CRYPTANALYSIS AND IMPROVEMENT OF REMOTE USER AUTHENTICATION SCHEMES IN TELECARE MEDICINE INFORMATION SYSTEM

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

CHAITANYA SADANAND NAYAK

DEPARTMENT OF MATHEMATICAL AND COMPUTATIONAL SCIENCES
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE - 575 025
FEBRUARY 2021

Dedicated to

Family

DECLARATION

By the Ph.D. Research Scholar

I hereby declare that the Research Thesis entitled **CRYPTANALYSIS AND IMPROVEMENT OF REMOTE USER AUTHENTICATION SCHEMES IN TELECARE MEDICINE INFORMATION SYSTEM** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Mathematical and Computational Sciences** is a *bonafide report of the research work carried out by me*. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

Place: NITK, Surathkal. (CHAITANYA SADANAND NAYAK)

Date: 155091 MA15F02

Department of Mathematical and Computational Sciences

CERTIFICATE

This is to *certify* that the Research Thesis entitled **CRYPTANALYSIS AND IMPROVEMENT OF REMOTE USER AUTHENTICATION SCHEMES IN TELECARE MEDICINE INFORMATION SYSTEM** submitted by **Ms. CHAITANYA SADANAND NAYAK**, (Register Number: 155091-MA15F02) as the record of the research work carried out by her is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. R. MADHUSUDHAN
Research Supervisor

Chairman - DRPC
(Signature with Date and Seal)

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my research guide, Dr. R. Madhusudhan, for his continuous support throughout the PhD program. His encouraging words and rare insights helped me a lot. I thank him for every lesson I learned from him.

My research journey began with the encouraging words from *Dr. B. M. Chandrashekara*, (Head, Dept of Maths, Dr. G Shankar GWFGC & PG Study Centre, Udupi), who made me realize what I can accomplish. With greatest pleasure, I express my sincere thanks to him. With deep sense of regard, my heartfelt gratitude to my *gajang chinhan chingu, Mr. R. G. Yatish*. I cannot thank him enough for being supportive and helping me in all possible ways to pursue my dream.

My heartfelt thanks to my RPAC members, *Dr. Kedarnath Senapati* (Dept of MACS) and *Dr. Savita Bhat* (School of Management) for their valuable feedback. My sincere gratitude to my HODs (Prof. S. S. Kamath, Prof. B. R. Shankar, Prof. Santhosh George, Prof. N. N. Murulidhar) for their co-operation during my PhD program. Also, I thank all the teaching and non-teaching staff of MACS Dept who have helped me in one way or the other.

My gratitude to *Mrs Parminder, Dr. Shrimathi Adiga, Dr. M. Ravisha, Dr. A. L. Mamatha, Mrs Helen, Prof. Bhavani, Mr G. Ramachandra Adiga, Mr Subraya Bhat* for their encouraging words during my student days. A big shootout to *Dr. Pooja Rao, Dr. L. Lobo, D. Kiran Kumar, U. Pratheeksha, Dr. G. N. Girisha* for sharing their precious time with me. Also, to *Mrs S. R. Shishira, Dr. Manjunath Hegde, Dr. Shashidhara, P. B. Venkataramana and Mrs. Mahadevi* for uplifting my spirits during hard times.

The list would be incomplete without mentioning my parents (*Mrs Sandhya and Mr Sadanand Nayak*) and my brother, *Nayak Chinmay Sadanand*. Without their help, I wouldn't have come this far. I thank them for their constant support.

Lastly, I thank all of those who have helped me knowingly and/or unknowingly throughout my PhD program.

Place: NITK, Surathkal

Date: CHAITANYA SADANAND NAYAK

ABSTRACT OF THE THESIS

The Internet with its high-speed development is making human jobs more easy and less time-consuming. This has enabled us its usage in all the fields, right from school-going kids to professionals working for Multinational Companies. One can not even imagine a day without the Internet. When this has become the scenario today, personnel from all the fields are trying to make the best out of it and medical health people are also a part of this. The traditional methods of waiting in queues for medical consultancy has been transformed to online consultancy. A patient sitting at one part of world can consult a physician at the other end using the Internet. Medical institutes, researchers in the field are able to work with the required data by obtaining them from the medical servers, where the required information is stored. These topics constitute the connected health care. This is a model for health care that uses technology to provide medical assistance remotely. Telecare Medicine Information System (TMIS) is one such system that supports health care delivery services. The information is stored in a server and since the Internet is open to all, preserving patient's identity and information is a very important and challenging task. In other words, authentication is most important. In past, this was easy. Two persons would identify each other by visual appearance. But at present, one cannot 'see' the other in reality. In such case, authentication becomes very complex, specially when the message to be transmitted is confidential. To fulfill this, many authentication schemes using smart cards were and are being proposed. However, many schemes are insecure or they have low efficiency. So, proposing an ideal scheme, which is robust and efficient is the main aim of this research.

Keywords: TMIS, User authentication, Smart card, Chaotic map, Hash function, Biometrics, Session key, BAN logic.

Table of Contents

List of Figures

List of Tables

ABBREVIATIONS

Following are the set of acronyms/abbreviations used throughout this report.

TMIS: Telecare Medicine Information System

EPR: Electronic Patient Record

OTP: One-Time Password

ECG: Electrocardiogram

EEG: Electroencephalogram

RSA: Rivest-Shamir-Adleman

ECC: Elliptic Curve Cryptography

MRI: Magnetic Resonance Imaging

XOR: eXclusive OR operation

EMG: Electromyography

BAN LOGIC: Burrows-Abadi-Needham logic

WBAN: Wireless Body Area Network

CHAPTER 1

INTRODUCTION

The evolution and swift progress in the Internet technology has left no stone unturned. With its high-speed development, the Internet has become a non-separable part of human life. That is to say, it seems very obvious to use the Internet anywhere and everywhere in all the fields. To cite a few, Internet-of-Things, virtual reality, robotics, revolving buildings, artificial intelligence, blockchain technology etc., have benefited the society in one way or the other. Gradually, jobs changed to online jobs, traditional banking changed to e-banking, currency got converted into crypto-currency and many more. So, digitization of user information is becoming more common these days and medical field is no exception to this. Technology has provided medical field with incredible products/procedures in the form of stethoscopes, X-ray machines, heart monitors, wireless brain sensors, robotic surgery, food scanners, cancer nanotherapy and many more. As a result, the concept of e-medicine is introduced and is gaining popularity with each passing day.

In medical organizations, medical personnel have to quickly understand the complete information of patients in order to make instant and accurate diagnoses as well to provide appropriate treatment. An important part of treatment is a patient's medical record. For this, the record must contain complete as well as accurate personal information. Hence, the traditional method of patient-based medical records came into existence. A patient medical record contains a large amount of information in different types of documents, which vary depending on the type of services provided (Takeda et al., 2000). The purpose of medical records is to provide continuity of care (Chen et al., 2012b). But then, the traditional methods have various drawbacks like disorganization, low data mobility, illegibility, space requirement, conservation difficulty and low transferability (Safran and Goldberg, 2000; Uslu and Stausberg, 2008; van Ginneken, 2002). To overcome these drawbacks, traditional paper-based medical records have been transformed to Electronic Patient Record (EPR). The advantages of this approach are accessibility, low cost, easy reporting, readability and diagnostic support (Lovis et al., 1998). Due to these reasons, this concept is gaining much popularity, specially Telecare Medicine Information System (TMIS) and Electronic Patient Record (EPR).

TMIS provides flexible and convenient e-health care. It provides certain healthcare services, which is definitely a feasible solution to the raising demand in medical and health care sector. Most of the medical institutes are developing medical information systems to facilitate connected health care services which provide an opportunity to improve financial and clinical performance (Mishra et al., 2014). It is of increasing importance to everyone as personal health care information becomes easier to access through modern electronic communication systems (Huston, 2001). Equally important is the integrated EPR system. EPR means electronic collection of clinical narrative and diagnostic reports specific to an individual patient that allow medical practitioners to practise in a paperless manner (Safran and Goldberg, 2000). In order to support patients and doctors, the integrated EPR systems are widely used.

Monitoring patient's health data and providing accurate information to medical institutions, analysis and maintenance of patient's health is mostly covered in EPR information system. These systems are more than just repositories for patient data; they combine data, knowledge and software tools, which help patients to become active participants in their own care (Tang et al., 2006). Doctors use this information (ex. ECG, EEG, treatment record of the diseases, etc.) to diagnose and treat disease (Nikooghadam and Zakerolhosseini, 2012). For medical personnel to quickly understand the complete information of patients to make instant and accurate diagnoses as well as to provide appropriate treatment, the record needs to be available in time. In such cases, TMIS comes handy.

A scenario of TMIS is shown in Fig 1.1. There is a patient residing at some place. Suppose that patient needs to get the updates about his current health condition but the required doctor is at some far-off place. Also, there is a remote medical server, which contains the data of the patient, using which the doctor can give the necessary updates. For mutual communication, the patient can either use WiFi or LAN or mobile device. These systems work by registration of patients as well as practitioners in the beginning. After registration, users will login to the system. After successful authentication from both sides, they can communicate with each other and get the required updates or information.

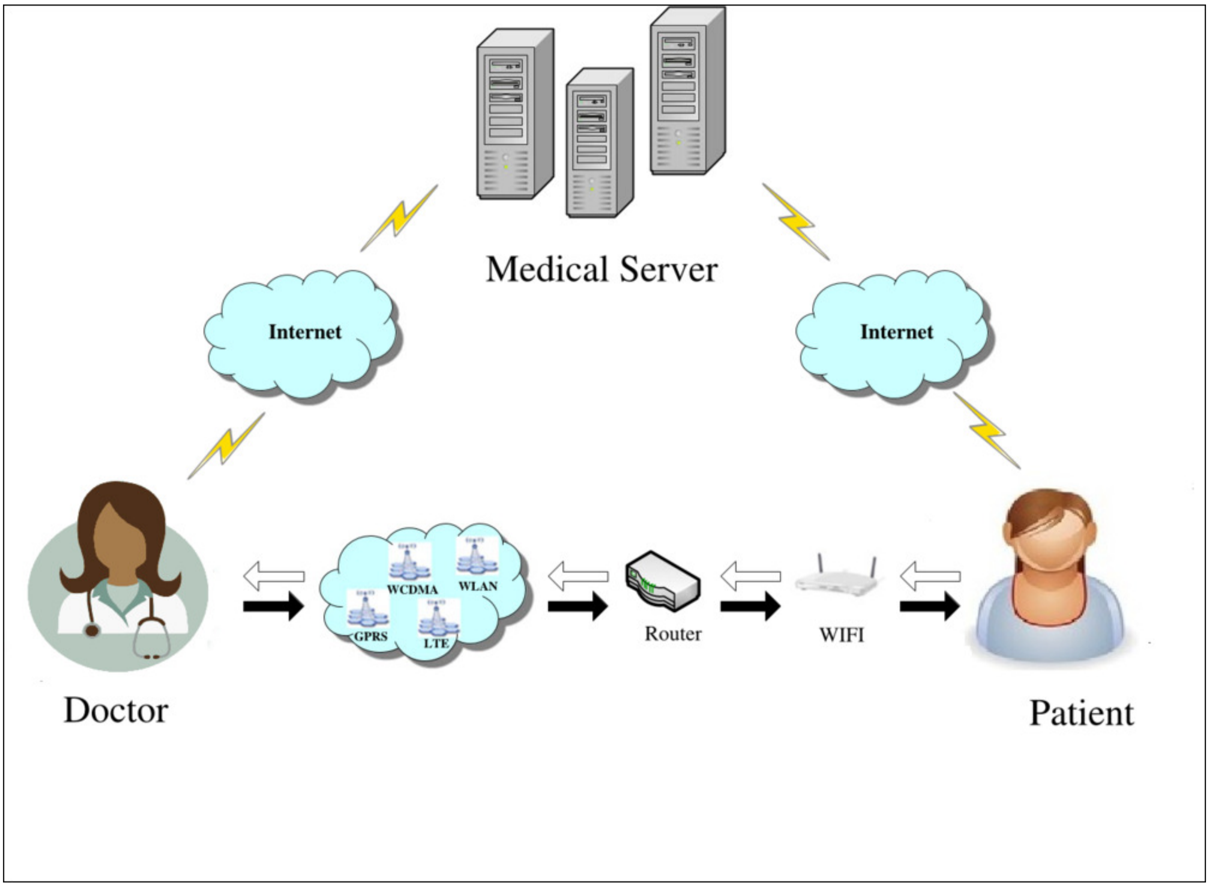


Figure 1.1 A scenario of TMIS

In these systems, records are shared through enterprise-wide, network-connected information system or other information networks and exchanges (Othman et al., 2014). During these data exchange, a lot of private and sometimes, highly confidential information will be transmitted over public channels and are exposed to insecure public networks. Since the Internet is open to all, it is vulnerable to various security attacks. Patients and physicians fear that medical records may not be secure because these systems are web-based (Anderson, 2007). Due to this, there is great risk for loss of privacy and this has to be controlled. The privacy of patients has to be maintained. It is the right of individuals to determine for themselves when, how and to what extent personal information is communicated to others (Arora et al., 2014). For e.g., in the US, federal regulations enacted under the Health Insurance Portability and Accountability Act (HIPAA) require members of the healthcare industry who use electronic information systems to protect the privacy of medical information (Breaux and Antón, 2008). Hence, only authorized users should be able to access data from the medical server. This can be achieved when the users on both the sides can confirm their identities to each other. In other words, user authentication has to be ensured before transmitting sensitive data.

1.1 User Authentication

To use services from a remote server, one must have proper access rights. Security plays a very important role. Here, security refers to authentication, integrity of data (information is unaltered), confidentiality (information in network remains private), access control (only authorized users can communicate) and non-repudiation (originator/receiver of the message cannot deny that he/she sent/received the message). These goals can be achieved with different security types such as firewall, intrusion detection system, cryptography, antivirus software etc.. This is important to ensure the safety of data or information stored in the server.

User authentication is the central component of any security infrastructure. Authentication is a process of positively verifying the identity of a user, device, or other entity in a computer system, as a prerequisite to allowing access to resources in the system (Stocksdale, 1998). Authentication guarantees that the systems resources are not obtained fraudulently by unauthorized users (Wu et al., 2012b). Other security measures depend on verification of the identity of the sender and the receiver of the information. Authorization grants privileges depending on identity. Audit trails would provide accountability only after successful authentication. Confidentiality and integrity are broken if it is not possible to reliably differentiate an authorized entity from an unauthorized entity (Council et al., 1990).

From a long time, user authentication was quite simple. One person, Alice, would meet another person Bob; they would recognize each other by seeing each other. If Alice could not recognize Bob, he could explain that he was a friend or a business partner and Alice could decide whether or not to believe him. But if Alice and Bob were spies, they would use other methods for mutual authentication like piecing together parts of a ripped page, exchanging pre-arranged nonsense statements etc. (Kahn, 1996). In modern era, authentication is different. With computers, one cannot 'see' the entity (who can be a friend, machine or an attacker) on the remote end of the network. The World Wide Web complicates the matter since attackers can access data without physical presence.

With the rapid development in the Internet technology, all human activities are influenced by it, but at the same time, malicious attackers are engrossed in challenging the security of these networks by retrieving the sensitive data or manipulating it by eavesdropping or hacking or any other means. The most recent development to support this statement is the Twitter hacks that took place on 20/07/15. The most powerful Twitter accounts including those of Obama, Elon Musk, Bill Gates etc. saying any Bitcoin sent to the mentioned link would be doubled and sent back to the user; and that this offer would be valid only for 30 minutes. As a result, around 100,000$ were received by a Bitcoin wallet via 300 transactions approximately. Due to this, certain functions were disabled and password reset requests were denied for some time. This incident was also termed as the worst hack of a popular social media platform till date. At this point, providing security is literally challenging and for this reason, authentication becomes very important. So different entities like passwords, tokens, smart cards etc. are routinely used in our interactions over computer networks. For this, strict protocols or mechanisms should be implemented, wherein only a legal user can access the information from a remote server. At the same time, user too should get assurance about the server from which it is accessing information.

