j = H(\alpha_j \| PW_j)$,	
$A'_{ID_j} = H(\alpha_j \| ID_{U_j} \| r_j)$,	
$SC'_{id_j} = (A_{ID_j} \oplus SC_{id_j}) \oplus A'_{ID_j}$,	
$TS'_{U_j} = (TS_{U_j} \oplus SC_{id_j}) \oplus SC'_{id_j}$,	
$A'_j = H(A_{ID_j} \| TS'_{U_j})$, $R'_{U_j} = H(W'_j \| A'_j \| GID_j)$.	
If $R'_{U_j} \stackrel{?}{=} R'_{U_j}$ holds, $S_j = X_j \oplus H(\alpha_j \| S_{ID})$.	
Select information type IT_i.	
Compute $N_j = H(S_j \| H(n_j \| GID_j) \| T_{U_j} \| IT_i)$.	
$\xrightarrow[\textbf{(public channel)}]{\langle N_j, IT_i, T_{U_j}, (TS'_{U_j} \oplus H(n_j \| GID_j)) \rangle}$	

Figure 7.2: User registration and login phases of the proposed scheme.

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 L3:** 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 L4:** 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.

This login phase is further summarized in Figure 7.2.

7.3.4 Authorization phase

This phase involves the following steps:

- **Step A1:** 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 A2:** 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 A3:** 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 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 A4:** 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 A5:** After receiving the message from the MAS in Step **A4**, U_j first checks if $|T_s - T_s^*| < \triangle T$ 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 A6:** 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 A7:** 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 A8:** 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.

This authorization phase is summarized in Figure 7.3.

7.3.5 Password change phase

This phase contains the following steps, and is also summarized in Figure 7.4:

- 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.

User (U_j)	Medical Application Server (MAS)
	Check $\lvert T_{U_j} - T^*_{U_j}\rvert \stackrel{?}{\leq} \triangle T$.
	Compute $S'_j = H(MK_S \lvert\rvert A_j)$,
	$H(n_j \lvert\rvert GID_j) = TS_{U_j} \oplus (TS_{U_j} \oplus H(n_j \lvert\rvert GID_j)$.
	Check $H(S'_j \lvert\rvert H(n_j \lvert\rvert GID_j) \lvert\rvert T_{U_j} \lvert\rvert IT_i) \stackrel{?}{=} N_j$.
	Check if $IT_i \in GID_j$?
	Select access key K_s and α.
	Compute $E_j = \{\{T_i^{\alpha}\}_{\forall i \in IT_i}\}_{\forall IT_i \in GID_j}$.
	Compute $E' = K_s Y^{\alpha}$.
Check $\lvert T_s - T^*_s\rvert \stackrel{?}{\leq} \triangle T$.	Generate and encrypt PM as $E_{K_s}(PM)$.
Compute $K'_j = H(W_j \lvert\rvert S_{ID}) \oplus$	Compute $H(A_j \lvert\rvert PM \lvert\rvert E' \lvert\rvert T_s \lvert\rvert T_{U_j})$,
$\mathrm{H}(\mathrm{H}(\mathrm{n}_j \lvert\rvert GID_j) \lvert\rvert RN_s)$.	$K_j = (X_j \oplus S'_j) \oplus H(H(n_j \lvert\rvert GID_j) \lvert\rvert RN_s)$
Check $K'_j \stackrel{?}{=} K_j$.	$\xleftarrow[\textbf{(public channel)}]{\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H(RN_s \lvert\rvert PM \lvert\rvert E' \lvert\rvert T_s \lvert\rvert T_{U_j}), E_j\rangle}$
Using access tree, obtain access key K_s.	
Compute $PM^* = D_{K_s}(E_{K_s}(PM))$.	
Check if $H(RN_s \lvert\rvert PM^* \lvert\rvert E' \lvert\rvert T_s \lvert\rvert T_{U_j})$ $\stackrel{?}{=}$ received hash value.	
If verification holds, compute $PM' = H(PM^* \lvert\rvert RN_j \lvert\rvert T_i)$.	
Establish session key $SK_{U_j,S}$ as $SK_{U_j,S} = H(PM \lvert\rvert\ RN_j \lvert\rvert\ RN_s\ \lvert\rvert K_s\ \lvert\rvert$ $T_{U_j}\ \lvert\rvert T_s\ \lvert\rvert A_j)$.	
$\xrightarrow[\textbf{(public channel)}]{\langle H(PM' \lvert\rvert RN_s \lvert\rvert M_j), M_j, E_{K_s}(PM'), RN_j\rangle}$	
	Decrypt PM as $PM' = D_{K_s}(E_{K_s}(PM'))$.
	Compute $PM'' = H(PM \lvert\rvert RN_j \lvert\rvert T_i)$.
	Check $PM'' \stackrel{?}{=} PM'$.
	If verification holds, access is granted to U_j.
	Establish session key SK_{S,U_j} as $SK_{S,U_j} = H(PM\ \lvert\rvert RN_j\ \lvert\rvert RN_s\ \lvert\rvert K_s\ \lvert\rvert T_{U_j}\ \lvert\rvert T_s \lvert\rvert A_j)$.
Store session key $SK_{U_j,S}$ $(= SK_{S,U_j})$.	Store session key SK_{S,U_j} $(= SK_{U_j,S})$.

Figure 7.3: Authorization phase of the proposed scheme.

User (U_j)	Smart card (SC_j)
Insert smart card SC_j into card reader. Input 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 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 verification holds, enter new password PW_j^{new}.
Input new changed password PW_j^{new}.	
	Calculate new $W_j^{new} = H(\alpha_j \,||PW_j^{new})$, $R_{U_j}^{new} = H(W_j^{new} \,||A'_j \,||GID_j)$. Replace R_{U_j} with $R_{U_j}^{new}$ in its memory.

Figure 7.4: Password change phase of the proposed scheme.

7.4 Security analysis

In this section, we provide both formal security and informal security analysis of the proposed scheme. In Section 7.4.1, formal security analysis is done using ROR Model. In Section 7.4.2, we provide formal authentication proof using BAN logic. In Section 7.4.3, through informal security analysis, we discuss on how the proposed scheme resists various other security threats and attacks.

7.4.1 Formal security analysis using ROR model

We present the formal security analysis of the proposed fine-grained access control scheme through the widely-used ROR model [14]. 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 the proposed scheme, we consider all possible 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.

We also define the following definitions [26], [223] prior to proving Theorem 7.1.

Definition 7.1. *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 7.2. *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 7.3 (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 7.4 (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*

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

Definition 7.5. *The proposed fine-grained access control with user authentication scheme (FGUA) 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" [158].*

Theorem 7.1. *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 7.5.*

Proof. 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,

$$Adv_{\mathcal{A}}^{FGUA} = |2Pr[Succ_0] - 1|. \tag{7.1}$$

- **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 7.2. *Send* query is simulated in Table 7.3. 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

$$Pr[Succ_1] = Pr[Succ_0]. \tag{7.2}$$

- **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

Table 7.2: 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 $H_1 = H(n_j \|\| GID_j)$ and $N_j = H(S_j \|\| H(n_j \|\| GID_j) \|\| T_{U_j} \|\| IT_i)$. 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 $H_2 = H(RN_s \|\| PM \|\| E' \|\| T_s \|\| T_{U_j})$. 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 $H_3 = H(PM' \|\| RN_s \|\| M_j)$. U_j sends message Msg_3 to MAS, where $Msg_3 = \{H_3, M_j, E_{K_s}(PM'), RN_j\}$. Note that $\langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle \leftarrow Send(U_j, \mathbf{start})$, $\langle K_j, RN_s, IT_i, E', E_{K_s}(PM), T_s, H_2, E_j\rangle \leftarrow Send(S, \langle N_j, IT_i, T_{U_j}, TS_{U_j} \oplus H_1\rangle)$ 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.

phases of the proposed 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

Table 7.3: 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 7.2.
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 7.3.
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 7.3 and Table 7.2.
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 7.3. 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 7.3 and Table 7.2.
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 7.3.