1.1.1 Authentication Methods

To provide authentication, different types of authentication services have been developed. The common ones are explained below:

1.1.1.1 Password authentication

Password is a string of characters that allows access to a system. This belongs to "what the user know" type and is often called one-factor authentication. This is widely used and popular because of its simplicity. Also, it is one of the most common control mechanisms for authenticating users of computerized information systems (Morris and Thompson, 1979). In this, each user has his own identity, ID and a password, PW. With the knowledge of the password, the remote user can use it to create and send a valid login message to the server for gaining the access right. On the other hand, authentication server also uses the shared password to check the validity of the login message for authenticating the remote user (Hsu, 2004). Passwords are user-friendly but the use of passwords is a major point of vulnerability in computer security, as they are often easy to guess by automated programs running dictionary attacks (Pinkas and Sander, 2002). In other words, low entropy passwords can be easily guessed. Usage of same password for multiple accounts can help an attacker to obtain information etc. (Furnell, 2005). An easy-to-remember password can often be guessed by an attacker whereas a long or changing password is difficult to guess.

1.1.1.2 Smart card authentication

This falls under "what the user has" category and is often termed as two-factor authentication. A smart card is a chip-based identification card that has an integrated circuit embedded in it, which has components for transmitting, storing and processing data. It contains a tamper-resistant security system and provides security services. The data can be transmitted using either contacts or electromagnetic fields, without any contacts. The card contains some of the information related to user using which he can get access to services from the medical server after authentication. The stored data can be protected against unauthorized access and manipulation (Rankl and Effing, 2004). These will be managed by administration system, which exchanges information and configuration settings securely with the card. In this authentication, user initially registers himself and after that, a smart card will be issued to the user using which along with ID and PW, he/she can access services from the required server. This falls under "what the user has" category and is known as two-factor authentication (when used with password). These are widely used due to their four main characteristics namely portability, security, open platform and memory management (Mohammed et al., 2004). They support a high-level application programming language. In addition to this, a single smart card can perform multiple tasks and act like a container for a number of digital credentials.

1.1.1.3 Biometric authentication

Biometric technologies are automated methods of verifying or recognizing the identity of a living person based on a physiological behavioral characteristic (Wayman et al., 2005). This system relies on the unique biological characteristics like iris, fingerprints, face, hand geometry, keystroke dynamics etc. of the user for authentication, thereby ensuring secure access to the system. This belongs to "what the user is" category and is termed as three-factor authentication (when used with smart card and password). These have multiple advantages like none can steal it, it is not possible to manipulate or forget them etc.. Used alone, or combined with smart cards biometrics promise to provide better security to any and every application.

1.2 Security Attacks and Goals for Authentication

A typical scheme has 4 phases, viz. registration, login, authentication and password change phases. In registration, user sends his credentials to the remote server. Certain computations are carried out on the server side and a smart card is generated. This smart card, containing values (related to user and server), is sent to the user. This completes the registration of the user. This smart card is used in the login phase whenever user needs to access services from the server. He inserts the smart card in the terminal and enters his identity and password. Once the verification of user is complete, the login request is sent to the server. Upon receiving this request, the server verifies the authenticity of that message. Some more computations are done and a session key is generated by the server. After this, an authentication request is sent to the user. This message is used by the user and the session key is verified. This completes the authentication phase. This session key is used to further communicate with the server. Whenever user needs to alter his password, he logins and after authentication, a new password is entered. This completes the password change phase.

Authentication schemes have been proposed from 1981 but right from there, most of the schemes have some kinda security weaknesses in them. Such weaknesses were pointed out and improved schemes were proposed. Tsai et al. (2006) defined the set of different attacks an authentication scheme should resist as well the goals they should satisfy. Additionally, a new set of design goals (including security requirements and attributes) have been proposed by Madhusudhan and Mittal (2012) to evaluate any authentication scheme. Till date, this set is most explicit, comprehensive and systematic set of criteria for evaluation of authentication schemes. These are explained below:

SA1 *Denial-of-Service Attack* : An attacker tries to disrupt the normal function and prevents other users from accessing the services by sending copies of fake requests thereby, making the server unable to reply to a request of a legit user.

SA2 *Impersonation Attack*: An attacker tries to alter the intercepted communications to pretend as the legal user to login the system.

SA3 *Insider Attack* : The insider of the server can perform an offline guessing attack to obtain a legal users' password.

SA4 *Parallel Session Attack*: An attacker can impersonate a legal user by creating a valid login message from the eavesdropped communication between that user and the server, without any information of users' password.

SA5 *Password Guessing Attack*: An attacker intercepts the authentication messages and stores them locally. Then tries to use a guessed password to check the correctness of the guesses using authentication messages.

SA6 *Replay Attack*: An attacker can pretend to be the legal user to login the system by intercepting the previous communications. That is, he sends the same login message again to login the system.

SA7 *Smart Card Loss Attack*: When the smart card is lost or stolen, an attacker can easily change password or can guess it using the password guessing method, or can impersonate the user to login the system.

SA8 *Stolen-verifier Attack*: The attacker can steal the hashed passwords from the server to impersonate the legal user.

SA9 *Reflection Attack*: This is a method of attacking a challenge-response authentication system that uses the same protocol in both the directions. That is, the attacker tricks the user into providing the answer to its own challenge.

An ideal authentication scheme should overcome all the above mentioned security attacks. Equally, the scheme should also achieve the following goals for it to be ideal.

G1 *User Anonymity and Untraceability*: An adversary should neither identify the users' identity from the authentication sessions, nor link the authentication sessions in which the same user is.

G2 *Mutual Authentication*: Server should authenticate the user as well the user should authenticate the server.

G3 *Session Key Agreement*: A session key should be established during the authentication process. After successful authentication, both parties will communicate using this session key, to provide confidentiality and secrecy of transmitted data.

G4 *Forward Secrecy*: This means that even if the servers' secret key is compromised by an attacker or he gets the previous session keys, he should not be able to derive the new session key.

G5 *No Verification Table*: The remote system should not keep a record of verification tables such as clear-text passwords or hashed passwords that are used to authenticate users.

G6 *No Delay in Wrong Password Detection*: If the user inputs wrong password during login phase, then the login phase has to be terminated at the smart card's side without further delay.

G7 *Smart Card Revocation*: In case of lost cards, there should be provision in the system for invalidating the further use of the lost smart card and obtaining a new smart card.

G8 *Freely Chosen Password by the User*: If the password is chosen by the server, user is left with no choice to choose a password which is not relevant in real-life applications. Secondly, password chosen by the server may be long and random making it difficult for the user to memorize. Also, it is more likely that the user forgets the password in case he does not use the system more frequently. So users must have the provision to choose their password.

G9 *Single Registration*: The user has to register only once with the server to obtain a valid secret parameters. Later, the user can use these details as authentication information to access services from the medical server.

G10 *Key Freshness*: Neither party can predetermine the shared session key being established.

G11 *Computation Efficiency* : The authentication protocol should have low computational and communication cost.

G12 *User Friendliness*: Not only the user should choose his password but also be able to update the password whenever required.

1.2.1 Security Issues in TMIS

The healthcare data contains personal information of patients, which needs to be protected in order to keep the system secure. Health information privacy is an individual's right to control the acquisition, uses or disclosures of his or her identifiable health data. Confidentiality, which is closely related, refers to the obligations of those who receive information to respect the privacy interests of those to whom the data relate. Security refers to physical, technological, or administrative safeguards or tools used to protect identifiable health data from unwarranted access or disclosure (Cohn, 2006). The system needs to efficiently support several security issues such as access, disclosure, modification, disruption, impersonation, and recording and replaying which are discussed below (Wazid et al., 2016):

1. *Access:* Legal users of the system are patients, doctors, nursing staff, researchers and pharmacists (each with his or her own access rights). Only these people can access the health data stored at the medical server. An attacker of the system always seeks to access the server illegally so that he can steal data and misuse it to achieve his malicious objective. Sometimes, they may sell the stolen patient's information to third parties which results in the patients receiving unwanted attention, related to their illness.

2. *Disclosure:* The confidentiality of the health data stored at the medical server is a major security issue. If the patient's medical record confidentiality is breached, it can have serious ramifications on the life of the patient. Private health information that is disclosed socially can cause harm to patient's reputation and personal life.

3. *Modification:* An attacker can modify the health data of patients. For example, the modified data of the patient cannot be used anymore if the patient has a high level of blood glucose value which has been modified intentionally to a low level by the attacker having unauthorized access to the data. This type of modification affects the patients as the doctor can recommend the medicine based on the low level of blood pressure.

4. *Medical Server Disruption:* An attacker can try to disrupt the services of the system by repeatedly sending bogus request messages to overload the medical server to such an extent that the server becomes too busy to reply to requests from legal users who are denied access to the system's services. A malicious user can shut off or alter the settings in the server without the knowledge of patient or doctor.

5. *Impersonation:* An attacker who tries to impersonate the legitimate user of the system can collect the health data and misguide the other users. Suppose a patient suffers from some disease and is admitted to a hospital, then the nursing staff consults the doctor regarding the medicine that should be given to the patient. If an attacker impersonates the actual doctor, he/she can misguide the nursing staff by giving the wrong prescription.

6. *Recording and Replaying:* An attacker can intercept and record the exchanged messages, and later replay them back to fool and mislead the legal users of the system. By reusing the recorded information, the attacker can later prove his identity and authenticity to the other party in order to get information such as the session key that may allow him to communicate with the legal users of the system.

In all the above cases, the results can be unexpectedly disastrous. So it has to be seen that data has to be passed safely. This transmitted and stored data should be kept confidential. Preserving user anonymity is very much necessary, specially in medical field. Also, the user should be assured that he/she is communicating with the right doctor on the other side. Equally, data inside the server has to be taken care of to avoid its manipulation. Hence, to ensure legality of patients and protect medical servers/resources from being damaged or accessed by unauthorized or illegal users, remote user authentication plays a crucial role in TMIS (Das, 2015).

As mentioned earlier, authentication is nothing but a method for verifying the identities of remote users in TMIS before they can access a service. For this, only password authentication is not suitable. Usage of smart cards combined with passwords can only provide two-factor authentication. With biometrics, the problem arises when a registered patient won't be able to login the server due to unavoidable conditions (like differently abled persons or those who have had serious accidents). Considering these aspects in health care systems, it can be observed that the best option in is a combination of all. That is, providing three-factor authentication using passwords, smart cards and biometrics can make a sensible statement.

1.3 Threat Model

An adversary is a malicious entity who prevents the users in a network from achieving privacy, integrity and availability of data. When an adversary gets access to any system in a network, it poses a threat to the legitimate users of that system. Hence, while analyzing authentication schemes, it is necessary to keep in mind the ways in which an adversary can enter a network and make the system insecure. In our study, we consider the most recognized and utilized Dolev and Yao (1983) adversary model, where the adversary can stand between user and server. This means the adversary has total control over the communication channel, which connects the remote user and medical server. As a result, he can intercept, insert, delete or modify any message transmitted over that channel. Also, we assume that an adversary can extract the information stored in a user's smart card by analyzing the power consumption of the smart card (Kocher et al., 1999; Messerges et al., 2002). This is not an unrealistic assumption since it is very much possible for an adversary to obtain a lost smart card. These play an important role in cryptanalysis and will be used throughout this report.

1.4 Burrows-Abadi-Needham (BAN) logic

BAN logic is a model used to formally verify an authentication scheme. In this model, various postulates and assumptions are used. Using these postulates, it is possible to prove that both the participants (user and server) believe each other on the freshness and authenticity of the session key generated during a session. During every session, a session key is generated and using this key, both the parties communicate with each other. In other words, this key is used for encrypting the messages transmitted after mutual authentication. So, assuring the safety of a session key plays a major role in providing secure communication between user and server. This safety of session key is assured using BAN logic and the same is used throughout this study for proof.

For verification, the following notations and constructs as given by Burrows et al. (1989) are used:

1. P believes X $(P \mid\equiv X)$: Principal P acts as though X is true.

2. P sees X $(P \triangleleft X)$: Someone has sent a message containing X to P, who can read and repeat X.

3. P said X $(P \mid\sim X)$: Principal P at some time sent a message including the statement X.

4. P controls X $(P \mid\Rightarrow X)$: P has jurisdiction over X meaning P is an authority on X and should be trusted on this matter.

5. fresh(X) $(\#(X))$: Formula X is fresh meaning that X has not been sent in a message at any time before the current run of the protocol. This is usually true for nonces.

6. $P \overset{K}{\leftrightarrow} Q$: P and Q may use the shared key K to communicate. The key K will never be discovered by any principal except P or Q, or a principal trusted by either P or Q.

7. $\overset{K}{\mid\!\rightarrow} P$: P has K as a public key. The matching secret key K^{-1} will never be discovered by any principal except P or a principal trusted by P.

8. $P \overset{X}{\rightleftharpoons} Q$: Formula X is a secret known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities.

9. $\{X\}_K$: Formula X is encrypted under the key K.

10. $\langle X \rangle_Y$: Formula X is combined with the formula Y, it is intended that Y be a secret and that its presence proves the identity of whoever utters $\langle X \rangle_Y$.

The protocol analysis along with the above constructs uses the following logical postulates in proving the identity of the parties involved.

1. The *message-meaning* rule concerns the interpretation of messages. They all explain how to derive beliefs about the origin of messages.
   For shared keys,
   $$\frac{P\ believes\ P \overset{K}{\leftrightarrow} Q,\ P\ sees\ \{X\}_K}{P\ believes\ Q\ said\ X}$$
   For public keys,
   $$\frac{P\ believes\ P \overset{K}{\leftrightarrow} Q,\ P\ sees\ \{X\}_{K^{-1}}}{P\ believes\ Q\ said\ X}$$
   For shared secrets,
   $$\frac{P\ believes\ P \overset{X}{\rightleftharpoons} Q,\ P\ sees\ \langle X \rangle_Y}{P\ believes\ Q\ said\ X}$$

2. The *nonce-verification* rule expresses the check that a message is recent and hence, the sender still believes in it.
   $$\frac{P\ believes\ \#(X),\ P\ believes\ Q\ said\ X}{P\ believes\ Q\ believes\ X}$$

3. The *jurisdiction* rule states that if P believes that Q has jurisdiction over X, then P trusts Q on the truth of X.
   $$\frac{P\ believes\ Q\ controls\ X,\ P\ believes\ Q\ believes\ X}{P\ believes\ X}$$

4. The *freshness* rule indicates that if one part of formula is fresh, then the entire formula is fresh.
   $$\frac{P\ believes\ \#(X)}{P\ believes\ \#(X,Y)}$$

5. If a principal sees a formula, then he also sees its components, provided he knows the necessary keys.
   $$\frac{P\ sees\ (X,Y)}{P\ sees\ X}$$

$$\frac{P\ believes\ \langle X \rangle_Y}{P\ sees\ X}$$

$$\frac{P\ believes\ P \overset{K}{\leftrightarrow} Q,\ P\ sees\ \langle X \rangle_K}{P\ sees\ X}$$

$$\frac{P\ believes\ \ |\overset{K}{\rightarrow} P,\ P\ sees\ \langle X \rangle_K}{P\ believes\ sees\ X}$$

$$\frac{P\ believes\ \ |\overset{K}{\rightarrow} P,\ P\ sees\ \langle X \rangle_{K^{-1}}}{P\ believes\ sees\ X}$$

1.5 Research Objectives

1.5.1 Research Gaps

We have studied a good number of authentication schemes proposed for TMIS. Observation of existing authentication schemes point out many security weaknesses like they are vulnerable to different attacks including password guessing, replay attacks, impersonation attack, insider attack, denial-of-service attack etc.. Out of these attacks, most of the schemes fail to provide user anonymity and are vulnerable to impersonation attacks implying they lack in providing security. On the other hand, others require high computations. Hence, they are not suitable for practical implementation. So, the schemes need to be more efficient for practical implementation. Hence, there is a need to propose light weight, robust and practically applicable authentication scheme for TMIS.

1.5.2 Objectives

1. To propose an efficient, enhanced or new authentication scheme for Telecare Medicine Information System.

2. Cryptanalyze the proposed scheme to verify that it overcomes all the mentioned attacks.

3. Verify the correctness of the proposed scheme.

4. Compute efficiency of proposed scheme and compare the obtained results with the existing schemes to demonstrate the practical applicability.

1.5.3 Contributions of the thesis

Main contributions of this work are stated below. Many authentication schemes TMIS are studied and analyzed.