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

probability of random numbers collision is at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. So, we have,

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}}. \tag{7.3}$$

- **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 autho-

rization phases of the proposed 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 $\frac{q_s}{2^{l_r}}$.
- **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,

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{2q_s}{2^{l_r}} + \frac{q_s}{2^{l_n}} + \frac{12q_h}{2^{l_h}}. \tag{7.4}$$

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

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 cases:

- **Case 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 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, $\varepsilon_{bm} \approx 2^{-14}$ [158]. 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,

$$Pr[Succ_4] = \frac{1}{2}. \tag{7.5}$$

Using the triangular inequality, we have,

$$\begin{aligned} |Pr[Succ_0] - \frac{1}{2}| &= |Pr[Succ_1] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| + |Pr[Succ_2] - Pr[Succ_4]| \\ &\leq |Pr[Succ_1] - Pr[Succ_2]| \\ &\quad + |Pr[Succ_2] - Pr[Succ_3]| + |Pr[Succ_3] - Pr[Succ_4]|. \end{aligned} \tag{7.6}$$

Using Equations (7.1)-(7.6), we obtain,

$$\begin{aligned} \frac{1}{2}Adv_{\mathcal{A}}^{FGUA} &= |Pr[Succ_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}} + \frac{2q_s}{2^{l_r}} + \frac{2q_s}{2^{l_n}} + \frac{12q_h}{2^{l_h}} \\ &\quad + \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned} \tag{7.7}$$

Finally, multiplying both sides by 2 in Equation (7.7) and rearranging the terms, we obtain the required result. Hence, the theorem is proved. □

7.4.2 Mutual authentication proof based on BAN-logic

The BAN logic is used in analyzing the security of authentication schemes 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. Basic BAN logic notations and logical postulates are provided in Section 2.6.

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.** $U_j \rightarrow MAS$: $(H(S_j \mid\mid H(n_j \mid\mid GID_j) \mid\mid T_{U_j} \mid\mid IT_i) \mid\mid IT_i \mid\mid T_{U_j} \mid\mid (TS'_{U_j} \oplus H(n_j \mid\mid GID_j)))$.
- **Message 2.** $MAS \rightarrow U_j$: $(X_j \oplus S'_j) \oplus H(H(n_j \mid\mid GID_j) \mid\mid RN_s) \mid\mid RN_s \mid\mid IT_i \mid\mid E' \mid\mid E_{K_s}(PM) \mid\mid T_s \mid\mid H(RN_s \mid\mid PM \mid\mid E' \mid\mid T_s \mid\mid T_{U_j} \mid\mid K_j) \mid\mid E_j)$.
- **Message 3.** $U_j \rightarrow MAS$: $(H(PM' \mid\mid RN_s \mid\mid M_j) \mid\mid M_j \mid\mid E_{K_s}(PM') \mid\mid RN_j \mid\mid T^1_{U_j})$.

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: $MAS \lhd (\langle R_j, T_{U_j}, IT_i\rangle_{S_j}, IT_i, T_{U_j}, \langle R_j\rangle_{TS_{U_j}})$.
- S_2: According to the inference rule (Rule 5), we obtain $MAS \lhd \langle R_j, T_{U_j}, IT_i\rangle_{S_j}$.
- 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}: $U_j \lhd (\langle X_j, \ R_j, \ RN_s \rangle_{S_j}, \ RN_s, \ IT_i, \ E', \langle PM \rangle_{K_s}, \ \langle RN_s, \ PM, E', \ T_s, \ T_{U_j}) \rangle_{S_j}, \ E_j)$.
- 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.

7.4.3 Discussion on other attacks

In this section, through the informal security analysis we show that the proposed 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 [119], [142] 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, the proposed 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 the proposed 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

By executing the power analysis attacks [119], [142], $\mathcal{A}$ 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, the proposed 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 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 the proposed 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, $\mathcal{A}$ 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, the proposed scheme protects the parallel session and reflection attacks.

7.5 Functionality analysis

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

7.5.1 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 used the Key-Policy Attribute-Based Encryption (KP-ABE) [77] 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.

7.5.2 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 $\mathcal{A}$ 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.

7.5.3 Mutual authentication

In authorization phase of the proposed 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.

7.5.4 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.

7.5.5 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.

7.5.6 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.

7.6 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.

7.6.1 Security comparison

We compare security of the proposed scheme with existing related authentication schemes for TMIS [24], [108], [146]. A detailed comparison on different security attacks are tabulated in Table 7.4. It is clear from this table that our proposed scheme overcomes most of the security weaknesses of the existing related schemes.

Table 7.4: Security comparison with existing authentication schemes for TMIS.

	Awasthi-Srivastava [24]	Jiang *et al.* [108]	Mishra *et al.* [146]	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.
✓: secure against a particular attack; X: insecure against a particular attack;

7.6.2 Functionality features comparison

In Table 7.5, we compare different functionalities of our proposed scheme with existing related authentication schemes for TMIS. For comparison, we consider 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.

Table 7.5: Functionality features comparison with existing authentication schemes for TMIS.

	Awasthi-Srivastava [24]	Jiang *et al.* [108]	Mishra *et al.* [146]	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.
✓: a scheme fulfills the functionality requirement; X: a scheme does not fulfill the functionality requirement.

7.6.3 Performance comparison

In this section, we compare the computation costs and communication costs among the proposed scheme and other related schemes, such as the schemes of Awasthi-Srivastava [24], Jiang *et al.* [108] and Mishra *et al.* [146].

1) Computation costs comparison

Table 7.6 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 7.6 are based on an existing experiment conducted on an Intel Pentium IV 2600 MHz processor with 1024 MB RAM [118]. 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 [87].

Table 7.6: 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	$\approx T_m$

Table 7.7: Computation costs comparison.

Phase /Scheme	Entity	Awasthi-Srivastava [24]	Jiang *et al.* [108]	Mishra *et al.* [146]	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

In Table 7.7, we analyze the efficiency on computation costs of the proposed scheme and the existing schemes for TMIS [24], [108], [146]. 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 computation 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

Table 7.8: 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

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 7.4 and 7.5.

2) Communication costs comparison

In Table 7.8, 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 [6]), the block size of symmetric encryption/decryption (for example, if we apply AES-128 [2]) 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 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 7.9, it is noted that the proposed 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 7.4 and 7.5.

Table 7.9: Comparison of communication costs.

Scheme	Total bits requred
Awasthi-Srivastava [24]	544
Jiang *et al.* [108]	896
Mishra *et al.* [146]	704
Our	$1760 + \|E_j\|$

7.7 Summary

In this chapter, we 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.

Chapter 8

Conclusion and Future Works

This chapter summarizes the major contributions of the thesis. It also highlights the road-map for future research directions in the design of user authentication and access control schemes in various applications of wireless communication.

8.1 Contributions

The contributions of the thesis are summarized as follows. In this thesis, we have proposed several novel user authentication and access control schemes for various applications of wireless communication, which are listed below:

- Biometric-based user authenticated key agreement scheme for multi-server environment
- Biometric-based user authenticated key agreement scheme for crowdsourcing IoT environment
- Biometric-based user authenticated key agreement scheme for mobile cloud computing services environment
- Fine-gained access control with user authentication scheme for telecare medicine information systems environment

In the first contribution provided in **Chapter** 4, we have designed a novel three factor user authentication scheme tailored for a multi-server system. The major achievement of the scheme is that, it avoids any involvement of the registration center (RC) during the login and authentication process. Moreover, the mutual authentication and session key establishment

process is quite efficient and secure. We avoided any computationally expensive cryptographic operations, and applied lightweight Chebyshev chaotic map, cryptographic hash function and symmetric key encryption-decryption techniques. Through various informal and formal security analysis and verification, we exhibited that the proposed scheme resists various security attacks. Comparison result indicates that the proposed scheme is more efficient than existing related schemes in terms of computation and communication costs.