1. Studied the evolution of authentication from static to dynamic identity schemes.
2. Conducted a survey on choice of user password of certain websites and password retrieval methods. Various guidelines for password practices are given.
3. Cryptanalyzed Li et al.'s scheme (chaotic-map based) and identified weaknesses. These have been addressed in the proposed scheme. Security proof using BAN logic is also provided which ensures the safety of session key. Performance comparison is tabulated.
4. Hash functions based authentication scheme proposed by Chen et al. is studied and various flaws have been detected. The proposed scheme provides security against these flaws as can be seen from the security analysis. Computational cost comparison is presented to verify the robustness of the proposed scheme.
5. A three factor authentication scheme is proposed in order to rectify the limitations that are found in Jung et al.'s scheme. Security analysis is discussed to highlight the security provided in the proposed scheme. To support this, overall comparison is also presented,
6. Han et al.'s biometric-based scheme is thoroughly analyzed and the security attacks identified are discussed. To overcome these analysis, a new biometric-based scheme is designed. Security proof using BAN logic is explained. Additionally, the proposed scheme is compared with existing schemes in terms of computations and performance.

1.6 Organization of the thesis

This thesis is organized as follows. Chapter 1 provides a detailed introduction to TMIS. The security issues in medical field are explained in detail. Threat model, security attacks and security goals for user authentication have been described. Also, objectives of this study have been presented.

Chapter 2 is broadly divided into two parts. The former part explains the evolution of authentication schemes starting from one-factor authentication. The latter part of this chapter discusses the password practices and password retrieval methods of certain websites. In addition to this, mathematical preliminaries required to understand and analyze these schemes are briefly explained.

In chapter 3, a chaotic-map based authentication scheme has been cryptanalyzed and the possible security attacks have been explained. To overcome these issues, an improved scheme has been proposed. Security analysis of the proposed scheme is discussed. Also, the performance comparison with existing schemes is demonstrated. A two-factor scheme proposed by Chen et al. is discussed in chapter 4. Starting with the cryptanalysis, the chapter further continues to explain the security flaws in their scheme. Also, the proposed scheme has been explained that overcomes the mentioned flaws. Then, security proof is provided to support the argument. Following this, the performance comparison is presented which gives the overall picture of computational cost, estimated execution time and security properties. In chapter 5, two biometrics-based authentication schemes have been studied. The first one is proposed by Jung et al. and the other one by Han et al.. This is followed by the details of their security weaknesses. These issues have been overcome in the respective proposed schemes which are explained in depth. Security proof and comparison of the proposed schemes along with existing schemes have been presented.

Chapter 6 gives a bird's-eye view of the thesis and presents ideas for future research directions.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we shed light on the available literature in the field of authentication. This chapter is broadly divided into two parts. The first part discusses the evolution of authentication schemes. Through this, certain flaws were identified in password authentication. To have a better understanding of this, a survey was conducted which considered issues like current password practices and security questions used by several websites. It also throws light on password retrieval methods of those sites. These topics are discussed in the second part of this chapter. These results have been tabulated in Madhusudhan and Nayak (2018). In addition to this, basic definitions of certain operations used in the authentication schemes are also explained, which are useful in understanding the computations that follow.

2.1 Authentication schemes in TMIS

To ensure the privacy of patients and to allow authorized access to remote medical servers, many authentication schemes have been proposed. The registered user who could be medical academic institutes, from large hospitals or from private clinics and even an individual patient, can request all the services from the medical server whenever required. Once they complete the verification process, they can have access to the necessary information. During these data exchange, a lot of private and highly confidential information will be transmitted over public channels. Security of data is very important as doctors use the information like electrocardiogram, magnetic resonance imaging, treatment record of the disease etc. to diagnose and treat the patient.

Lamport (1981) first introduced the password-based authentication scheme which was a one factor authentication scheme. But then it was vulnerable to replay attack. To overcome this, Haller (1994) introduced the concept of session key in which only a single use of password ever crosses the network at any time. Using this concept, various authentication schemes were proposed. Tzu et al. (2002) proposed an OTP based scheme using smart cards. But, Lee and Chen (2005) pointed out security gaps in Tzu et al. (2002) and improved the scheme. You and Jung (2006) recognized weaknesses in

Lee and Chen (2005) and proposed an improved scheme. But the traditional schemes, Lamport (1981); Haller (1994) maintained a password table in the server making the scheme vulnerable to various password attacks.

Wu et al. (2012b) proposed an efficient authentication scheme for TMIS using Discrete Logarithm Problem (DLP) and smart cards. But then, Debiao et al. (2012) showed that Wu et al.'s scheme suffered from impersonation attack and insider attack after which they proposed an improved scheme. In the same year, Wei et al. (2012) demonstrated that Debiao et al.'s scheme could not provide two-factor security and proposed an improved scheme. But then, Zhu (2012) showed that Wei et al.'s scheme could not resist offline password guessing attack and he proposed RSA-based authentication scheme for TMIS. In these schemes, the identity of the user was static. With passing time, dynamic-identity-based authentication schemes were designed.

In 2012, Wu et al. (2012a) proposed an authentication scheme for the integrated EMR systems using lightweight hash functions. Lee et al. (2013) proved that Wu et al.'s scheme was not resistant to smart card loss and stolen-verifier attacks; who in turn gave the improved scheme. But then, Wen (2014) pointed out that the scheme in Lee et al. (2013) could not be protected from off-line password guessing attack and he improved the same. Various authentication schemes were proposed in 2013 and 2014 (Jiang et al., 2013; Wu and Xu, 2013; Xie et al., 2013; Chaturvedi et al., 2013; Wen and Guo, 2014).

In 2015, the scheme proposed by Zhu (2012) was analyzed by Arya and Vidwansh (2015), who showed that validity of the user's login request message was incorrect in Zhu's scheme. So, they proposed a dynamic authentication scheme to rectify that flaw. But, Kang et al. (2017) showed that scheme in Arya and Vidwansh (2015) has weaknesses like no user anonymity and was vulnerable to offline password guessing attack, user impersonation attack and session key derived attack. So, they proposed an authentication scheme for TMIS. Their scheme was thoroughly cryptanalyzed in Chen et al. (2018). They proved that the scheme in Kang et al. (2017) was not secure against password guessing attack, does not verify password and does not preserve user anonymity. So, they proposed an improved scheme. After cryptanalysis, Madhusudhan and Nayak (2018) showed Chen et al.'s scheme had weaknesses viz. user impersonation, password guessing, server impersonation, no user anonymity. These issues were addressed in their improved scheme in Madhusudhan and Nayak (2018).

The scheme proposed by Wen (2014) was studied by Das (2015) who showed that Wen's scheme had defects in password change phase and showed it was exposed to privileged insider attack. Then he improved Wen's scheme while maintaining the original idea of it, stating that his scheme is resistant to known possible attacks. But, Mir et al. (2015) pointed out that the scheme in Das (2015) could not resist password guessing attack and they improved the scheme. Additionally, Li et al. (2015) proved that Das (2015) was vulnerable to modification and user duplication attacks. They proposed an enhanced authentication scheme. But, Jung et al. (2017) proved that the scheme in Li et al. (2015) was unprotected against password guessing attack, does not verify password and does not preserve user anonymity. So, they proposed an improved scheme.

Li et al. (2016) proposed an authentication scheme based on chaotic maps for e-healthcare systems. But, Madhusudhan and Nayak (2019) proved that the scheme in Li et al. (2016) was vulnerable to impersonation attacks and password guessing attack; the scheme provided no security for user anonymity, session key and smart card revocation was inconvenient. So, a robust authentication scheme was proposed by them. Other than this, several two-factor schemes (Liu et al., 2016; Sutrala et al., 2016; Chaturvedi et al., 2017; Irshad et al., 2017; Xiong et al., 2017; Chang et al., 2017; Li et al., 2018; Qiu et al., 2018) and three-factor and chaotic-map based schemes (Chen et al., 2012; Yeh et al., 2013; Khan et al., 2014; Wu et al., 2015; Lu et al., 2015a; Siddiqui et al., 2016; Jiang et al., 2018) were proposed for TMIS providing enhanced security.

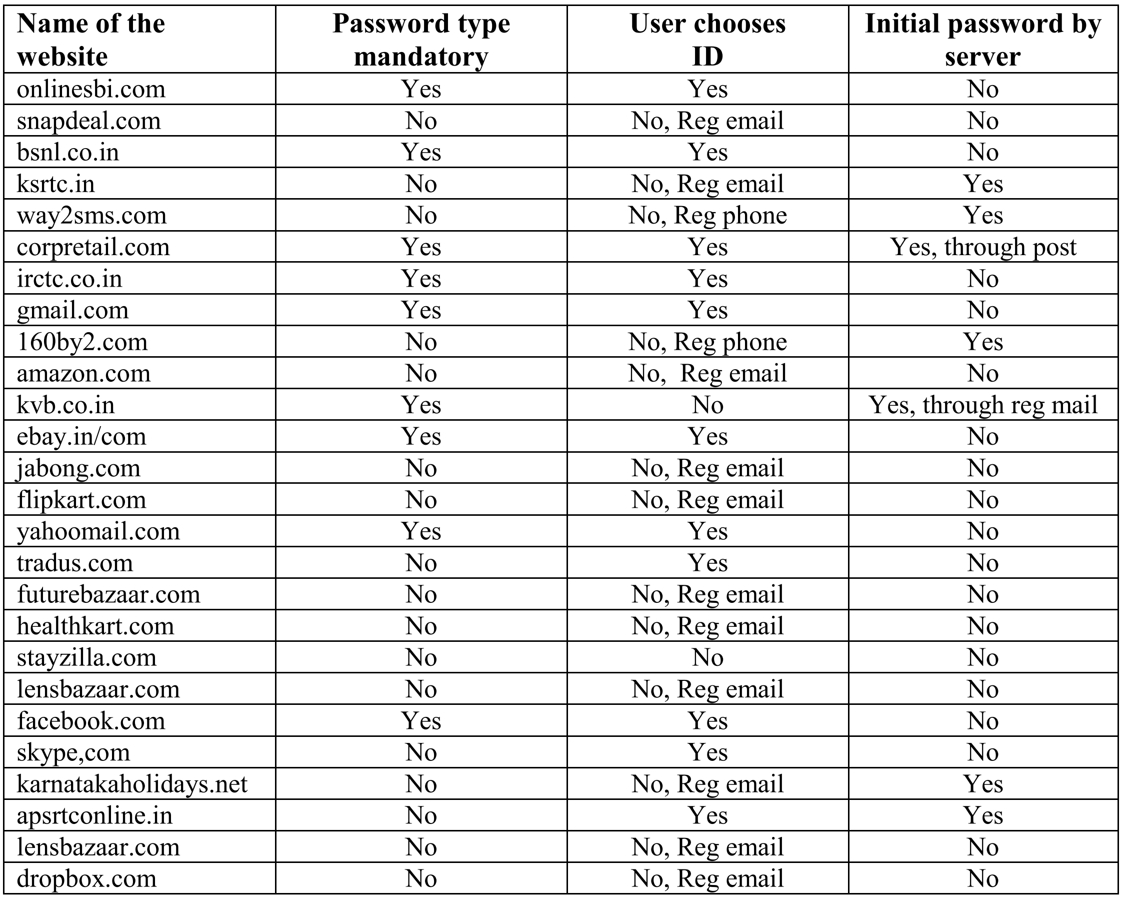
Awasthi and Srivastava (2013) proposed a biometrics-based authentication scheme using a nonce for TMIS.Mishra et al. (2014) cryptanalyzed their scheme and mentioned that it had inefficient password change phase and could not overcome password guessing attack. In addition to this, Tan also studied Awasthi et al.'s scheme and pointed out that it failed to withstand reflection attack and did not preserve user anonymity. So, he proposed a scheme overcoming the mentioned flaws (Tan, 2014). But, Arshad and Nikooghadam (2014) cryptanalyzed Tans' scheme and claimed that it did not resist replay and denial-of-service attacks; they proposed an improved scheme using ECC for TMIS. However, in 2015, their scheme was analyzed by Lu et al. who revealed that it was vulnerable to offline password guessing and user impersonation attacks. To overcome these flaws, they proposed an improved scheme with fewer computations (Lu et al., 2015b). But, Han et al. (2018) reviewed their scheme and claimed that it could not resist user and server impersonation attacks and could not preserve user anonymity. Then they proposed a new scheme. Very recently, Madhusudhan and Nayak (2020) found several weaknesses in Han et al.'s scheme and an improved scheme was proposed.

2.2 Assessment of website authentication mechanisms

To get a clear picture of the security flaws, we made a survey on password practices including the password type, ways of retrieval of passwords and the restrictions on passwords of 26 different websites. Besides the bad practices of users, we found that there are many websites, which do not restrict the users to choose secure passwords and do not give guidelines for choosing a strong password. So, a few guidelines for strong password practices have been given.

2.2.1 Comparison of websites based on password practices

In this, we have done a survey on different websites, which includes different types like social networking, e-commerce, online banking, mail service, etc. and the results are shown in Fig 2.1. We have considered issues like registration of new user and choice of passwords. We have also made a survey on the password retrieval methods (in case of forgotten passwords) of various websites and restrictions in choosing the password. The results are presented in Fig 2.2.



Name of the website	Password type mandatory	User chooses ID	Initial password by server
onlinesbi.com	Yes	Yes	No
snapdeal.com	No	No, Reg email	No
bsnl.co.in	Yes	Yes	No
ksrtc.in	No	No, Reg email	Yes
way2sms.com	No	No, Reg phone	Yes
corpretail.com	Yes	Yes	Yes, through post
irctc.co.in	Yes	Yes	No
gmail.com	Yes	Yes	No
160by2.com	No	No, Reg phone	Yes
amazon.com	No	No, Reg email	No
kvb.co.in	Yes	No	Yes, through reg mail
ebay.in/com	Yes	Yes	No
jabong.com	No	No, Reg email	No
flipkart.com	No	No, Reg email	No
yahoomail.com	Yes	Yes	No
tradus.com	No	Yes	No
futurebazaar.com	No	No, Reg email	No
healthkart.com	No	No, Reg email	No
stayzilla.com	No	No	No
lensbazaar.com	No	No, Reg email	No
facebook.com	Yes	Yes	No
skype,com	No	Yes	No
karnatakaholidays.net	No	No, Reg email	Yes
apsrtconline.in	No	Yes	Yes
lensbazaar.com	No	No, Reg email	No
dropbox.com	No	No, Reg email	No