In the second contribution provided in **Chapter 5**, we have proposed a new biometric-based user authentication for a remote e-healthcare based application of crowdsourcing Internet of Things. As wireless communication is susceptible to various kinds of threats and attacks, remote user authentication is essential for a secure access of medical services over the Internet. Considering the efficiency and security of the proposed scheme, it is quite suitable for resource constrained or battery limited user mobile devices. Security verification is performed using pi calculus based ProVerif 1.93 tool. Security of authentication goals is checked using the Burrows-Abadi-Needham (BAN) logic. Formal security analysis is done using Real-Or-Random (ROR) model. Finally, through extensive performance comparison, we have shown that we have considerably reduced total computation and communication costs as compared to other existing related schemes. As a result, the proposed scheme is very suitable for e-healthcare systems.

In the third contribution provided in **Chapter 6**, we have proposed a new mobile user authentication scheme in a distributed mobile cloud computing environment. The major contribution of the proposed multi-server authenticaiton scheme is that the mutual authentication and session key establishment is done using only cryptographic hash function, bitwise XOR operation and fuzzy extractor function. We avoided any cryptosystem that involves high computation or communication cost. Before providing any access of cloud service to a mobile user, mutual authentication of a mobile user and the cloud service provider is necessary. We have also provided the formal security proof through the ROR model and also the formal security verification through the ProVerif 1.93 simulation tool. Moreover, mutual authentication proof is provided by the BAN logic. As a whole, high security and low communication and computation costs make the proposed scheme very suitable for the practical applications in the mobile cloud computing domain.

In the fourth and final contribution provided in **Chapter 7**, we have proposed a new fine-grained access control using smart card and biometric based user authentication scheme 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. The proposed scheme uses both the user

password and biometric to provide better security as compared to two-factor authentication schemes. The proposed scheme provides group-based user authentication depending on the access rights provided for the genuine users in TMIS. The formal security of the authentication process is verified through widely-accepted ROR model, and then the mutual authentication is achieved using the BAN logic. In addition, the informal security analysis shows that the proposed scheme is also resistant to various security attacks. The proposed scheme provides better security and functinality features as compared to other existing schemes for TMIS.

8.2 Future research directions

In this section, we suggest some directions for possible future works. Several research directions are worth investigating as follows.

8.2.1 Device-to-device authentication in IoT environment

Device authentication in IoT is a security mechanism to ensure that only the authorized devices can connect to a given IoT network environment [102], [178], [182]. Both in consumer and business environments, IoT technology shows its potential growth. In these environments, overall security management of endpoint devices through device authentication and authorization is extremely important, especially in the areas of machine-to-machine (M2M) communications. An IoT network usually contains a large number of devices, which might be heterogeneous in nature. These devices connect to the network in an intermittent way and require to communicate securely with other devices as well as the backend infrastructure. As a result, designing of an efficient device-to-device authentication technique in the IoT environment could be an interesting a future research direction.

8.2.2 Physically secure user authentication in IoT environment

A Physically Unclonable Function (PUF) is considered as a one-way function that maps a set of challenges to another set of responses based on the unique physical micro structure of a device. PUF is a promising primitive to achieve authentication, access control, and traceability. It is also very useful for secure and low-cost authentication [138]. An ideal PUF has the following important properties:

- The output of the PUF is always dependent on a physical system.

- It is easy to evaluate and construct the PUF.
- PUF output is unpredictable in nature and it works as a random function.
- PUF is also uncloneable.

It is therefore interesting to devise a lightweight privacy-preserving authentication scheme for the IoT system by considering PUF as it is done for other environment [76].

8.2.3 Fine-grained access control in IoT environment

IoT system contains various network devices that generate massive heterogeneous data. Intrinsically, these data are meant to be accessed by various classes of users, who might be in different hierarchies with respect to the IoT environment. One feasible solution to handle this issue is to provide a fine-grained access of the data to the users [171]. It is worth mentioning that many of the IoT devices are quite resource constrained. Hence, a traditional approach for handling fine-grained access control through an attribute-based encryption might be an impractical approach. This is because modular exponentiation and bilinear pairing could be computationally too expensive for the resource-constrained IoT devices. Overall, obtaining a more comprehensive fine-grained solution for the entire application related to the IoT system is also another interesting future research direction.

8.2.4 Real-world implemention

The proposed protocols in this thesis are aimed to design for the real-life applications having high societal impact like telecare nedicine information system, crowdsourcing IoT applications, and mobile cloud computing environment. In this thesis, we mainly focus on two parts for security analysis: i) fulfilling application specific security and functionality requirements, and ii) providing formal analysis, informal (non-mathematical) security analysis and also formal security verification using the software based formal verification tools, such as AVISPA and ProfVerif.

In the future, we will evaluate the designed protocols proposed in this thesis in a real-world environment. This allows us to fine-tune the protocols, if necessary, to offer better security and performance in a real-world deployment.

Bibliography

[1] ABI Research Report, Mobile Cloud Applications. Available at: https://www.abiresearch .com/market-research/product/1004607-enterprise-mobile-cloud-computing/. Accessed on June 2018.

[2] Advanced encryption standard. FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001.

[3] AVISPA Tool Documentation. Automated Validation of Internet Security Protocols and Applications. http://www.avispa-project.org/package/user-manual.pdf. Accessed on March 2018.

[4] Mobile Cloud Computing. Available at https://en.wikipedia.org/wiki/Mobile-cloud-computing, Accessed on June 2018.

[5] ProVerif. http://proverif.rocq.inria.fr/index.php. Accessed on Jaunary 2017.

[6] Secure Hash Standard. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. http://csrc.nist.gov/publications/fips /fips180-4/ fips-180-4.pdf. Accessed on August 2018.

[7] Security in Telecommunications and Information Technology: An Overview of Issues and the Deployment of Existing ITU-T Recommendations for Secure Telecommunications. Available at https://www.itu.int/itudoc/itu-t/85097.pdf, Accessed on June 2018.

[8] Smart Card Overview. http://www.oracle.com/technetwork/java/javacard/ documentation /smartcards-136372.html. Accessed on August 2016.

[9] US National Security Agency: Information Assurance. Available at http://www.nsa.gov/ia/ia-at-nsa/index.shtml, Accessed on June 2018.

[10] ITU Internet Reports: The Internet of Things, 2005. International Telecommunication Union (ITU), Geneva. Available at https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf. Accessed on September 2018.

[11] IoT Devices & Products, 2019. https://www.postscapes.com/internet-of-things-award/winners/.

[12] M. Abadi, B. Blanchet, and H. Comon-Lundh. Models and Proofs of Protocol Security: A Progress Report. In *21st International Conference on Computer Aided Verification (CAV'09)*, pages 35–49, Grenoble, France, 2009.

[13] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. *SIGPLAN Notice*, 36(3):104–115, 2001.

[14] M. Abdalla, P. A. Fouque, and D. Pointcheval. Password-based authenticated key exchange in the three-party setting. In *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science*, volume 3386, pages 65–84, Les Diablerets, Switzerland, 2005.

[15] I. F. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.

[16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communication Surveys & Tutorials*, 17(4):2347–2376, 2015.

[17] M. Ali, S. U. Khan, and A. V. Vasilakos. Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305:357–383, 2015.

[18] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cullar, P. Drielsma, P.-C. Ham, O. Kouchnarenko, J. Mantovani, S. Mdersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigan, and L. Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *17th International Conference on Computer Aided Verification (CAV'05), Lecture Notes in Computer Science (LNCS), Springer-Verlag*, volume 3576, pages 281–285, Edinburgh, UK, 2005.

[19] L. Atzori, A. Iera, and G. orabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[20] J. P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia. Quark: A Lightweight Hash. *Journal of Cryptology*, 26(2):313–339, 2013.

[21] J. P. Aumasson, L. Henzen, W. Meier, and M. N. Plasencia. Quark: A Lightweight Hash. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES'10), Lecture Notes in Computer Science (LNCS)*, volume 6225, pages 1–15, Santa Barbara, California, USA, 2010.

[22] AVISPA. Automated Validation of Internet Security Protocols and Applications. http://www.avispa-project.org/. Accessed on March 2016.

[23] AVISPA. SPAN, the Security Protocol ANimator for AVISPA. http://www.avispa-project.org/. Accessed on March 2016.

[24] A. K. Awasthi and K. Srivastava. A Biometric Authentication Scheme for Telecare Medicine Information Systems with Nonce. *Journal of Medical Systems*, 37(5):1–4, 2013.

[25] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad. Proposed embedded security framework for Internet of Things (IoT). In *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*, pages 1–5, Chennai, India, 2011.

[26] M. Ballare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security (CCS'93)*, pages 62–73, Fairfax, Virginia, USA, 1993.

[27] K. C. Baruah, S. Banerjee, M. P. Dutta, and C. T. Bhunia. An improved biometric-based multi-server authentication scheme using smart card. *International Journal of Security and Its Applications*, 9(1):397–408, 2015.

[28] D. Basin, S. Modersheim, and L. Vigano. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, 4(3):181–208, 2005.

[29] P. Bergamo, P. D'Arco, A. D. Santis, and L. Kocarev. Security of public-key cryptosystems based on chebyshev polynomials. *IEEE Transactions on Circuits and Systems*, 52(7):1382–1393, 2005.

[30] G. R. Blakley. Safeguarding cryptographic keys. In *American Federation of Information Processing Societies National Computer Conference*, pages 313–317, Montvale, New Jersey, USA, 1979.

[31] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001*, pages 213–229, Santa Barbara, California, USA, 2001.

[32] J. Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *IEEE Symposium on Security and Privacy (S & P'12)*, pages 538–552, San Francisco, California, USA, May 2012.

[33] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment.* Springer Publishing Company Incorporated, Berlin, 1st edition, 2010.

[34] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[35] Y. Cai, F. Yu, and S. Bu. Cloud computing meets mobile wireless communications in next generation cellular networks. *IEEE Network*, 28(6):54–59, 2014.

[36] S. Challa, A. K. Das, S. Kumari, V. Odelu, F. Wu, and X. Li. Provably secure three-factor authentication and key agreement scheme for session initiation protocol. *Security and Communication Networks*, 9(18):5412–5431, 2016.

[37] S. Challa, A. K. Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan, and A. V. Vasilakos. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 69:534–554, 2018.

[38] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E. J. Yoon, and K. Y. Yoo. Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications. *IEEE Access*, 5(1):3028–3043, 2017.

[39] Y. F. Chang, S. H. Yu, and D. R. Shiao. A Uniqueness-and-Anonymity-Preserving Remote User Authentication Scheme for Connected Health Care. *Journal of Medical Systems*, 37(2):1–9, 2013.

[40] S. Chatterjee and A. K. Das. An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks. *Security and Communication Networks*, 8(9):1752–1771, 2015.

[41] 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 *International Conference on Signal Propagation and Computer Technology, (ICSPCT'14)*, pages 313–317, Ajmer, India, 2014.

[42] S. Chatterjee, A. K. Gupta, and G. Sudhakar. An efficient dynamic fine grained access control scheme for secure data access in cloud networks. In *IEEE International Conference on Electrical Computer and Communication Technologies (ICECCT'15)*, pages 1–8, Coimbatore, India, 2015.

[43] S. Chatterjee and S. Roy. Cryptanalysis and Enhancement of A Distributed Fine-grained Access Control in Wireless Sensor Networks. In *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI'14)*, pages 2074–2083, New Delhi, India, 2014.

[44] S. Chatterjee, S. Roy, and S. Chattopadhyay. An efficient fine-grained access control scheme for hierarchical wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing (InderScience)*, 29(3):161–180, 2018.

[45] 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 Transactions on Dependable and Secure Computing*, 15(5):824–839, 2018.

[46] A. Chaturvedi, A. K. Das, D. Mishra, and S. Mukhopadhyay. Design of a secure smart card-based multi-server authentication scheme. *Journal of Information Security and Applications*, 30:64–80, 2016.

[47] S. A. Chaudhry, K. Mahmood, H. Naqvi, and M. K. Khan. An Improved and Secure Biometric Authentication Scheme for Telecare Medicine Information Systems Based on Elliptic Curve Cryptography. *Journal of Medical Systems*, 39(11):1–12, 2015.

[48] F. Chen, C. Zhang, F. Wang, J. Liu, X. Wang, and Y. Liu. Cloud-Assisted Live Streaming for Crowdsourced Multimedia Content. *IEEE Transactions on Multimedia*, 17(19):1471–1483, 2015.

[49] H. Chen, J. Lo, and C. Yeh. An Efficient and Secure Dynamic ID-based Authentication Scheme for Telecare Medical Information Systems. *Journal of Medical Systems*, 36(6):3907–3915, 2012.

[50] K. R. Choo. Mobile Cloud Storage Users. *IEEE Cloud Computing*, 1(3):20–23, 2014.

[51] M. C. Chuang and M. C. Chen. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications*, 41(4):1411–1418, 2014.

[52] Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen. Access control in user hierarchy based on elliptic curve cryptosystem. *Information Sciences*, 178(1):230–243, 2008.

[53] C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In *8th IMA International Conference on Cryptography and Coding*, pages 360–363, Cirencester, UK, 2001.

[54] D. Meilander, F. Glinka, S. Gorlatch, L. Lin, W. Zhang, and X. Liao. Bringing mobile online games to clouds. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 48–56, Toronto, Ontario, Canada, 2014.

[55] A. K. Das. Analysis and Improvement on an Efficient biometric-based remote user authentication scheme using smart cards. *IET Information Security*, 5(3):145–151, 2011.

[56] A. K. Das. A Secure and Efficient User Anonymity-Preserving Three-Factor Authentication Protocol for Large-Scale Distributed Wireless Sensor Networks. *Wireless Personal Communications*, 82(3):1377–1404, 2015.

[57] A. K. Das. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. *Peer-to-Peer Networking and Applications*, 9(1):223–244, 2016.

[58] A. K. Das. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *International Journal of Communication Systems*, 30(1):1–25, 2017.

[59] 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. *Journal of Medical Systems*, 37(5):1–17, 2013.

[60] A. K. Das and A. Goswami. A Secure and Efficient Uniqueness-and-Anonymity-Preserving Remote User Authentication Scheme for Connected Health Care. *Journal of Medical Systems*, 37(3):1–16, 2013.

[61] A. K. Das and A. Goswami. A robust anonymous biometric-based remote user authentication scheme using smart cards. *Journal of King Saud University - Computer and Information Sciences*, 27(2):193–210, 2015.

[62] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang. Provably secure user authentication and key agreement scheme for wireless sensor networks. *Security and Communication Networks*, 9(16):3670–3687, 2016.

[63] A. K. Das, V. Odelu, and A. Goswami. A Secure and Robust User Authenticated Key Agreement Scheme for Hierarchical Multi-medical Server Environment in TMIS. *Journal of Medical Systems*, 39(9):1–24, 2015.

[64] A. K. Das, A. K. Sutrala, S. Kumari, V. Odelu, M. Wazid, and X. Li. An efficient multi-gateway based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Security and Communication Networks*, 9(13):2070–2092, 2016.