Figure 2.1 Survey on password choice of websites

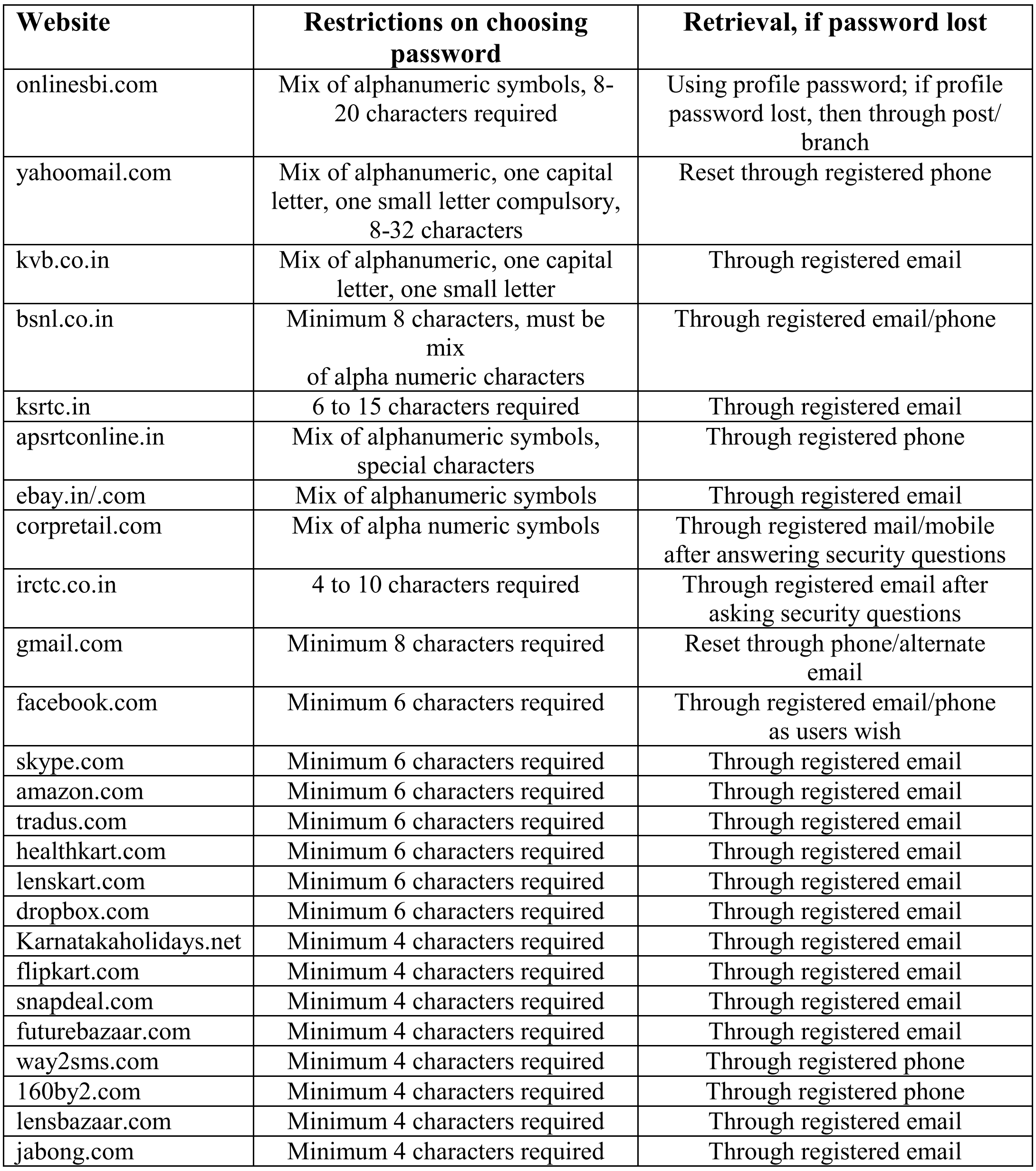
From Fig 2.1, it is evident that most of the banking sites allow user to choose his/her own username but password is sent in a different manner. Almost all the e-commerce websites make use of the registered email as his/her user id and ask to choose a password in the registration process. But some websites like ticket booking sites are sending initial password to the registered mail or mobile number and later they allow the user to change the initial password. Almost all the websites offers SMS service to send the initial password to the mobile for the sake of verification of mobile number. This suggests that most of the sites let the user choose the password without imposing much conditions.

The websites that offer SMS service sends the new password or the old password to the registered mobile number. When it comes to password retrieval phase, most of the websites system of password retrieval is through registered email as can be observed from Fig 2.2. Some popular websites like irctc, Facebook and eBay ask for security questions during password retrieval phase to authenticate the user.

There are several websites that monitor the password and show status of the password like weak, medium or strong. At the same time, sites (like e-commerce sites in our survey) do not restrict the pattern of password. Upon that, even the restriction on length of passwords is considerably low. The leakage of passwords from three major websites, LinkedIn, eHarmony and Last.fm focused on the threats of weak password practices (Identifiable, 2012). Besides the bad practices of users, we found that there are many websites, does not give guidelines for choosing a strong password. In addition to this, a survey revealed that the most common passwords used were password, 123456, 12345678, abc123, qwerty etc. (Doel, 2013). This suggests that users still have significant lessons to learn regarding effective choice of passwords. In some cases, they are ignorant about the threats they might have to face while accessing online services due to weak passwords. Choosing such passwords may be either because users are aware of the challenges but find it difficult to memorize complicated passwords or they have zero knowledge about possible danger they are in due to weak passwords. Hence, it is necessary for security departments to guide users regarding security issues and password selection.

2.2.2 Security questions used for identifying authentic user

It often happens that user forgets the password due to various reasons. In such situations, the websites should help the user in retrieval of password and/or provide options for the user to choose a new password; making sure that only the legal user is provided with such an option. Usually, this is done through registered email or with the help of users' mobile number or by asking the so called security questions. Most of the web-



Website	Restrictions on choosing password	Retrieval, if password lost
onlinesbi.com	Mix of alphanumeric symbols, 8-20 characters required	Using profile password; if profile password lost, then through post/ branch
yahoomail.com	Mix of alphanumeric, one capital letter, one small letter compulsory, 8-32 characters	Reset through registered phone
kvb.co.in	Mix of alphanumeric, one capital letter, one small letter	Through registered email
bsnl.co.in	Minimum 8 characters, must be mix of alpha numeric characters	Through registered email/phone
ksrtc.in	6 to 15 characters required	Through registered email
apsrtconline.in	Mix of alphanumeric symbols, special characters	Through registered phone
ebay.in/.com	Mix of alphanumeric symbols	Through registered email
corpretail.com	Mix of alpha numeric symbols	Through registered mail/mobile after answering security questions
irctc.co.in	4 to 10 characters required	Through registered email after asking security questions
gmail.com	Minimum 8 characters required	Reset through phone/alternate email
facebook.com	Minimum 6 characters required	Through registered email/phone as users wish
skype.com	Minimum 6 characters required	Through registered email
amazon.com	Minimum 6 characters required	Through registered email
tradus.com	Minimum 6 characters required	Through registered email
healthkart.com	Minimum 6 characters required	Through registered email
lenskart.com	Minimum 6 characters required	Through registered email
dropbox.com	Minimum 6 characters required	Through registered email
Karnatakaholidays.net	Minimum 4 characters required	Through registered email
flipkart.com	Minimum 4 characters required	Through registered email
snapdeal.com	Minimum 4 characters required	Through registered email
futurebazaar.com	Minimum 4 characters required	Through registered email
way2sms.com	Minimum 4 characters required	Through registered phone
160by2.com	Minimum 4 characters required	Through registered phone
lensbazaar.com	Minimum 4 characters required	Through registered email
jabong.com	Minimum 4 characters required	Through registered email

Figure 2.2 Survey of password practices and retrieval methods

sites go for security questions. If the user enters the correct answers (initially saved in the server during registration), it means he/she is the legal user and is eligible to choose a new password. But, it is not necessary that only the legal user must produce correct answers. Most of the times, the questions will be in such a manner that a close friend or spouse can guess the answers correctly with less or zero effort. This raises a question as to whether such questions can really be called as secure questions! Unfortunately, the answer is no. The common security questions that eBay, yahoo, etc., use are stated below:

- What street did you grow up on?

- What is your mothers maiden name?
- What is the name of your first school?
- What is your fathers birthplace?
- Who is your best childhood friend?
- Who was your childhood hero?
- What is your all-time favorite sports team?
- Where did you first meet your spouse?
- When was your first child born?
- What is the registration number of your first vehicle ?
- What is your pets name ?
- What is your fathers middle name ?
- What is your favorite pastime ?
- Who is your youngest cousin ?

Some websites use these type of security questions in password retrieval phase. Such questions can be easily answered by family members or friends. Then where comes the security when people other than the legitimate user can answer the questions accurately? According to Schechter et al. (2009), 17% of its participants were able to answer the 'secret questions' of strangers and also indicated that the most popular questions were in fact the easiest ones to answer. Hence, the best practice of choosing the question is to choose them in such a way that the answer should not be related to public information and should not be known by spouse/close friends etc. Also, the question should be framed in such a way that there should not be any ambiguity for the user to answer.

By observing the survey results, it is clear that the websites that offer banking services take maximum care in matters of password restrictions. Unlike other websites, Facebook restricts its users from reusing any of their old passwords. Many sites do advise and prescribe what should be done without really explaining why. Hence, the user fails to understand the need for a strong password.

2.2.3 Guidelines for choosing password

It is common tendency that human being always tries to do things in a simple fashion without giving much work to his brain. This is true in case of choice of passwords as well. In order to make it more simple and memorable, the user chooses passwords which are easy for him to remember but can be cracked by an adversary with less effort. As a result, a number of online problems like password theft, guessing attacks, denial-of-service attacks, stolen-verifier attacks etc. happen. But unfortunately most of the users are totally unaware of these attacks. They have least idea that the mentioned attacks can happen to their accounts as well! As a result, security measures are usually overlooked. Users are never motivated to behave in a secure manner. So, making users aware about these matters is of at most importance

The best password practice of choosing passwords is a combination of alphabets, numbers and special symbols. Only the online banking sites and popular sites are making mandatory rules for choosing passwords, while many other sites have no strong restrictions on the style of the password. A few rules that can be followed by any user are given below (Gehringer, 2002):

- Should not contain name.
- Should contain one or more numbers.
- Should contain at least 8 characters.
- Must be different from previously used passwords.
- Use of virtual keyboard to enter credentials.
- Regular change of password.
- Different passwords for multiple accounts.
- Change of password in case shared.

Other than this, user can use some techniques that will help him remember a password. For example, in the phrase “The entire world is termed as a global village recently”, the first alphabets or last alphabets of every word, “Tewitaagvr” or "eedsdsaleY" can become passwords. To make them complex, the user simply needs to add special characters or numbers in between the letters. In addition to this, users must be guided properly in choosing secure password during the registration phase itself, and most importantly, awareness must be created among them regarding the necessity of such measures. Systems must spread the word among its users about various online attacks so that users can take such matters more seriously. Changing passwords at regular intervals of time must be made mandatory at least for websites containing confidential data.

Based on these observations, it can be inferred that only password authentication (one-factor authentication) cannot be relied upon for authentication purposes. To overcome this, smart cards and biometrics are highly used for authentication. So, a study has been carried out on the same.

2.3 Mathematical Preliminaries

The basic topics required to understand authentication schemes are briefly explained. These concepts will be used throughout.

2.3.1 XOR (eXclusive-OR) operation

This is the most commonly used operation to design an authentication scheme. A simple XOR operation is defined as follows:

1. $A \oplus A = 0$

2. $A \oplus 0 = A$

3. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$ (Associative property)

4. $(A \oplus B) \oplus B = A \oplus (B \oplus B) = A \oplus 0 = A$

2.3.2 Hash function

A one-way cryptographic hash function, $h : \{0,\ 1\}^* \rightarrow \{0,\ 1\}^*$ takes a string $q\ \varepsilon\ \{0,\ 1\}^*$ of any arbitrary length as an input and produces a string of fixed length, say *n bits*, $h(q)\varepsilon\{0,\ 1\}^*$ as output (Stallings, 2006). These functions satisfy the following properties (Preneel, 1993):

1. For a given h and x, computation of $h(x)$ is *easy*.

2. For a given y, it is *computationally infeasible* to obtain x such that $h(x) = y$.

3. For a given x and $h(x)$, it is *computationally infeasible* to find $x^{/} \neq x$ such that $h(x^{/}) = h(x)$.

2.3.2.1 Biohashing

This is a process of randomization and binarization in which secure templates are generated in the form of a set of non-invertible binary strings from the biometrics produced by the user with the help of user-specific random numbers. The output is a binary string, commonly known as *Biohash* (Zhou and Kalker, 2010).

2.3.3 Chaotic maps

These maps produce required confusion and diffusion. It is a map which exhibits some sort of random behavior. They are unstable dynamical systems with high sensitivity to initial conditions (Devaney et al., 1993). In other words, a negligible change in the input makes a huge difference in the resulting output. Properties of chaotic maps like ergodicity and sensitive dependence on initial conditions and system parameters are quite advantageous to construct secure communication schemes, where irregularity in code sequences, sensitive dependence on plain texts and keys are required (Masuda and Aihara, 2002). One of the known maps, Chebyshev polynomial is used in this study. The definition and basic properties are mentioned below:

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

$$T_n(x) = \begin{cases} \cos(n.\arccos(x)) & \text{if } x \; \varepsilon \; [-1,1] \\ \cos(n\theta) & \text{if } x = \cos\theta, \theta \; \varepsilon \; [0,\pi] \end{cases} \tag{2.3.1}$$

  The recurrence relation of the polynomial in (1) is given by

$$T_n(x) = \begin{cases} 1 & \text{if } n \;=\; 0 \\ x & \text{if } n \;=\; 1 \\ 2xT_{n-1}(x) - T_{n-2}(x) & \text{if } n \geq 2 \end{cases} \tag{2.3.2}$$

- The semi-group property of the enhanced Chebyshev polynomial holds on the interval $(-\infty,+\infty)$ and is defined as follows
  For $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2$ where p is a large prime and $x \; \varepsilon \; (-\infty,-\infty)$, $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x))(mod \; p)$ always holds where $r,s \; \varepsilon \; Z_p^* = \{a|0 < a < p, \; gcd(a,p) = 1\} = \{1,2,...,p-1\}$.

- For any given x and y, it is computationally infeasible to find integer s such that

$T_s(x) = y$. This property is referred to as the chaotic map-based discrete logarithm problem (CMDLP).

2.3.4 Elliptic curve (Islam and Biswas, 2013)

- Let p be a large prime and E/F_p be a set of elliptic curve points over a prime field F_p, defined by the non-singular elliptic curve

$$y^2 \ (mod \ p) = (x^3 + ax + b) \ mod \ p \tag{2.3.3}$$

  where $x, \ y, \ a, \ b \ \varepsilon \ F_p$ and $4a^3 + 27b^2 \ mod \ p \neq 0$.
  A point $P(x,y)$ is an elliptic curve point if it satisfies 2.3.3.

- Addition of two points P and Q is defined to be the mirror image of the point of the line of intersection of P and Q.

- The scalar point multiplication of a point P on the elliptic curve is defined as $kP = P + P + \cdots + P(k \, times)$.

- A generator point $P \ \varepsilon \ E/F_p$ has order n if n is the smallest positive integer and $nP = O$ where O is called *point at infinity*.

2.4 Estimated execution time

Computation of estimated execution time is used as a factor for performance comparison of proposed schemes. This computation is based on an experiment conducted by Kocarev and Lian (2011). The experiment was conducted on an Intel Pentium4 2600 MHz processor with 1024MB RAM. According to their study, $T_h = 0.0005s$, $T_{ch} = 0.02102s$, $T_{ed} = 0.0087s$ and $T_{pm} = 0.063075s$, where T_h indicates the time required for execution of one-way hash operation, T_{ch} for computing $T_n(x)(mod \ p)$ in Chebyshev chaotic map, T_{ed} for execution of symmetric key encryption/decryption and T_{pm} for executing elliptic curve point multiplication where n and p are 1024 bits long. Moreover, compared to these operations, computational cost of XOR operation can be ignored. These values have been used to compute the estimated execution time of authentication schemes.

In addition to this, other notations are also used for various computations. These are given in the following table and will be extensively used hereafter.

Table 2.1 Notations and symbols used

Symbol	Meaning
U_i	i^{th} user
$S/S_i/S_j$	TMIS server
ID_i	Identity of the user U_i
PW_i	Password of the user U_i
B_i	Biometrics of U_i
d/x	S's private key(Li et al's/proposed scheme)
h(.)	Secure collision-free hash function
B(.)	Biohashing function
$T_d(.)$	Chebyshev chaotic map
E_k	Symmetric key encryption with key k
D_k	Symmetric key decryption with key k
SK	Established session key between S and U_i
$\|\|$	Concatenation operation
$\oplus$	XOR operation
P	Base point of the chosen elliptic curve, E
T_i	Time stamp

CHAPTER 3

ANALYSIS OF CHAOTIC-MAP BASED AUTHENTICATION SCHEME

To design an authentication scheme, various operations are used. Most common ones are XOR, concatenation, hashing etc.. An addition to this list is using chaotic map, the reason being its random behavior. Hence, we have cryptanalyzed a scheme proposed by Li et al. (2016) for e-healthcare systems. Their scheme uses Chebyshev polynomial (as mentioned in section (2.3.1)). They claim that their scheme prevents illegal intrusions by quick detection of wrong inputs and that the future use of a lost or stolen smart card can be invalidated. Also, they state that their scheme ensures anonymous user interaction, resists privileged insider attack, efficiently identifies the correctness of user inputs, provides protection against lost/stolen smart card, resists offline password guessing attack, is secure from participation impersonation, ensures perfect forward secrecy and mutual authentication. By cryptanalysis of their scheme, we have found certain flaws like no user anonymity, prone to user & server impersonation, password guessing attacks and has inconvenient smart card revocation. To overcome these flaws, an enhanced authentication scheme is proposed using Chebhyshev polynomial. To prove this, security analysis is provided along with security proof using BAN logic. Following this, the proposed scheme is compared with existing schemes in terms of computational cost, estimated execution time as well as security properties. All these topics constitute this chapter. The notations used are as given in Table 2.1.