[65] A. K. Das, M. Wazid, N. Kumar, M. K. Khan, K. K. R. Choo, and Y. Park. Design of Secure and Lightweight Authentication Protocol for Wearable Devices Environment. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1310–1322, 2018.

[66] A. K. Das, S. Zeadally, and D. He. Taxonomy and analysis of security protocols for internet of things. *Future Generation Computer Systems*, 89:110–125, 2018.

[67] A. K. Das, S. Zeadally, and M. Wazid. Lightweight authentication protocols for wearable devices. *Computers & Electrical Engineering*, 63:196–208, 2017.

[68] M. L. Das, A. Saxena, and V. P. Gulati. A dynamic ID-based remote user authentication scheme. *IEEE Transactions on Consumer Electronics*, 50(2):629–631, 2004.

[69] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology - EUROCRYPT 2004*, pages 523–540, Interlaken, Switzerland, 2004.

[70] D. Dolev and A. Yao. On the security of public key protocols . *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[71] M. S. Farash, M. Turkanovic, S. Kumari, and M. Holbl. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Networks*, 36(1):152–176, 2016.

[72] F.Wen. A Robust Uniqueness-and-Anonymity-Preserving Remote User Authentication Scheme for Connected Health Care. *Journal of Medical Systems*, 37(6):1–9, 2013.

[73] D. Gafurov, E. Snekkenes, and T. E. Buvarp. Robustness of Biometric Gait Authentication Against Impersonation Attack. In *International Workshop on Information Security, On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 479–488, Montpellier, France, 2006.

[74] J. Geng and L. Zhang. A dynamic id-based user authentication and key agreement scheme for multi-server environment using bilinear pairings. In *Workshop on Power Electronics and Intelligent Transportation System (PEITS 2008)*, pages 33–37, Guangzhou, China, 2008.

[75] P. Gope and A. K. Das. Robust anonymous mutual authentication scheme for n-times ubiquitous mobile cloud computing services. *IEEE Internet of Things Journal*, 4(5):1764–1772, 2017.

[76] P. Gope, J. Lee, and T. Q. S. Quek. Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions. *IEEE Transactions on Information Forensics and Security*, 13(11):2831–2843, 2018.

[77] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM conference on Computer and Communications Security (CCS'06)*, pages 89–98, Alexandria, Virginia, USA, 2006.

[78] J. Granjal, E. Monteiro, and J. S. Silva. Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials*, 17(3):1294–1312, 2015.

[79] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[80] C. Guo and C. C. Chang. Chaotic maps-based password authenticated key agreement using smart cards. *Communications in Nonlinear Science and Numerical Simulation*, 18(6):1433–1440, 2013.

[81] F. Hamad, L. Smalov, and A. James. Energy-aware Security in M-Commerce and the Internet of Things. *IETE Technical Review*, 26(5):357–362, 2009.

[82] X. Hao, J. Wang, Q. Yang, X. Yan, and P. Li. A chaotic map-based authentication scheme for telecare medicine information systems. *Journal of Medical Systems*, 37(2):1–7, 2013.

[83] D. He. Security flaws in a biometrics-based multi-server authentication with key agreement scheme. *IACR Cryptology ePrint Archive*, 2011:1–9, 2011.

[84] D. He, J. Bu, S. Chan, C. Chen, and M. Yin. Privacy-Preserving Universal Authentication Protocol for Wireless Communications. *IEEE Transactions on Wireless Communications*, 10(2):431–436, 2011.

[85] D. He, C. Jianhua, and Z. Ru. A More Secure Authentication Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 37(3):1989–1995, 2012.

[86] D. He, N. Kumar, M. K. Khan, and J. H. Lee. Anonymous Two-factor Authentication for Consumer Roaming Service in Global Mobility Networks. *IEEE Transactions on Consumer Electronics*, 59(4):811–817, 2013.

[87] D. He, N. Kumar, J. H. Lee, and R. S. Sherratt. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Transactions on Consumer Electronics*, 60(1):30–37, 2014.

[88] D. He and D. Wang. Robust Biometrics-Based Authentication Scheme for Multiserver Environment. *IEEE Systems Journal*, 9(3):816–823, 2015.

[89] D. He and S. Zeadally. An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography. *IEEE Internet of Things Journal*, 2(1):72–83, 2015.

[90] R. Heartfield and D. Gan. Social engineering in the internet of everything. *Cutter IT Journal*, 29(7):20–29, 2016.

[91] F. B. Hildebrand. *Introduction to Numerical Analysis*. New York: Dover, second edition, 1974.

[92] W. B. Hsieh and J. S. Leu. Efficient and Anonymous Mobile User Authentication Protocol Using Self-Certified Public Key Cryptography for Multi-Server Architectures. *The Journal of Supercomputing*, 70(1):133–148, 2014.

[93] C. L. Hsu and T. W. Lin. Password authenticated key exchange protocol for multi-server mobile networks based on chebyshev chaotic map. In *IEEE International Conference on Pervasive Computing and Communications (PERCOM'13)*, pages 90–95, San Diego, California, USA, 2013.

[94] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu. Further Observations on Smart-Card-Based Password-Authenticated Key Agreement in Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1767–1175, 2014.

[95] X. Huang, P. Craig, H. Lin, and Z. Yan. SecIoT: a security framework for the Internet of Things. *Security and Communication Networks*, 9(16):3083–3094, 2016.

[96] X. Huang, X. Gao, and Z. Yan. Security protocols in body sensor networks using visible light communications. *International Journal of Communication Systems*, 29(16):2349–2363, 2016.

[97] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu. Robust Multi-Factor Authentication for Fragile Communications. *IEEE Transactions on Dependable and Secure Computing*, 11(6):568–581, 2014.

[98] R. V. Jadhav, S. S. Lokhande, and V. N. Gohokar. Energy Management System in Smart Grid using Internet of Things. In *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES'16)*, pages 1–4, Delhi, India, 2016.

[99] S. Jaiganesh, K. Gunaseelan, and V. Ellappan. IOT agriculture to improve food and farming technology. In *IEEE Conference on Emerging Devices and Smart Systems (ICEDSS)*, pages 260–266, Tiruchengode, India, 2017.

[100] A. Jain, L. Hong, and S. Pankanti. Biometric Identification. *Communications of the ACM*, 43(2):90–98, 2000.

[101] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(14):4–20, 2004.

[102] S. Jang, D. Lim, J. Kang, and I. Joe. An Efficient Device Authentication Protocol Without Certification Authority for Internet of Things. *Wireless Personal Communications*, 91(4):1681–1695, 2016.

[103] P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, and A. Zaslavsky. Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt. *Sensors*, 16(11):1–17, 2016.

[104] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He. A privacy preserving three-factor authentication protocol for e-health clouds. *The Journal of Supercomputing*, 72(10):3826–3849, 2016.

[105] Q. Jiang, J. Ma, G. Li, and L. Yang. An Enhanced Authentication Scheme with Privacy Preservation for Roaming Service in Global Mobility Networks. *Wireless Personal Communications*, 68(4):1477–1491, 2013.

[106] Q. Jiang, J. Ma, G. Li, and L. Yang. An Efficient Ticket Based Authentication Protocol with Unlinkability for Wireless Access Networks. *Wireless Personal Communications*, 77(2):1489–1506, 2014.

[107] Q. Jiang, J. Ma, X. Lu, and Y. Tian. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. *Peer-to-Peer Networking and Applications*, 8(6):1070–1081, 2014.

[108] 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. *Journal of Medical Systems*, 38(2):1–8, 2014.

[109] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi. Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. *Nonlinear Dynamics*, 83(4):2085–2101, 2015.

[110] A. T. B. Jina, D. N. C. Linga, and A. Goh. Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245–2255, 2004.

[111] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu. Security of the Internet of Things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, 2014.

[112] H. J. Jo, J. H. Paik, and D. H. Lee. Efficient privacy-preserving authentication in wireless mobile networks. *IEEE Transactions on Mobile Computing*, 13(7):1469–1481, 2014.

[113] K. Chard, S. Caton, O. Rana, and K. Bubendorfer. Social cloud: Cloud computing in social networks. In *IEEE 3rd International Conference on Cloud Computing*, pages 99–106, Miami, Florida, USA, 2010.

[114] M. K. Khan, K. Alghathbar, and J. Zhang. Secure and Tokenless Privacy-Protecting Chaotic Revocable Biometrics Authentication Scheme. *Telecommunication Systems*, 47(3-4):227–234, 2011.

[115] H. Kim, W. Jeon, K. Lee, Y. Lee, and D. Won. Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme. In *12th International Conference on Computational Science and Its Applications (ICCSA'12)*, pages 391–406, Salvador de Bahia, Brazil, 2012.

[116] K. W. Kim and J. D. Lee. On the Security of Two Remote User Authentication Schemes for Telecare Medical Information Systems. *Journal of Medical Systems*, 38(5):1–11, 2014.

[117] N. Koblitz. Elliptic Curves Cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[118] L. Kocarev and S. Lian. Chaos-Based Cryptography: Theory, Algorithms and Applications. *SCI Book Series, Springer*, 2011.

[119] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO'99, LNCS*, volume 1666, pages 388–397, Santa Barbara, California, USA, 1999.

[120] W.-C. Ku. Impersonation Attack on a Dynamic ID-Based Remote User Authentication Scheme Using Smart Cards. *IEICE Transactions on Communications*, E88-B(5):2165–2167, 2005.

[121] S. Kumari and M. K. Khan. Cryptanalysis of A Robust Smart-Card-Based Remote User Password Authentication Scheme. *International Journal of Communication Systems*, 27(12):3939–3955, 2104.

[122] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. *Future Generation Computer Systems*, 63:56–75, 2016.

[123] L. Lamport. Password Authentication with Insecure Communication. *Communications of the ACM*, 24(11):770–772, 1981.

[124] C. C. Lee, P. S. Chung, and M. S. Hwang. A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments. *International Journal of Network Security*, 15(4):231–240, 2013.

[125] M. Lee, J. Hwang, and H. Yoe. Agricultural Production System Based on IoT. In *16th IEEE International Conference on Computational Science and Engineering*, pages 833–837, Sydney, Australia, 2013.

[126] T. F. Lee. An Efficient Chaotic Map-Based Authentication and Key Agreement Scheme Using Smartcards for Telecare Medicine Information Systems. *Journal of Medical Systems*, 37(6):1–9, 2013.

[127] C. T. Li, C. C. Lee, and C. Y. Weng. A Secure Chaotic Maps and Smart Cards Based Password Authentication and Key Agreement Scheme with User Anonymity for Telecare Medicine Information Systems. *Journal of Medical Systems*, 38(9):1–11, 2014.

[128] H. Li, F. Li, C. Song, and Y. Yan. Towards smart card based mutual authentication schemes in cloud computing. *KSII Transactions on Internet and Information Systems*, 9(7):2719–2735, 2015.

[129] X. Li, J. Ma, W. D. Wang, Y. P. Xiong, and J. S. Zhang. A novel smart card and dynamic id based remote user authentication scheme for multi-server environments. *Mathematical and Computer Modelling*, 58(1-2):85–95, 2012.

[130] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, and C.-L. Liu. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, 34(1):73–79, 2011.

[131] X. Li, Q. Wen, W. Li, H. Zhang, and Z. Jin. A biometric-based Password Authentication with key Exchange Scheme using Mobile Device for Multi-Server Environment. *Applied Mathematics & Information Sciences*, 9(3):1123–1137, 2015.

[132] Y. P. Liao and S. S. Wang. A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, 31(1):24–29, 2009.

[133] Y. Lu, L. Li, X. Yang, and Y. Yang. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. *PloS One*, 10(5):1–13, 2015.

[134] A. Lumini and L. Nanni. An improved BioHashing for human authentication. *Pattern Recognition*, 40(3):1057–1065, 2007.

[135] C. G. Ma, D. Wang, and S. D. Zhao. Security flaws in two improved remote user authentication schemes using smart cards. *International Journal of Communication Systems*, 27(10):2215–2227, 2014.

[136] T. Maitra, S. H. Islam, R. Amin, D. Giri, M. K. Khan, and N. Kumar. An enhanced multiserver authentication protocol using password and smartcard: cryptanalysis and design. *Security and Communication Networks*, 9(17):4615–4638, 2016.

[137] M. Mana, M. Feham, and B. A. Bensaber. SEKEBAN (Secure and Efficient Key Exchange for wireless Body Area Network). *International Journal of Advanced Science and Technology*, 12:1–16, 2009.

[138] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer. Implementation and Characterization of a Physical Unclonable Function for IoT: A Case Study With the TERO-PUF. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):97–109, 2018.

[139] M. Masdari and S. Ahmadzadeh. A survey and taxonomy of the authentication schemes in Telecare Medicine Information Systems. *Journal of Network and Computer Applications*, 87:1–19, 2017.

[140] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *14th ACM international conference on Mobile computing and networking (MobiCom'08)*, San Francisco, California, USA, 2008.

[141] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, New York, 3rd edition, 2001.

[142] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5):541–552, 2002.

[143] R. Miceli. Energy Management and Smart Grids. *Energies*, 6(4):2262–2290, 2013.

[144] D. Mishra, A. K. Das, and S. Mukhopadhyay. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications*, 41(18):8129–8143, 2014.

[145] 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-to-Peer Networking and Applications*, 1(9):171–192, 2016.

[146] D. Mishra, J. Srinivas, and S. Mukhopadhyay. A Secure and Efficient Chaotic Map-Based Authenticated Key Agreement Scheme for Telecare Medicine Information Systems. *Journal of Medical Systems*, 36(10):1–10, 2014.

[147] M. B. Mollah, M. A. K. Azad, and A.Vasilakos. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, 84:38–54, 2017.

[148] J. Moon, Y. Choi, J. Kim, and D. Won. An Improvement of Robust and efficient Biometrics based Password Authentication scheme for Telecare Medicine Information systems using Extended chaotic maps. *Journal of Medical Systems*, 40(3):1–11, 2016.

[149] O. Mouaatamid, M. Lahmer, and M. Belkasmi. Internet of Things Security: Layered classification of attacks and possible Countermeasures. *Electronic Journal of Information Technology*, 9:66–80, 2016.

[150] J. Munilla and A. Peinado. Off-line password-guessing attack to Peyravian-Jeffries's remote user authentication protocol. *Computer Communications*, 30(1):52–54, 2006.

[151] R. W. D. Nickalls. A new approach to solving the cubic: Cardans solution revealed. *The Mathematical Gazette*, 480(77):354–359, 1993.

[152] V. Odelu, A. K. Das, and A. Goswami. A Secure and Scalable Group Access Control Scheme for Wireless Sensor Networks. *Wireless Personal Communications*, 85(4):1765–1788, 2015.

[153] V. Odelu, A. K. Das, and A. Goswami. A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards. *IEEE Transactions on Information Forensics and Security*, 10(9):1953–1966, 2015.

[154] V. Odelu, A. K. Das, and A. Goswami. An efficient biometric based privacy-preserving three-party authentication with key agreement protocol using smart cards. *Security and Communication Networks*, 8(18):4136–4156, 2015.

[155] V. Odelu, A. K. Das, and A. Goswami. SEAP: Secure and Efficient Authentication Protocol for NFC Applications Using Pseudonyms. *IEEE Transactions on Consumer Electronics*, 62(1):30–38, 2016.