3.1 Review of Li et al.'s scheme

This section presents the details of the scheme proposed by Li et al. (2016). All the phases are explained in detail.

1. Registration Phase

   To access services from the telecare medicine information system server S, a new user U_i must register himself/herself at the server S. The following steps are performed during registration:

R1. U_i initially selects his/her identity ID_i, a password PW_i and a random number b. U_i computes the masked password $W = h(PW_i \parallel b)$ and sends the registration message $\{ID_i,\ W\}$ to the server S via a secure channel.

R2. On receiving the registration request message $\{ID_i,\ W\}$ from the user U_i, the server S validates the identity ID_i of U_i. If it is valid, the server S calculates $T_d(ID_i \parallel Q)$ and computes $v = W \oplus T_d(ID_i \parallel Q)$. It then stores ID_i and Q in its database. If it is U_i's initial registration, S sets $Q = 0$ in the field of registration times for U_i; else S sets $Q = Q + 1$.

R3. S then issues a smart card containing the information $\{v,\ h(.),\ Q\}$ to the user U_i via a secure channel.

R4. U_i calculates $T_d(ID_i \parallel Q) = v \oplus W$, $X = b \oplus h(ID_i \parallel PW_i) \oplus h(ID_i \parallel PW_i)$, $Y = h(T_d(ID_i \parallel Q) \parallel b \parallel h(ID_i \parallel PW_i) \oplus h(ID_i \parallel PW_i)$ and stores X and Y in the smart card.

2. Login Phase

Whenever a registered user wants to login to the TMIS system server S, the following steps will be executed:

L1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i and password PW_i. The smart card of U_i computes $h(ID_i \parallel PW_i)$, $b = X \oplus h(ID_i \parallel PW_i)$, $W = h(PW_i \parallel b)$ and $T_d(ID_i \parallel Q) = v \oplus W$. It then verifies if $h(T_d(ID_i \parallel Q) \parallel b \parallel h(ID_i \parallel PW_i)) = Y$ holds or not. If it holds, it goes to next step. Otherwise, the login session is terminated.

L2. U_i's smart card generates a random number, a and computes $K_{US} = T_a T_d(ID_i \parallel Q)$, $R = h(ID_i \parallel T_a(ID_i \parallel Q) \parallel T_d(ID_i \parallel Q))$ and $V = K_{US}(ID_i \parallel R)$.

L3. Finally, U_i sends the login request message $\{T_a(ID_i \parallel Q),\ V\}$ through a public channel to the server S.

3. Authentication Phase

On receiving the login request message $\{T_a(ID_i \parallel Q),\ V\}$ from U_i, server S performs the following steps:

A1. S calculates $K^l_{US} = T_d T_a(ID_i \parallel Q)$, verifies the validity of ID_i and Q. It then verifies if $h(ID_i \parallel T_a(ID_i \parallel Q) \parallel T_d(ID_i \parallel Q)) = R$ holds or not. If it does not hold, S rejects the service request message and the authentication phase is terminated.

A2. S computes session key $SK = h(ID_i \parallel T_d(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q))$ and $Z = h(ID_i \parallel SK \parallel T_d(ID_i \parallel Q))$ and sends the authentication request message $\{Z\}$ to the user U_i.

A3. On receiving the authentication request message $\{Z\}$ from S, U_i computes $SK^l = h(ID_i \parallel T_d(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q))$ to check if the condition $h(ID_i \parallel SK^l \parallel T_d(ID_i \parallel Q)) = Z$ holds or not. If the condition holds, the server S is authenticated.

After successful authentication, the session key $SK = h(ID_i \parallel T_d(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q)) = SK^l$ provides a secure channel for S and U_i to communicate with each other.

4. Password Change Phase

Suppose a user U_i wishes to change his/her password, the following steps are performed:

P1. U_i inserts his/her smart card into the card reader terminal, enters his/her identity ID_i and old password PW_i. The smart card computes $h(ID_i \parallel PW_i)$, $b = X \oplus h(ID_i \parallel PW_i)$, $W = h(PW_i \parallel b)$, $T_d(ID_i \parallel Q) = v \oplus W$ and verifies if $h(T_d(ID_i \parallel Q) \parallel b \parallel h(ID_i \parallel PW_i)$ equals Y or not. If the verification holds, U_i enters a new password PW_{new}, else the request is denied.

P2. The smart card computes $W_{new} = h(PW_{new} \parallel b)$, $v_{new} = W_{new} \oplus T_d(ID_i \parallel Q)$, $X_{new} = b \oplus h(ID_i \parallel PW_{new})$ and $Y_{new} = h(T_d(ID_i \parallel Q) \parallel b \parallel h(ID_i \parallel PW_{new}))$.

P3. Finally, the smart card replaces v, X and Y with v_{new}, X_{new} and Y_{new}.

5. Smart Card Revocation Phase

Suppose a legal user U_i loses his/her smart card, U_i informs S regarding his/her revocation in person. S then confirms the authenticity of U_i by verifying U_i's identification papers. On successful authentication, S asks U_i to select a new password and a new random number and execute the same steps of the registration phase. Finally S sets $Q = Q + 1$ in the field of registration times for U_i.

3.2 Crytpanalysis of Li et al.'s Scheme

In this, their scheme has been cryptanalyzed in depth and identified weaknesses are explained in detail. Based on the assumptions mentioned in section 1.3, the scheme is checked for security attacks.

Obtaining secret value d of the server S

Suppose an adversary initially registers as a legal user in the system with his own ID and PW, then the system issues a smart card with all registered parameters in it meaning an adversary can obtain all the parameters inside the card. So, adversary now has the parameters $\{X,\ Y,\ h(.),\ v,\ Q\}$ stored in it by analyzing the power consumption. Now, he calculates $T_d(ID \parallel Q) = v \oplus W$, where $W = h(PW_i \parallel b)$, where password and random number are of the adversary. He then calculates

$$d^l = \frac{arccosT_d(ID_i) + 2k\pi}{arccosT_d(x)}$$

such that $T_d^l(ID \parallel Q) = T_d(ID \parallel Q)$.

3.2.1 No user anonymity

Assume that an adversary comes in possession of the smart card of a user U_i with the values $\{X,\ Y,\ h(.),\ v,\ Q\}$ and he has stored these values. Suppose he eavesdrops the login message $\{T_a(ID_i \parallel Q),\ v\}$ and authentication message $\{Z\}$ between U_i and S, then from the above argument, adversary calculates $a^l = \frac{arccosT_a(ID_i)+2k\pi}{arccosT_a(x)}$ such that $T_a^l(ID_i \parallel Q) = T_a(ID_i \parallel Q)$. He then chooses ID^l, computes $T_a^l(ID^l \parallel Q)$ using the value of Q stored in the smart card and checks if the computed value equals $T_a(ID_i \parallel Q)$. If they are equal, the adversary has guessed the correct identity. If not, he repeats the procedure with different values for ID until he guesses the correct identity. So, user anonymity is not preserved in this scheme.

3.2.2 Vulnerable to password guessing attack

Suppose an adversary gets a smart card having the values $\{X,\ Y,\ h(.), v,\ Q\}$, which he stores for his further purposes, and has eavesdropped login message $\{T_a(ID_i \parallel Q),\ v\}$ and authentication message $\{Z\}$ between U_i and S. From 3.2.1, the adversary already has obtained the ID_i of U_i. Also he has originally calculated d^l such that $T_i^l(ID \parallel Q) = T_d(ID \parallel Q)$. Now using ID_i and Q from smart card, he computes $T_d^l(ID_i \parallel Q)$ which is $T_d(ID_i \parallel Q)$. Using $v = W \oplus T_d(ID_i \parallel Q)$, adversary computes $W = v \oplus T_d(ID_i \parallel Q)$. Also from $X = b \oplus h(ID_i \parallel PW_i)$, b is computed as $b = X \oplus h(ID_i \parallel PW_i)$. The value $Y = h(T_d(ID_i \parallel Q) \parallel b \parallel h(ID_i \parallel PW_i))$ is stored in the smart card. Substituting b in this expression, $Y = h(T_d^l(ID_i \parallel Q) \parallel X \oplus h(ID_i \parallel PW_i) \parallel h(ID_i \parallel PW_i))$ where X is obtained from the smart card. Now adversary guesses a value PW, computes $Y^l = h(T_d^l(ID_i \parallel Q) \parallel X \oplus h(ID_i \parallel PW \parallel h(ID_i \parallel PW))$ and checks if $Y^l = Y$ holds. If it holds, adversary has guessed the correct password. If not, he repeats the procedure with different values for PW until he guesses the correct value. So the scheme cannot provide protection against offline password guessing attack.

3.2.3 Vulnerable to user impersonation attack

Assume that an adversary has the values ID_i and U_i of a legal user along with the stored smart card values $\{X, Y, h(.), v, Q\}$. He chooses a random number a^{ll} and computes $T_a^{ll}(ID_i \parallel Q)$ from the stored value Q. He further computes $K_{US}^{ll} = T_a^{ll} T_d^l(ID_i||Q)$, $R^{ll} = h(ID_i \parallel T_a^{ll}(ID_i \parallel Q) \parallel T_d^l(ID_i \parallel Q))$ and $v_1^l = K_{US}^{ll}(ID_i \parallel R^{ll})$, sends the login request message $\{T_a^{ll}(ID_i \parallel Q),\ v^{ll}\}$ to the server S. On receiving this message, server does the required computations including the session key, $SK^{ll} = h(ID_i \parallel T_d^l(ID_i \parallel Q) \parallel T_a^{ll}(ID_i \parallel Q))$ and sends Z^{ll} to the user, where $Z^{ll} = h(ID_i \parallel SK^{ll} \parallel T_d^l(ID_i \parallel Q))$. So, the adversary successfully impersonated as the legal user. Hence, their scheme cannot resist user impersonation attack.

3.2.4 Inconvenient smart card revocation

In this scheme, if a legal user wants to revoke a lost smart card, he/she must inform the medical server in person meaning revocation cannot take place online and the user has to be physically present causing inconvenience to the user and hence it can be concluded that smart card revocation is not very convenient from the point of view of the user.

3.2.5 Insecure session key

Assume that an adversary has eavesdropped the login message $\{T_a(ID_i \parallel Q),\ v\}$ as well as the authentication message $\{Z\}$ of a user and has stored the values from the smart card. As explained in 3.2.1, adversary can obtain the ID_i. Also he has the value of $T_a(ID_i \parallel Q)$ from the login message and he has calculated d^l such that $T_d^l(ID_i \parallel Q) = T_d(ID_i \parallel Q)$. He can easily compute the session key as $SK = h(ID_i \parallel T_d^l(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q))$. Therefore, the session key is not secure in their scheme.

3.2.6 Vulnerable to server impersonation

As explained in 3.2.1, an adversary has user identity ID_i from the smart card and obtains the required password PW_i of the user U_i as explained in 3.2.2. Also, he has the value $T_d^l(ID_i \parallel Q) = T_d(ID_i \parallel Q)$ as explained above. Assume that the adversary obtains the user login request message $\{T_a(ID_i \parallel Q),\ v\}$. Using $T_d^l(ID_i \parallel Q)$, he computes $SK =$

$h(ID_i \parallel T_d^l(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q)), Z = h(ID_i \parallel SK \parallel T_d(ID_i \parallel Q))$ and sends the $\{Z\}$ to the user U_i. On receiving $\{Z\}$, U_i computes $SK^l = h(ID_i \parallel T_d(ID_i \parallel Q) \parallel T_a(ID_i \parallel Q))$ and the condition $h(ID_i \parallel SK^l \parallel T_d(ID_i \parallel Q)) = Z$ holds since $T_d^l(ID_i \parallel Q) = T_d(ID_i \parallel Q)$. User authenticates him as the authentic server and communicates with him proving that the adversary has successfully impersonated the server.

3.3 The Proposed Scheme

In this section, the proposed scheme is explained in detail. Assuring secure communication between user and server being the primary concern, Chebyshev polynomial (explained in 2.3.1) is used in the proposed scheme, along with hash operations.

1. Registration Phase

   To access services from a trusted medical server, new user U_i has to register himself/herself initially. This phase is shown in Fig 3.1. The steps in this phase are as follows.

   R1. User U_i chooses a username ID_i, a password PW_i and a secret number b. Then he computes the masked password $A_i = b \oplus h(ID_i \parallel PW_i)$ and sends the registration message $\{ID_i,\ A_i\}$ to the server S_j via a secure channel.

   R2. On receiving the registration request message $\{ID_i,\ A_i\}$ from the user U_i, the server S_j validates the identity ID_i of U_i. If it is valid, the server S_j computes $T_x(ID_i \parallel m_i)$, where x is the secret key of the server and m_i is a random number chosen by server for U_i(which is unique for every user). It then computes $B_i = T_x(ID_i \parallel m_i) \oplus h(A_i)$.

   R3. S_j then issues a smart card containing the information $\{h(.),\ B_i,\ m_i\}$ to the user U_i via a secure channel

   R4. After receiving the smart card securely from the server S_j, U_i computes $T_x(ID_i \parallel m_i) = B_i \oplus h(A_i)$, $C_i = b \oplus h(PW_i \parallel ID_i)$, $D_i = h(C_i \parallel h(PW_i) \parallel T_x(ID_i \parallel m_i)$ and stores C_i and D_i in the smart card.

   This completes the registration phase of a new user U_i and Fig. 1 represents this phase. The smart card now has the values $\{B_i,\ h(.),\ C_i,\ D_i,\ m_i\}$ stored in it.

   This completes the registration phase of a new user U_i.

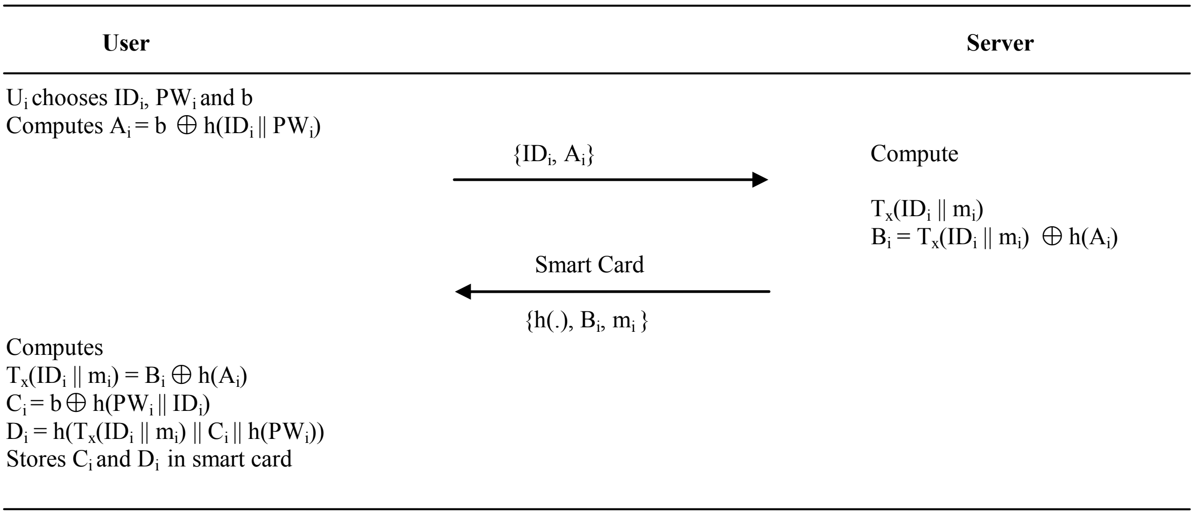
2. Login Phase

   If a registered user wants to login to the TMIS server S_j, the following steps will be executed:

L1. U_i inserts his/her smart card the card reader of a terminal, inputs his/her identity ID_i and password PW_i. The smart card of U_i computes $b^l = C_i \oplus h(PW_i \parallel ID_i)$ and $A_i^l = b^l \oplus h(ID_i \parallel PW_i)$.

L2. Using A_i^l, the smart card obtains $T_x(ID_i \parallel m_i)^l = h(A_i^l) \oplus B_i$ and computes $D_i^l = h(C_i \parallel h(PW_i) \parallel T_x(ID_i \parallel m_i)^l)$. Then it checks if $D_i^l = D_i$ holds or not. If it does not hold, the session is aborted. Else step L3 is executed.