[156] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid. Provably secure authenticated key agreement scheme for distributed mobile cloud computing services. *Future Generation Computer Systems*, 68:74–88, 2017.

[157] P. Pessl and S. Mangard. Enhancing side-channel analysis of binary-field multiplication with bit reliability. In *Cryptographers' Track at the RSA Conference (CT-RSA'16)*, pages 255–270, San Francisco, California, USA, 2016.

[158] D. Pointcheval and S. Zimmer. Multi-factor authenticated key exchange. In *International Conference on Applied Cryptography and Network Security (ACNS'08)*, pages 277–295, New York, USA, 2008.

[159] S. Pokharel, K.-K. R. Choo, and J. Liu. Mobile cloud security: An adversary model for lightweight browser security. *Computer Standards & Interfaces*, 49:71–78, 2017.

[160] V. Prokhorenko, K.-K. R. Choo, and H. Ashman. Web application protection techniques: A taxonomy. *Journal of Network and Computer Applications*, 60:95–112, 2016.

[161] H. Qi and A. Gani. Research on mobile cloud computing: Review, trend and perspectives. In *Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP'12)*, Bangkok, Thailand, 2012.

[162] D. Quick and K.-K. R. Choo. Pervasive social networking forensics: Intelligence and evidence from mobile device extracts. *Journal of Network and Computer Applications*, 86:24–33, 2017.

[163] S. Rane, Y. Wang, S. C. Draper, and P. Ishwar. Secure Biometrics : Concepts , Authentication Architectures & Challenges. *IEEE Signal Processing Magazine*, 30(5):51–64, 2013.

[164] A. G. Reddy, A. K. Das, V. Odelu, and K. Y. Yoo. An Enhanced Biometric Based Authentication with Key-Agreement Protocol for Multi-Server Architecture Based on Elliptic Curve Cryptography. *PLoS One*, 11(5):1–28, 2016.

[165] A. G. Reddy, A. K. Das, E.-J. Yoon, and K.-Y. Yoo. A Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography. *IEEE Access*, 4(1):4394–4407, 2016.

[166] A. G. Reddy, A. K. Das, E. J. Yoon, and K. Y. Yoo. An anonymous authentication with key agreement protocol for multi-server architecture based on biometrics and smartcards. *KSII Transactions on Internet and Information Systems*, 10(7):3371–3396, 2016.

[167] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo. Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment. *IEEE Access*, 5(1):3622–3639, 2017.

[168] S. Roy and S. Chatterjee. Cryptanalysis of a Chaotic Map-Based Authentication and Key Agreement Scheme for Telecare Medicine Information Systems. In *4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA'15)*, pages 527–537, Durgapur, India, 2015.

[169] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos. On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. *IEEE Access*, 5(1):25808–25825, 2017.

[170] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo. Chaotic Map-based Anonymous User Authentication Scheme with User Biometrics and Fuzzy Extractor for Crowdsourcing Internet of Things. *IEEE Internet of Things Journal*, 5(4):2884 – 2895, 2018.

[171] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. Rodrigues. Provably Secure Fine-Grained Data Access Control over Multiple Cloud Servers in Mobile Cloud Computing Based Healthcare Applications. *IEEE Transactions on Industrial Informatics*, 15(1):457–468, 2019.

[172] S. Ruj, A. Nayak, and I. Stojmenovic. Distributed ne-grained access control in wireless sensor networks. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS'11)*, pages 352–362, Anchorage, Alaska, USA, 2011.

[173] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *Advances in Cryptology (EUROCRYPT'05), Springer*, pages 457–473, Aarhus, Denmark, 2005.

[174] S. Saleem, S. Ullah, and H. S. Yoo. On the security issues in wireles body area networks. *International Journal of Digital Content Technology and its Applications*, 3(3):178–184, 2009.

[175] P. Sarkar. A Simple and Generic Construction of Authenticated Encryption with Associated Data. *ACM Transactions on Information and System Security*, 13(4):1–16, 2010.

[176] A. Sawand, S. Djahel, Z. Zhang, and F. N. Abdesselam. Toward Energy-Efficient and Trustworthy eHealth Monitoring System. *China Communications*, 12(1):46–65, 2015.

[177] B. Schneier. *Applied Cryptography Protocols Algorithms and Source Code in C.* John Wiley and Sons Inc., 2nd edition, 1996.

[178] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi. Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption. *IEEE Embedded Systems Letters*, 9(1):1–4, 2017.

[179] M. Scott, N. Costigan, and W. Abdulwahab. Implementing Cryptographic Pairings on Smartcards. In *8th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'06)*, pages 134–147, Yokohama, Japan, 2006.

[180] D. Sgandurra and E. Lupu. Evolution of Attacks, Threat Models, and Solutions for Virtualized Systems. *ACM Computing Surveys*, 48(3):1–38, 2016.

[181] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology (CRYPTO'84)*, pages 47–53, Santa Barbara, California, USA, 1984.

[182] Y. Sharaf-Dabbagh and W. Saad. On the authentication of devices in the Internet of Things. In *17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM'16)*, pages 1–3, Coimbra, Portugal, 2016.

[183] H. Shen, C. Gao, D. He, and L. Wu. New biometrics-based authentication scheme for multi-server environment in critical systems. *Journal of Ambient Intelligence and Humanized Computing*, 6(6):825–834, 2015.

[184] T. Shon, J. Cho, K. Han, and H. Choi. Toward Advanced Mobile Cloud Computing for the Internet of Things: Current Issues and Future Direction. *Mobile Networks and Applications*, 19:404–413, 2014.

[185] V. Shoup. Sequences of games: A tool for taming complexity in security proofs. cryptology ePrint archieve, Report 2004/332. Available at http://eprint.iacr.org/2004/332, 2004.

[186] K. Simoens, J. Bringer, H. Chabanne, and S. Seys. A Framework for Analyzing Template Security and Privacy in Biometric Authentication Systems. *IEEE Transactions on Information Forensics and Security*, 7(2):833–841, 2012.

[187] V. Sivaprasatham, J. Venkateswaran, and H. Omar. Integrated Authentication Based on CDMA Modulation for Physical Layer Security of Wireless Body Area Network. *European Journal of Scientific Research*, 106(3):388–403, 2013.

[188] T. Song, R. Li, B. Mei, J. Yu, X. Xing, and X. Cheng. A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes. *IEEE Internet of Things Journal*, 4(6):1844–1852, 2017.

[189] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, Englewood Cliffs, 3rd edition, 2004.

[190] D. R. Stinson. Some Observations on the Theory of Cryptographic Hash Functions. *Designs, Codes and Cryptography*, 38(2):259–277, 2006.

[191] D. R. Stinson and M. Paterson. *Cryptography: Theory and Practice*. Chapman and Hall/CRC, United Kingdom, 4th edition, 2018.

[192] H. Sun, Q. Wen, H. Zhang, and Z. Jin. A novel remote user authentication and key agreement scheme for mobile client-server environment. *Applied Mathematics & Information Sciences*, 7(4):1365–1374, 2013.

[193] P. Syverson and I. Cervesato. The Logic of Authentication Protocols. In *Foundations of Security Analysis and Design (FOSAD'01)*, pages 63–137, Bertinoro, Italy, 2001.

[194] J. L. Tsai and N. Lo. A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services. *IEEE Systems Journal*, 9(3):805–815, 2015.

[195] J. L. Tsai, N. W. Lo, and T. C. Wu. Novel Anonymous Authentication Scheme Using Smart Cards. *IEEE Transactions on Industrial Informatics*, 9(4):2004–2013, 2013.

[196] Y. M. Tseng, S. S. Huang, T. T. Tsai, and J. H. Ke. List-Free ID-Based Mutual Authentication and Key Agreement Protocol for Multiserver Architectures. *IEEE Transactions on Emerging Topics in Computing*, 4(1):102–112, 2016.

[197] Y. M. Tseng, T. Y. Wu, and J. Wu. A Pairing-Based User Authentication Scheme for Wireless Clients with Smart Cards. *Informatics*, 19(2):285–302, 2008.

[198] G. Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pages 1–4, Pisa, Italy, 2006.

[199] K. Vangani. Crowdsourcing for IoT!, 2016. Available at https://medium.com/intelligent-cities/crowdsourcing-for-iot-f6636289ba09.

[200] D. von Oheimb. The high-level protocol specification language hlpsl developed in the eu project avispa. In *3rd APPSEM II Workshop on Applied Semantics (APPSEM 2005)*, pages 1–17, Frauenchiemsee, Germany, 2005.

[201] C. Wang, X. Zhang, and Z. Zheng. Cryptanalysis and Improvement of a Biometric-Based Multi-Server Authentication and Key Agreement Scheme. *PLoS One*, 11(2):1–25, 2016.

[202] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian. Zipf's Law in Passwords. *IEEE Transactions on Information Forensics and Security*, 12(11):2776–2791, 2017.

[203] D. Wang, Q. Gu, H. Cheng, and P. Wang. The Request for Better Measurement: A Comparative Evaluation of Two-Factor Authentication Schemes. In *11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS'16)*, pages 475–486, Xi'an, China, 2016.

[204] D. Wang, D. He, P. Wang, and C. H. Chu. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment. *IEEE Transactions on Dependable and Secure Computing*, 12(4):428–442, 2015.

[205] D. Wang and C. G. Ma. Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards. *The Journal of China Universities of Posts and Telecommunications*, 19(5):104–114, 2012.

[206] D. Wang, C. G. Ma, P. Wang, and Z. Chen. Robust Smart Card Based Password Authentication Scheme Against Smart Card Security Breach. Cryptology ePrint Archive, Report 2012/439, 2012. Available at http://eprint.iacr.org/2012/439.pdf.

[207] D. Wang and P. Wang. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Computer Networks*, 73:41–57, 2014.

[208] D. Wang and P. Wang. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Networks*, 20:1–15, 2014.

[209] D. Wang and P. Wang. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Transactions on Dependable and Secure Computing*, 15(4):708–722, 2018.

[210] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang. Targeted Online Password Guessing: An Underestimated Threat. In *ACM Conference on Computer and Communications Security (CCS'16)*, pages 1242–1254, Vienna, Austria, 2016.

[211] M. Wazid, A. K. Das, M. K. Khan, A. A.-D. Al-Ghaiheb, N. Kumar, and A. V. Vasilakos. Secure Authentication Scheme for Medicine Anti-counterfeiting System in IoT Environment. *IEEE Internet of Things Journal*, 4(5):1634–1646, 2017.

[212] M. Wazid, A. K. Das, N. Kumar, V. Odelu, A. G. Reddy, K. Park, and Y. Park. Design of Lightweight Authentication and Key Agreement Protocol for Vehicular Ad Hoc Networks. *IEEE Access*, 5(1):14966–14980, 2017.

[213] M. Wazid, A. K. Das, N. Kumar, and J. P. C. Rodrigues. Secure Three-factor User Authentication Scheme for Renewable Energy Based Smart Grid Environment. *IEEE Transactions on Industrial Informatics*, 13(6):3144–3153, 2017.

[214] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu. Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS. *Security and Communication Networks*, 9(13):1983–2001, 2016.

[215] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo. Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment. *IEEE Transactions on Dependable and Secure Computing*, 2017. DOI: 10.1109/TDSC.2017.2764083.

[216] M. Wazid, S. Zeadally, and A. K. Das. Mobile Banking: Evolution and Threats: Malware Threats and Security Solutions. *IEEE Consumer Electronics Magazine*, 8(2):56–60, 2019.

[217] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos. Security and privacy for storage and computation in cloud computing. *Information Sciences*, 258:371–386, 2014.

[218] Y. Wen, X. Zhu, J. J. Rodrigues, and C. W. Chen. Cloud mobile media: reflections and outlook. *IEEE Transactions on Multimedia*, 16(4):885–902, 2014.

[219] F. Wu, L. Xu, S. Kumari, and X. Li. A privacy-preserving and provable user authentication scheme for wireless sensor networks based on Internet of Things security. *Journal of Ambient Intelligence and Humanized Computing*, 8(1):101–116, 2017.

[220] S. Wu, Y. Zhu, and Q. Pu. Robust smart-cards-based user authentication scheme with user anonymity. *Security and Communication Networks*, 5(2):236–248, 2012.

[221] X. Yang, T. Pan, and J. Shen. On 3G mobile e-commerce platform based on cloud computing. In *3rd IEEE International Conference on Ubi-Media Computing*, pages 198–201, Jinhua, China, 2010.

[222] Q. Xie, W. Liu, S. Wang, L. Han, B. Hu, and T. Wu. Improvement of a uniqueness-and-anonymity-preserving user authentication scheme for connected health care. *Journal of Medical Systems*, 38(9):1–10, 2014.

[223] J. Xu, W. T. Zhu, and D. G. Feng. An improved smart card based password authentication scheme with provable security. *Computer Standards & Interfaces*, 31(4):723–728, 2009.

[224] L. Xu and F. Wu. Cryptanalysis and Improvement of a User Authentication Scheme Preserving Uniqueness and Anonymity for Connected Health Care. *Journal of Medical Systems*, 39(2):1–9, 2014.

[225] X. Xu, P. Zhu, Q. Wen, Z. Jin, H. Zhang, and L. He. A secure and efficient authentication and key agreement scheme based on ECC for telecare medicine information systems. *Journal of Medical Systems*, 38(1):1–7, 2014.

[226] Z. Yan, X. Li, M. Wang, and A. V. Vasilakos. Flexible Data Access Control based on Trust and Reputation in Cloud Computing. *IEEE Transactions on Cloud Computing*, 5(3):485–498, 2017.

[227] Z. Yan, P. Zhang, and A. V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42:120–134, 2014.

[228] H. Yenginer, C. Cetiz, and E. Dursun. A review of energy management systems for smart grids. In *3rd International Istanbul Smart Grid Congress and Fair (ICSG)*, pages 1–4, Istanbul, Turkey, 2015.

[229] Z. Yin, F. R. Yu, S. Bu, and Z. Han. Joint cloud and wireless networks operations in mobile cloud computing environments with telecom operator cloud. *IEEE Transactions on Wireless Communications*, 14(7):4020–4033, 2015.

[230] E. Yoon and K. Yoo. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The Journal of Supercomputing*, 63(1):235–255, 2013.

[231] S. Yu, K. Ren, and W. Lou. FDAC: Toward Fine-Grained Distributed Data Access Control in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(4):673–686, 2011.

[232] L. Zhang. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos, Solitons and Fractals 37*, 50(1):669–674, 2008.

[233] Q. Zhang, Y. Yin, D. Zhan, and J. Peng. A Novel Serial Multimodal Biometrics Framework Based on Semisupervised Learning Techniques. *IEEE Transactions on Information Forensics and Security*, 9(10):1681–1694, 2014.

[234] D. Zhao, H. Peng, S. Li, and Y. Yang. An efficient dynamic ID based remote user authentication scheme using self-certified public keys for multi-server environment. Available at https://arxiv.org/pdf/1305.6350v1.pdf, Accessed on March 2016.

[235] J. Zhou, Z. Cao, X. Dong, N. Xiong, and A. V. Vasilakos. 4S: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks. *Information Sciences*, 314:255–276, 2015.

[236] J. Zhou, X. Dong, Z. Cao, and A. V. Vasilakos. Secure and Privacy Preserving Protocol for Cloud-Based Vehicular DTNs. *IEEE Transactions on Information Forensics and Security*, 10(6):1299–1314, 2015.