L3. The smart card computes generates a random number y and computes $E_i = h(D_i) \oplus T_y(ID_i \parallel m_i)$, $CID_i = h(ID_i) \oplus T_y(T_x(ID_i \parallel m_i)$ and $F_i = h(E_i \parallel T_y(ID_i \parallel m_i) \parallel h(ID_i)$. Finally the smart card of U_i sends the login request message $\{CID_i,\ D_i,\ E_i,\ F_i\}$ to S_j through a public channel.



User
Server
U_i chooses ID_i, PW_i and b
Computes $A_i = b \oplus h(ID_i \parallel PW_i)$
$\{ID_i, A_i\}$
Compute
$T_x(ID_i \parallel m_i)$
$B_i = T_x(ID_i \parallel m_i) \oplus h(A_i)$
Smart Card
$\{h(.), B_i, m_i\}$
Computes
$T_x(ID_i \parallel m_i) = B_i \oplus h(A_i)$
$C_i = b \oplus h(PW_i \parallel ID_i)$
$D_i = h(T_x(ID_i \parallel m_i) \parallel C_i \parallel h(PW_i))$
Stores C_i and D_i in smart card

Figure 3.1 User registration of chaotic maps based proposed scheme

3. Authentication Phase

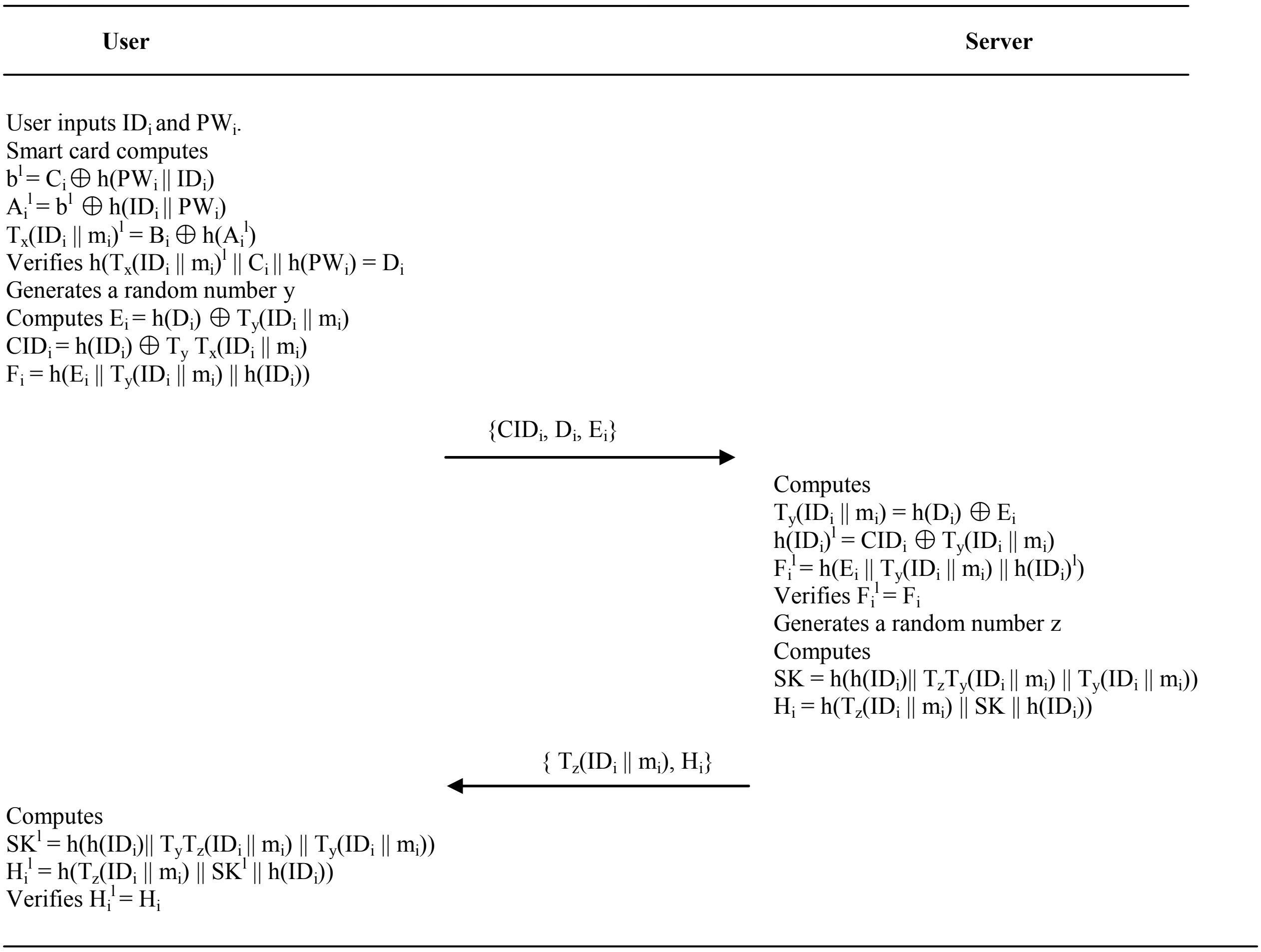
On receiving the login request message $\{CID_i,\ D_i,\ E_i,\ F_i\}$ from U_i, server S_j performs the following steps to authenticate the user U_i. Fig 3.2 demonstrates the login and authentication phases of the proposed scheme.

A1. S_j obtains $T_y(ID_i \parallel m_i) = h(D_i) \oplus E_i$ and computes $h(ID_i) = CID_i \oplus T_x(T_y(ID_i \parallel m_i))$. Then it computes $h(E_i \parallel T_y(ID_i \parallel m_i) \parallel h(ID_i)$ and checks if it is equal to F_i received in the login message. If it holds, S_j authenticates U_i and executes step A2; otherwise this session is terminated.

A2. S_j generates a random number z and computes the session key, $SK = h(T_z(T_y(ID_i) \parallel m_i) \parallel h(ID_i) \parallel T_y(ID_i \parallel m_i))$ and $H_i = h(T_z(ID_i \parallel m_i) \parallel SK \parallel h(ID_i))$. Then it sends the authentication message $\{T_z(ID_i \parallel m_i),\ H_i\}$ to U_i through a public channel.

A3. On receiving $\{T_z(ID_i \parallel m_i),\ H_i\}$ from S_j, U_i computes $SK^l = h(T_y(T_z(ID_i \parallel m_i)) \parallel h(ID_i) \parallel T_y(ID_i \parallel m_i))$ and $H_i^l = h(T_y(ID_i \parallel m_i) \parallel SK^l \parallel h(ID_i))$. Then it verifies if the condition $H_i^l = H_i$ holds or not. If it holds, U_i authenticates S_j. Else, U_i aborts the session.

After mutual authentication, U_i and S_j agree on the shared session key, $SK = h(T_z(T_y(ID_i \parallel m_i)) \parallel h(ID_i \parallel T_y(ID_i \parallel m_i))$ to communicate with each other and this completes the authentication phase.



User **Server**

User inputs ID_i and PW_i.
Smart card computes
$b^l = C_i \oplus h(PW_i \parallel ID_i)$
$A_i^l = b^l \oplus h(ID_i \parallel PW_i)$
$T_x(ID_i \parallel m_i)^l = B_i \oplus h(A_i^l)$
Verifies $h(T_x(ID_i \parallel m_i)^l \parallel C_i \parallel h(PW_i) = D_i$
Generates a random number y
Computes $E_i = h(D_i) \oplus T_y(ID_i \parallel m_i)$
$CID_i = h(ID_i) \oplus T_y\ T_x(ID_i \parallel m_i)$
$F_i = h(E_i \parallel T_y(ID_i \parallel m_i) \parallel h(ID_i))$

$\{CID_i, D_i, E_i\}$

Computes
$T_y(ID_i \parallel m_i) = h(D_i) \oplus E_i$
$h(ID_i)^l = CID_i \oplus T_y(ID_i \parallel m_i)$
$F_i^l = h(E_i \parallel T_y(ID_i \parallel m_i) \parallel h(ID_i)^l)$
Verifies $F_i^l = F_i$
Generates a random number z
Computes
$SK = h(h(ID_i)\parallel T_zT_y(ID_i \parallel m_i) \parallel T_y(ID_i \parallel m_i))$
$H_i = h(T_z(ID_i \parallel m_i) \parallel SK \parallel h(ID_i))$

$\{ T_z(ID_i \parallel m_i), H_i\}$

Computes
$SK^l = h(h(ID_i)\parallel T_yT_z(ID_i \parallel m_i) \parallel T_y(ID_i \parallel m_i))$
$H_i^l = h(T_z(ID_i \parallel m_i) \parallel SK^l \parallel h(ID_i))$
Verifies $H_i^l = H_i$

Figure 3.2 Login and authentication phases of chaotic maps based proposed scheme

4. Password Change Phase

Whenever a user wants to change or update his/her password, the following steps will be performed:

P1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i and password PW_i. The smart card of U_i computes $b^l = C_i \oplus h(PW_i \parallel ID_i)$ and $A_i^l = b^l \oplus h(ID_i \parallel PW_i)$. Using A_i^l, the smart card obtains

$T_x(ID_i \parallel m_i)^l = h(A_i^l) \oplus B_i$ and computes $D_i^l = h(C_i \parallel h(PW_i) \parallel T_x(ID_i \parallel m_i)^l)$. Then it checks if $D_i^l = D_i$ holds or not. If it does not hold, the session is aborted and user cannot change password. Else, the server requests for a new password from the user.

P2. User U_i enters the new password PW_i^{new}. The smart card computes $A_i^{new} = b \oplus h(ID_i \parallel PW_i^{new})$, $B_i^{new} = h(A_i^{new}) \oplus T_x(ID_i \parallel m_i)$, $C_i^{new} = b \oplus h(PW_i^{new} \parallel ID_i)$ and $D_i^{new} = h(C_i^{new} \parallel h(PW_i^{new}) \parallel T_x(ID_i \parallel m_i)^l)$. It then replaces the values B_i, C_i and D_i with B_i^{new}, C_i^{new} and D_i^{new} respectively.
This completes the password change phase.

5. Smart Card Revocation Phase
If a user loses his/her smart card, he/she can send an online request to the server entering the identity ID_i. After checking if that ID_i is valid, the user has to answer the security questions. On successful verification of the legality of the user, the server deactivates the old smart card concerned with that ID_i and requests the user to enter a new masked password A_i^{new}. Using A_i^{new} and ID_i, the server does the required computations as in registration phase and issues a new smart card to the user with the newly computed values. On receiving the new smart card, the user computes C_i and D_i and stores in the smart card.

3.4 Security Analysis of the Proposed Scheme

In this section, the informal proof for the security of the proposed scheme is discussed. The security flaws that were found in Li et al.s' scheme are rectified in the proposed scheme. The proof for this is discussed below.

3.4.1 Preserves user anonymity

The identity ID_i of user is not stored in smart card and in login phase, identity is sent as dynamic identity CID_i, which changes during every session. Suppose an adversary eavesdrops the login and/or authentication messages, $\{CID_i,\ C_2,\ C_3,\ T_1\}$ and/or $\{a,\ b,\ T_3\}$ respectively of the user U_i, revealing ID_i is impossible as it is combined with two secret values H and K known only to the server S_j. Hence, the user anonymity is preserved in the proposed scheme.

3.4.2 Secure against password guessing attack

In the proposed scheme, the password PW_i of a user U_i is covered with his/her own secret value X_u and identity ID_i. Now suppose an adversary gets to know ID_i, it is not possible to guess PW_i without the knowledge of X_u, which is a secret number known only to the user U_i. Suppose the adversary obtains the values $\{E_i,\ h(.),\ N,\ p,\ s_i\}$ from the smart card of the user U_i, even then the scheme forbids password guessing because the password PW_i is protected with two secret keys of the server as well as ID_i and X_u in the expressions $RPW_i = X_u \oplus h(ID_i \parallel PW_i)$ and $N = h(H \parallel K) \oplus RPW_i$. Therefore, the proposed scheme resists offline password guessing attack.

3.4.3 Secure against user impersonation attack

Since time stamps have been used in the proposed scheme, impersonation attack is not possible. If an adversary has all the smart card values $\{E_i,\ h(.),\ N,\ p,\ s_1\}$ and previously intercepted login message $\{CID_i,\ C_2,\ C_3,\ T_1\}$, it is not possible to generate a valid login message in the next session since time stamp and a nonce, r_1 have been used in the proposed scheme. Guessing these values and forming a valid login message within the time stamp is practically impossible. So, protection against impersonation attack is provided in the proposed scheme.

3.4.4 Efficient smart card revocation

Unlike Li et al's scheme, the user need not go in person to revoke a lost smart card. He/she has to send an online request and verify his/her identity after which a new smart card will be issued. It has to be noted that in the proposed scheme, server deactivates the old smart card on verification of the legality of the user of the lost smart card so that even if adversary comes in possession with that card, he/she cannot misuse the card. So, smart card revocation is efficient in the proposed scheme.

3.4.5 Secure session key

In the proposed scheme, the session key SK computed in steps A5 and A6 contains two random numbers r_1 and r_2. If the adversary gets to know these numbers during any session, even then session key cannot be compromised because these random numbers vary during each session with which the session key keeps changing. Along with the knowledge of r_1 and r_2, an adversary needs to have the correct ID_i to obtain the session key of the user U_i. Also if the adversary gets any information regarding previous session key, still it will not help him in computing the next session key due to its format, $SK = h(r_{[}1 \parallel r_2^{l} \parallel ID_i \parallel C_1 \parallel v)$. So, the session key is well protected in the proposed scheme.

3.4.6 Secure against server impersonation attack

Time stamps have been used in the proposed scheme because of which server impersonation is not possible. If an adversary has all the smart card values $\{E_i,\ h(.),\ N,\ p,\ s_1\}$ and previously intercepted login message $\{CID_i,\ C_2,\ C_3,\ T_1\}$, it is not possible to generate a valid authentication message in the next session since time stamp along with a nonce, r_2 have been used in the proposed scheme. Guessing these values and forming a valid authentication message within the time stamp is impossible. So, protection against server impersonation attack is provided in the proposed scheme.

3.5 Security Analysis using BAN logic

In this, the legitimacy of session keys in the proposed scheme is verified. The constructs used in the proof are mentioned in section 1.4. For this, the messages transmitted between user U_i and server S_j are written in idealized form as shown below.
Idealized protocol:
$U \rightarrow S : \langle ID \rangle_{U \overset{s_2}{\leftrightarrow} S}, \langle ID \rangle_{\{U \overset{s_2}{\leftrightarrow} S\}_{r_1}}, (U \overset{SK}{\leftrightarrow} S, \{U \overset{s_2}{\leftrightarrow} S\}_{r_2})_{U \overset{s_2}{\leftrightarrow} S}$

$S \rightarrow U : (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{s_2}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{s_2}} S, \langle ID \rangle_{\{U \overset{s_2}{\leftrightarrow} S\}_{r_2}}$
According to the logical postulates, the proposed scheme should satisfy the following goals:

G1. $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$

G2. $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$

The following assumptions are made to achieve the desired goals:

A1. $S \mid\equiv U \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A2. $U \mid\equiv S \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A3. $U \mid\equiv \#(r_1)$

A4. $S \mid\equiv \#(r_2)$

A5. $U \mid\equiv U \overset{s_2}{\leftrightarrow} S$

A6. $S \mid\equiv U \overset{s_2}{\leftrightarrow} S$

Analysis:

P1. Since $U \lhd (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{s_2}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{s_2} S}$, applying message-meaning rule using A5, we obtain $U \mid\equiv S \mid\sim (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{s_2}{\leftrightarrow} S\}_{r_1})$.

P2. From A3 and P1, application of nonce-verification rule yields $U \mid\equiv S \mid\equiv (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{s_2}{\leftrightarrow} S\}_{r_1})$.

P3. From P2 and A5, we can break the conjunction to obtain $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$ (G1 is achieved).

P4. Since $S \lhd (U \overset{SK}{\leftrightarrow} S, \{U \overset{s_2}{\leftrightarrow} S\}_{r_2})_{U \overset{s_2}{\leftrightarrow} S}$, using A6 and applying message-meaning rule, we obtain $S \mid\equiv U \mid\sim (U \overset{SK}{\leftrightarrow} S, \{U \overset{s_2}{\leftrightarrow} S\}_{r_2})$.

P5. From A4 and P4, using nonce-verification rule, we obtain $S \mid\equiv U \mid\equiv (U \overset{SK}{\leftrightarrow} S, \{U \overset{s_2}{\leftrightarrow} S\}_{r_2})$.

P6. Using P5 and A6, we obtain $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$ (G2 is achieved).

From G1 and G2, it can be observed that both the user U_i and server S_j believe that the session key $SK = h(r_1^l \parallel r_2 \parallel ID_i \parallel C_1 \parallel v)$ is shared between them.

3.6 Performance Comparison

In this section, a detailed comparison of the proposed scheme with Li et al's scheme and other schemes (Das, 2015; Mir et al., 2015) has been made in terms of computational cost, execution time and performance. Table 3.1 shows the computational cost comparison. Table 3.2 compares the execution time of different schemes and these are computed using experiment mentioned in section 2.4. Comparison of performance of the proposed scheme with other schemes is presented in Table 3.3.

Table 3.1 Computational cost comparison with Li et al.'s scheme

Phase	Li et al. (2016)	Das (2015)	Mir et al. (2015)	Proposed
Registration	$3T_h + 1T_{ch}$	$4T_h$	$6T_h$	$5T_h + 1T_{ch}$
Login	$4T_h + 2T_{ch}$	$4T_h$	$7T_h$	$7T_h + 2T_{ch}$
Authentication	$5T_h + 2T_{ch}$	$11T_h$	$13T_h$	$6T_h + 2T_{ch}$
Total	$12T_h + 5T_{ch}$	$19T_h$	$26T_h$	$18T_h + 5T_{ch}$

From Table 3.1, it can be observed that the proposed scheme requires five hash operations more than that of Li et al's scheme but the number of chaotic operations is same. But when compared to Das (2015), the proposed scheme uses one hash operation less. From Table 3.3, it can be clearly seen that even with less hash operations, the proposed scheme is able to provide more security than Das (2015). In comparison with Mir et al. (2015) also, the proposed scheme uses seven less hash operations but makes use of chaotic maps.

Table 3.2 Execution time comparison with Li et al.'s scheme(s)

Phase	Li et al. (2016)	Das (2015)	Mir et al. (2015)	Proposed
Registration	0.02252	0.0020	0.0030	0.02352
Login	0.04404	0.0020	0.0035	0.04554
Authentication	0.04454	0.0055	0.0065	0.04504
Total	0.11110	0.0095	0.0130	0.11410

Table 3.3 Performance comparison with Li et al.'s scheme

Security Properties	Li et al.	Das	Mir et al.	Proposed
Provides user anonymity	No	No	Yes	Yes
Resists user impersonation	No	No	Yes	Yes
Resists stolen-verifier attack	Yes	Yes	Yes	Yes
Resists replay attack	Yes	Yes	Yes	Yes
Secure session key	No	No	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes
Resists offline password guessing	No	No	Yes	Yes
Resists man-in-the-middle attack	No	No	Yes	Yes
Resists server impersonation	No	No	Yes	Yes
Resists privileged insider attack	Yes	Yes	Yes	Yes
Resists parallel session attack	Yes	Yes	Yes	Yes
Efficient smart card revocation	No	No	No	Yes

From Table 3.2, it is clear that the proposed scheme requires 0.003s more than Li et al. (2016). But this extra time can be justified from Table 3.3 by noting that the proposed scheme is able to overcome those security attacks which were found in Li et al.'s scheme, thereby providing more security. The scheme in Mir et al. (2015) requires less time for execution but, the scheme is not user friendly since there is no option for smart card revocation. Also, Das (2015) estimated time is less than that of the proposed scheme but Table 3.3 clearly shows that their scheme fails to overcome user and server impersonation, offline password guessing attacks; does not preserve user anonymity, insecure session key. From this, the schemes using less hash operations are not able to overcome all the attacks but the proposed scheme overcomes the mentioned security attacks. To achieve these properties, it is worth the additional operations. So, the proposed scheme is robust and more secure when compared to Li et al.'s scheme.

CHAPTER 4

CRYPTANALYSIS OF AN ANONYMOUS TWO-FACTOR AUTHENTICATION SCHEME

Chen et al. (2018) proposed an anonymous and lightweight authentication scheme for TMIS. Their aim was to enhance the security of Kan et al.'s scheme proposed for TMIS. They claim that their scheme ensures mutual authentication, user anonymity, session key security, and resists password guessing, stolen verifier attack, impersonation, replay attack as well as traceability attack. This two-factor scheme has been cryptanalyzed in this section. In this, weaknesses like impersonation, password guessing and server masquerading attacks were found. Also, the scheme fails to preserve user anonymity. These issues have been explained in detail. In order to resist these attacks, an improved scheme has been proposed in this chapter. Following this, the security analysis is explained using BAN logic. Also, the proposed scheme is compared with contemporary schemes which prove that proposed scheme outcomes those schemes in terms of computational efficiency.

4.1 Review of Chen et al.'s scheme

In this section, a detailed review of the scheme proposed in Chen et al. (2018) is given. The various phases are given below.

1. Registration Phase
   To access services from the telecare medicine information system server S_i, a new user U_i must register himself/herself at the server S_i. The various notations used in their scheme are given in Table 2.1. The following steps are performed during registration:

R1. U_i initially selects his/her identity ID_i, a password PW_i and a random number b. U_i computes $RPW_i = h(PW_i \parallel b)$ and sends the registration message $\{ID_i,\ RPW_i\}$ to the server S_i via a secure channel.

R2. On receiving the registration request message $\{ID_i,\ RPW_i\}$ from the user U_i, S_i generates a random integer N_i and computes $PID_i = E_{X_s}(ID_i \parallel N_i)$, $A_i = h(ID_i \parallel RPW_i)$ and $B_i = h(ID_i \parallel X_s) \oplus h(ID_i) \oplus RPW_i$.

R3. S_i then issues a smart card with values $\{PID_i,\ B_i,\ h(.),\ A_i\}$ to the user U_i via a secure channel.

R4. On receiving the smart card from server, U_i stores b in it and the smart card now has values $\{PID_i,\ b,\ B_i,\ h(.),\ A_i\}$.

2. Login Phase

Whenever a registered user wants to login to the TMIS system server S_i, the following steps will be executed:

L1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i^l and password PW_i^l. The smart card of U_i computes $RPW_i^l = h(PW_i^l \parallel b)$ and $A_i^l = h(ID_i^l \parallel RPW_i^l)$. It then verifies if $A_i^l = A_i$ holds or not. If it holds, it goes to next step. Otherwise, the login session is terminated.

L2. U_i's smart card generates a random integer, r_u and acquires the current time stamp T_1. Then it computes $C_i = B_i \oplus h(ID_i^l) \oplus RPW_i^l$, $D_i = C_i \oplus r_u$ and $E_i = h(ID_i^l \parallel D_i \parallel C_i \parallel T_1)$.

L3. Finally, U_i sends the login request message $\{T_1,\ D_i,\ PID_i,\ E_i\}$ through a public channel to the server S_i.

3. Authentication Phase

On receiving the login request message $\{T_1,\ D_i,\ PID_i,\ E_i\}$ from U_i, S_i performs the following steps:

A1. S_i retrieves the current time stamp T_2 and verifies the freshness of U_i's time stamp, T_1. If valid, it obtains $(ID_i \parallel N_i) = D_{X_s}(PID_i)$ and computes $C_i^l = h(ID_i \parallel X_s)$ and $E_i^l = h(ID_i^l \parallel D_i \parallel C_i^l \parallel T_1)$. It then verifies if $E_i^l = E_i$ holds or not. If it holds, user is authenticated and authentication process continues. If it does not hold, S_i rejects the service request message and the authentication phase is terminated.

A2. S_i generates two random integers, N_i^{new}, r_s and the time stamp T_3. It then computes $PID_i^{new} = E_{X_s}(ID_i \parallel N_i^{new})$, $r_u^l = D_i \oplus C_i^l$, $F_i = C_i^l \oplus r_s$, $SK = h(ID_i \parallel r_u^l \parallel r_s \parallel C_i^l)$ and $H_i = h(ID_i \parallel F_i \parallel C_i^l \parallel SK \parallel T_3)$. Then it sends the authentication request message $\{PID_i^{new},\ F_i,\ H_i,\ T_3\}$ to the user U_i.

A3. On receiving the message $\{PID_i^{new},\ F_i,\ H_i,\ T_3\}$ from S_i, U_i retrieves the current time stamp T_4 and verifies the freshness of S_i's time stamp T_3. Then it computes $r_s^l = F_i \oplus C_i$, $SK^l = h(ID_i^l \parallel r_u \parallel r_s^l \parallel C_i)$, $H_i^l = h(ID_i^l \parallel F_i \parallel C_i \parallel SK^l \parallel T_3)$ and checks if the condition $H_i^l = H_i$ holds or not. If the condition holds, S_i is authenticated and PID_i is replaced by PID_i^{new} in the smart card's memory. Otherwise, the session is terminated.

A4. U_i generates the current time stamp T_5, computes $M_i = h(SK^l \parallel C_i \parallel T_5)$ and sends the response message $\{M_i,\ T_5\}$ to S_i.

A5. On receiving $\{M_i,\ T_5\}$ from U_i, S_i retrieves the current time stamp T_6 and verifies the freshness of U_i's time stamp T_5. Then it computes $M_i^l = h(SK \parallel C_i^l \parallel T_5)$ and checks if $M_i^l = M_i$ holds or not. If it holds, server believes that they have established the session key SK.

4. Password Change Phase

Suppose a user U_i wishes to change his/her password, the following steps are performed:

P1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i^l and password PW_i^l. The smart card of U_i computes $RPW_i^l = h(PW_i^l \parallel b)$ and $A_i^l = h(ID_i^l \parallel RPW_i^l)$. It then verifies if $A_i^l = A_i$ holds or not. If the verification holds, U_i enters a new password PW_{new}, else the request is denied.

P2. The smart card computes $A_i^{new} = h(ID_i^l \parallel h(PW_i^{new} \parallel b))$ and $B_i^{new} = B_i \oplus RPW_i^l \oplus h(PW_i^{new} \parallel b)$.

P3. Finally, the smart card replaces A_i and B_i with A_i^{new} and B_i^{new} respectively.

4.2 Security weaknesses in Chen et al.'s Scheme

In this part, Chen et al.'s scheme is cryptanalyzed using the assumptions given in 1.3. Security weaknesses that were identified are discussed.

Obtaining secret value X_s of the server S_i

An adversary initially registers as a legal user in the system with his own ID, PW, random integer b_a and $RPW_a = h(PW \parallel b_a)$. The system issues a smart card with parameters $\{PID_a,\ B_a,\ h(.),\ A_a\}$ in it. Then he stores b_a in the smart card. Now he guesses a value X_s^l and computes $h(ID \parallel X_s^l) \oplus h(ID) \oplus RPW_a$. Then he checks if the computed value equals B_a or not. If yes, he has guessed the correct X_s; else he repeats the procedure with different values for X_s until he guesses the correct value. Once he has guessed the correct value of X_s, he obtains $(ID \parallel N_a) = D_{X_s}(PID_a)$, where N_a is the random integer chosen by the server during registration. Since ID is known, adversary obtains N_a and learns the format of it and stores it.

4.2.1 No user anonymity

Assume that an adversary comes in possession of the smart card of a user U_i with the values $\{PID_i,\ b,\ B_i,\ h(.),\ A_i\}$. As explained above, he has knowledge of X_s. He decrypts PID_i to obtain $(ID_i \parallel N_i) = D_{X_s}(PID_i)$. Since he has stored the format of N_i, he separates that many bits from $ID_i \parallel N_i$ and obtains the ID_i. So, user anonymity is not preserved.

4.2.2 Vulnerable to password guessing attack

Suppose an adversary gets a smart card of a user with values $\{PID_i,\ b,\ B_i,\ h(.),\ A_i\}$, which he stores for his further purposes. From 4.2.1, the adversary already has obtained the ID_i of U_i and also knows X_s. He obtains $RPW_i = h(ID_i \parallel X_s) \oplus h(ID_i) \oplus B_i$, where B_i is obtained from smart card. Now he guesses a value PW^l, computes $RPW_i^l = h(PW_i^l \parallel b)$ (where b is stored in the smart card) and checks if $RPW_i^l = RPW_i$ holds. If it holds, adversary has guessed the correct password. If not, he repeats the procedure with different values for PW_i until he guesses the correct value. So the scheme cannot provide protection against password guessing attack.

4.2.3 Vulnerable to user impersonation attack

Assume that an adversary has the values ID_i and PW_i of a legal user along with the stored smart card values $\{PID_i,\ b,\ B_i,\ h(.),\ A_i\}$. He chooses a random number r_a and computes $C_i = B_i \oplus h(ID_i) \oplus RPW_i$, $D_i = C_i \oplus r_a$ and $E_i = h(ID_i \parallel D_i \parallel C_i \parallel T_a)$, where T_a is the time stamp. He then sends $\{T_a,\ D_i,\ PID_i,\ E_i\}$ through a public channel to the server S. On receiving this message, server does the required computations and sends $\{PID_i^{new},\ F_i,\ H_i,\ T_3\}$ to the adversary assuming him to be the legal user. So, the adversary successfully impersonated as the legal user. Hence, the scheme cannot resist user impersonation attack.

4.2.4 Vulnerable to server impersonation attack

As explained in 4.2.1, an adversary has user identity ID_i from the smart card and obtains the required password PW_i of the user U_i as explained in 4.2.2. Suppose he eavesdrops the login message $\{T_1, D_i, PID_i, E_i\}$ of U_i, he chooses two random integers, N_a, r_{as} and the time stamp T_*. He then computes $PID_i^{new} = E_{X_s}(ID_i \| N_a)$, $r_u = D_i \oplus C_i$, $F_i = C_i \oplus r_{as}$, $SK = h(ID_i \| r_u \| r_{as} \| C_i)$ and $H_i = h(ID_i \| F_i \| C_i \| SK \| T_*)$, where $C_i = h(ID_i \| X_s)$. Then it sends the authentication request message $\{PID_i^{new}, F_i, H_i, T_*\}$ to the user U_i. On receiving this, U_i retrieves the current time stamp T_u and verifies the freshness of time stamp T_*. Then he computes $r_s^l = F_i \oplus C_i$, $SK^l = h(ID_i \| r_u \| r_s^l \| C_i)$, $H_i^l = h(ID_i \| F_i \| C_i \| SK^l \| T_*)$ and checks if the condition $H_i^l = H_i$ holds or not and it does hold. Then U_i generates the current time stamp T_{ua}, computes $M_i = h(SK^l \| C_i \| T_{ua})$ and sends the response message $\{M_i, T_{ua}\}$ to the adversary believing him to be the server. Thus, an adversary has successfully impersonated the server.

4.3 Proposed scheme based on hash functions

This section presents the proposed scheme. There are four phases. All these phases are described below in detail.

1. Registration Phase
   This is the first phase where a user has to register himself to a medical server in case he wants to use services from the TMIS server. Notations used are as shown in 2.1. The different steps for registration are explained below and are shown in Fig. 4.1.

R1. U_i selects his identity ID_i, a password PW_i and a random number b. U_i computes the masked password $RPW_i = h(PW_i \| b)$ and sends the registration message $\{ID_i, RPW_i\}$ to the server S_i via a secure channel.

R2. On receiving the message $\{ID_i, RPW_i\}$ from U_i, S_i generates a random integer N_i(unique for each U_i) and computes $CID_i = h(ID_i) \oplus N_i$, $A_i = h(ID_i \| RPW_i)$ and $B_i = h(h(ID_i) \| X_s) \oplus h(ID_i) \oplus RPW_i$, where X_s is the secret key of the S_i.

R3. S_i then issues a smart card to the user U_i with values $\{CID_i, B_i, h(.), A_i\}$ via a secure channel.

R4. On receiving the smart card from server, U_i stores b in it and the smart card now has values $\{CID_i, b, B_i, h(.), A_i\}$.

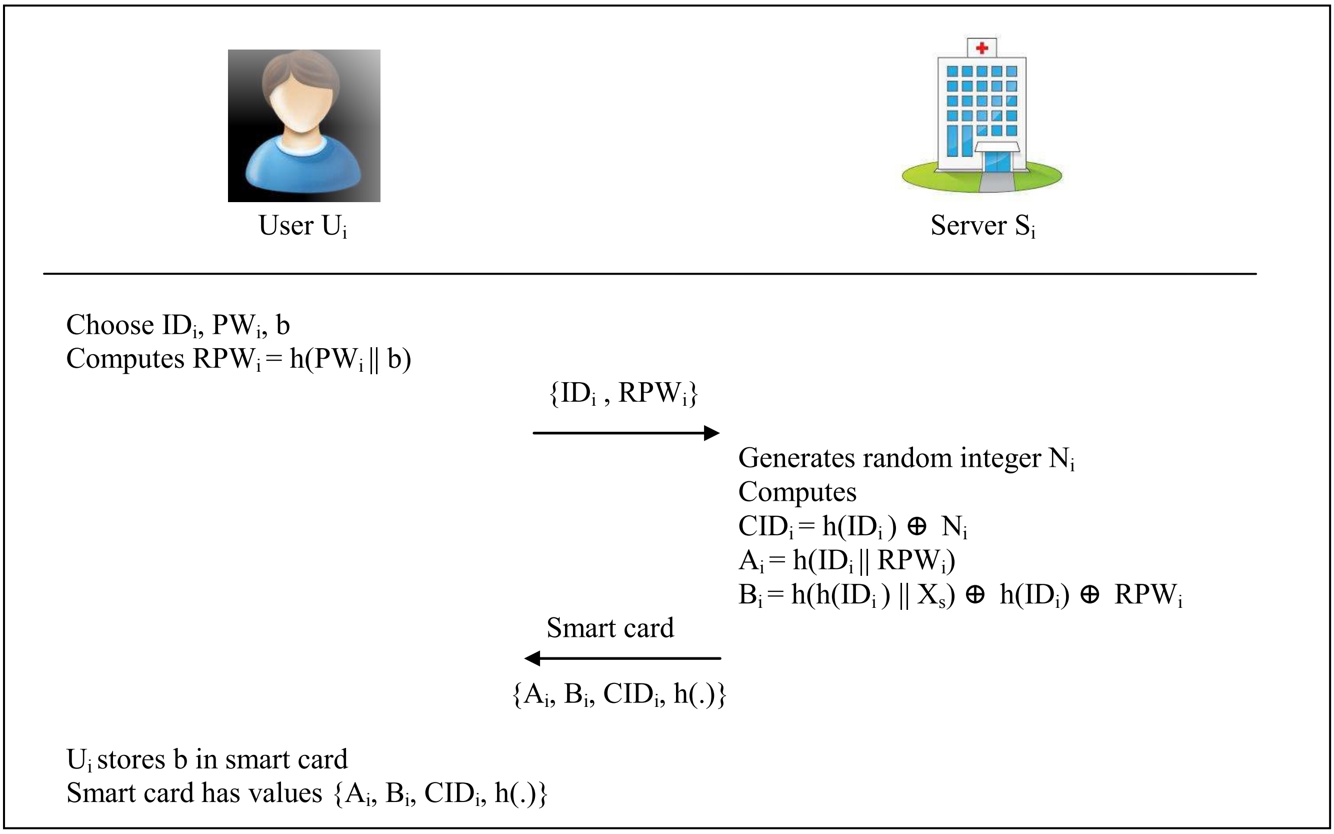

User U_i
Server S_i
Choose ID_i, PW_i, b
Computes $RPW_i = h(PW_i \| b)$
$\{ID_i , RPW_i\}$
Generates random integer N_i
Computes
$CID_i = h(ID_i) \oplus N_i$
$A_i = h(ID_i \| RPW_i)$
$B_i = h(h(ID_i) \| X_s) \oplus h(ID_i) \oplus RPW_i$
Smart card
$\{A_i, B_i, CID_i, h(.)\}$
U_i stores b in smart card
Smart card has values $\{A_i, B_i, CID_i, h(.)\}$

Figure 4.1 Registration phase of hash functions based proposed scheme

2. Login Phase

Whenever a registered user wants to login to the TMIS system server S_i, the following steps will be executed. Sequence diagram of this phase along with authentication is demonstrated in Fig 4.2.

L1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i^l and password PW_i^l. The smart card of U_i computes $RPW_i^l = h(PW_i^l \| b)$ and $A_i^l = h(ID_i^l \| RPW_i^l)$. It then verifies if $A_i^l = A_i$ holds or not. If it holds, it goes to next step. Otherwise, the login session is terminated.

L2. U_i's smart card generates a random integer, r_u and acquires the current time stamp T_1. Then it computes $C_i = B_i \oplus h(ID_i^l) \oplus RPW_i^l$, $D_i = C_i \oplus r_u$ and $E_i = h(h(ID_i^l) \| D_i \| C_i \| T_1)$.

L3. Finally, U_i sends the login request message $\{T_1, D_i, CID_i, E_i\}$ through a public channel to the server S_i.

3. Authentication Phase

On receiving the login request message $\{T_1, D_i, CID_i, E_i\}$ from U_i, S_i performs the following steps:

A1. S retrieves the current time stamp T_2 and verifies the freshness of U_i's time stamp T_1. If valid, it obtains $h(ID_i) = CID_i \oplus N_i$ and computes $C_i^l = h(h(ID_i) \parallel X_s)$ and $E_i^l = h(h(ID_i) \parallel D_i \parallel C_i^l \parallel T_1)$. It then verifies if $E_i^l = E_i$ holds or not. If it holds, U_i is authenticated and authentication process continues. If it does not hold, S_i rejects the service request message and the authentication phase is terminated.

A2. S_i generates two random integers, N_i^{new}, r_s and the time stamp T_3. It then computes $CID_i^{new} = h(ID_i) \oplus N_i^{new}$, $r_u^l = D_i \oplus C_i^l$, $F_i = C_i^l \oplus r_s$, $SK = h(h(ID_i) \parallel r_u^l \parallel r_s \parallel C_i^l)$ and $H_i = h(F_i \parallel C_i^l \parallel SK \parallel T_3)$. Then it sends the authentication request message $\{CID_i^{new},\ F_i,\ H_i,\ T_3\}$ to the user U_i.

A3. On receiving the message $\{CID_i^{new},\ F_i,\ H_i,\ T_3\}$ from S_i, U_i retrieves the current time stamp T_4 and verifies the freshness of S_i's time stamp T_3. Then it computes $r_s^l = F_i \oplus C_i$, $SK^l = h(h(ID_i^l) \parallel r_u \parallel r_s^l \parallel C_i)$, $H_i^l = h(F_i \parallel C_i \parallel SK^l \parallel T_3)$ and checks if the condition $H_i^l = H_i$ holds or not. If the condition holds, the server S_i is authenticated and CID_i is replaced by CID_i^{new} in the smart card's memory. Otherwise, the session is terminated.

A4. U_i generates the current time stamp T_5, computes $M_i = h(SK^l \parallel C_i \parallel T_5)$ and sends the response message $\{M_i,\ T_5\}$ to S_i.

A5. On receiving $\{M_i,\ T_5\}$ from U_i, server retrieves the current time stamp T_6 and verifies the freshness of U_i's time stamp T_5. Then it computes $M_i^l = h(SK \parallel C_i^l \parallel T_5)$ and checks if $M_i^l = M_i$ holds or not. If it holds, U_i and S_i believe that they have established the session key SK.

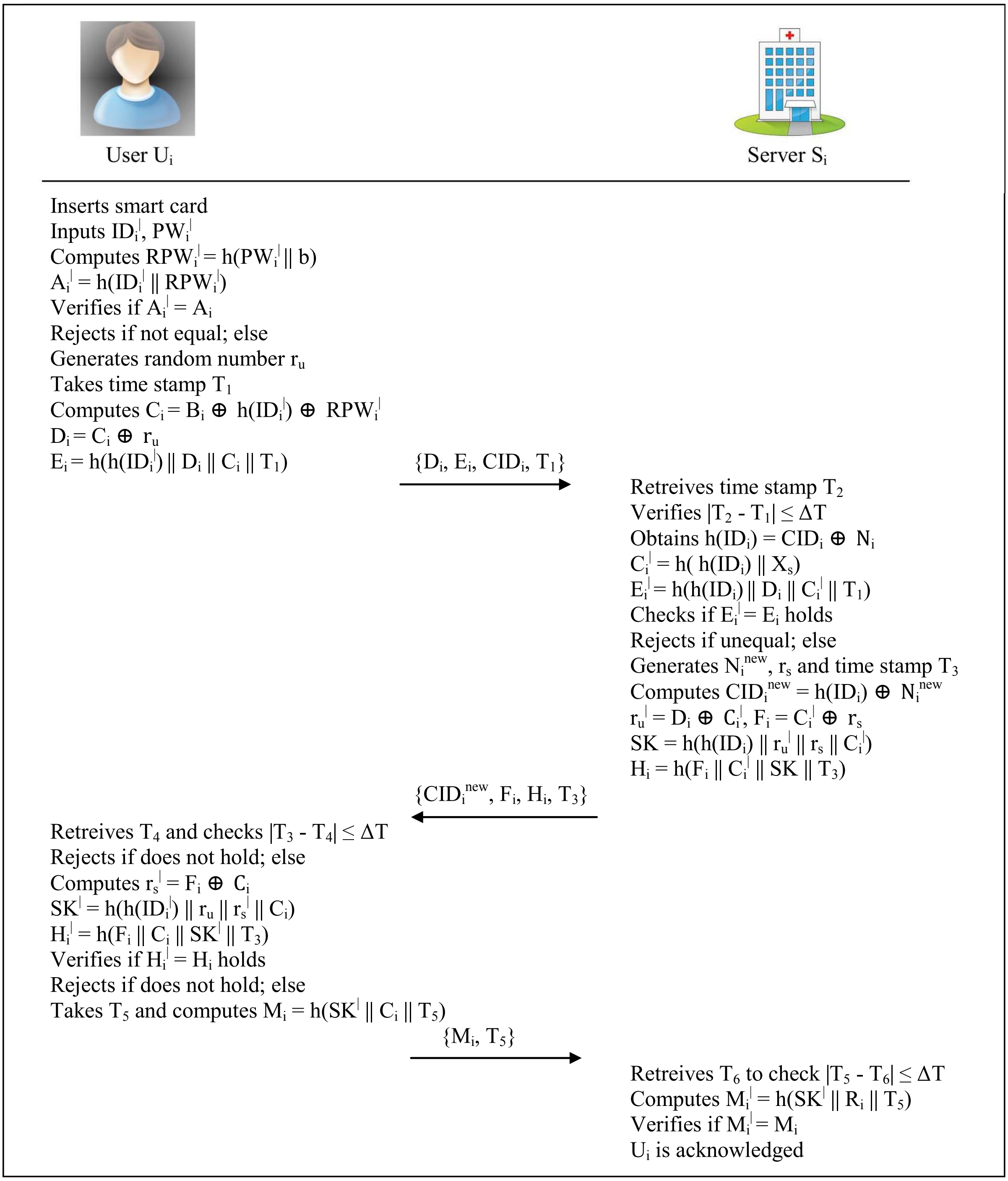
4. Password Change Phase

Suppose a user U_i wishes to change his/her password, the following steps are performed:

P1. U_i inserts his/her smart card into the card reader of a terminal, inputs his/her identity ID_i^l and password PW_i^l. The smart card of U_i computes $RPW_i^l = h(PW_i^l \parallel b)$ and $A_i^l = h(ID_i^l \parallel RPW_i^l)$. It then verifies if $A_i^l = A_i$ holds or not. If the verification holds, U_i enters a new password PW_{new}, else the request is denied.

P2. The smart card computes $A_i^{new} = h(ID_i^l \parallel h(PW_i^{new} \parallel b))$ and $B_i^{new} = B_i \oplus RPW_i^l \oplus h(PW_i^{new} \parallel b)$.

P3. Finally, the smart card replaces A_i and B_i with A_i^{new} and B_i^{new} respectively.



User U_i

Server S_i

Inserts smart card
Inputs $ID_i^{|}$, $PW_i^{|}$
Computes $RPW_i^{|} = h(PW_i^{|} \| b)$
$A_i^{|} = h(ID_i^{|} \| RPW_i^{|})$
Verifies if $A_i^{|} = A_i$
Rejects if not equal; else
Generates random number r_u
Takes time stamp T_1
Computes $C_i = B_i \oplus h(ID_i^{|}) \oplus RPW_i^{|}$
$D_i = C_i \oplus r_u$
$E_i = h(h(ID_i^{|}) \| D_i \| C_i \| T_1)$

$\xrightarrow{\{D_i, E_i, CID_i, T_1\}}$

Retreives time stamp T_2
Verifies $|T_2 - T_1| \leq \Delta T$
Obtains $h(ID_i) = CID_i \oplus N_i$
$C_i^{|} = h(h(ID_i) \| X_s)$
$E_i^{|} = h(h(ID_i) \| D_i \| C_i^{|} \| T_1)$
Checks if $E_i^{|} = E_i$ holds
Rejects if unequal; else
Generates N_i^{new}, r_s and time stamp T_3
Computes $CID_i^{new} = h(ID_i) \oplus N_i^{new}$
$r_u^{|} = D_i \oplus C_i^{|}$, $F_i = C_i^{|} \oplus r_s$
$SK = h(h(ID_i) \| r_u^{|} \| r_s \| C_i^{|})$
$H_i = h(F_i \| C_i^{|} \| SK \| T_3)$

$\xleftarrow{\{CID_i^{new}, F_i, H_i, T_3\}}$

Retreives T_4 and checks $|T_3 - T_4| \leq \Delta T$
Rejects if does not hold; else
Computes $r_s^{|} = F_i \oplus C_i$
$SK^{|} = h(h(ID_i^{|}) \| r_u \| r_s^{|} \| C_i)$
$H_i^{|} = h(F_i \| C_i \| SK^{|} \| T_3)$
Verifies if $H_i^{|} = H_i$ holds
Rejects if does not hold; else
Takes T_5 and computes $M_i = h(SK^{|} \| C_i \| T_5)$

$\xrightarrow{\{M_i, T_5\}}$

Retreives T_6 to check $|T_5 - T_6| \leq \Delta T$
Computes $M_i^{|} = h(SK^{|} \| R_i \| T_5)$
Verifies if $M_i^{|} = M_i$
U_i is acknowledged

Figure 4.2 Login-authentication phases of hash functions based proposed scheme

4.4 Security analysis of the proposed scheme

This section analyzes the hash functions based proposed scheme. In other words, it explains in detail how the proposed scheme is able to resist the security flaws that were found in Chen et al.'s scheme.

4.4.1 User anonymity is preserved

It can be clearly observed that the identity ID_i of any user is not stored in smart card. In login phase, identity of U_i is sent as dynamic identity CID_i in the login message, $\{T_1,\ D_i,\ CID_i,\ E_i\}$. This identity CID_i changes during every session and is replaced by the new identity CID_i^{new}. If an adversary eavesdrops this login message, he cannot obtain ID_i without the knowledge of N_i. Obtaining N_i is impossible since it is unique to every user U_i and is changed during every session by the server. Hence, the user anonymity is preserved in the proposed scheme.

4.4.2 Resists password guessing attack

Suppose an adversary comes gets hold of a smart card with values $\{CID_i,\ b,\ B_i,\ h(.),\ A_i\}$, guessing PW_i is not feasible since it is combined with ID_i and X_s in $B_i = h(h(ID_i) \parallel X_s) \oplus h(ID_i) \oplus RPW_i$. If an adversary obtains the login message $\{T_1,\ D_i,\ CID_i,\ E_i\}$, PW_i is still safe due to the usage of random number, r_u in $D_i = C_i \oplus r_u$ and in turn E_i is computed as $E_i = h(h(ID_i^l) \parallel D_i \parallel C_i \parallel T_1)$. If an adversary eavesdrops the authentication message $\{CID_i^{new},\ F_i,\ H_i,\ T_3\}$, he cannot obtain PW_i since random number, r_s is used in $F_i = C_i^l \oplus r_s$ which is used to calculate $H_i = h(F_i \parallel C_i^l \parallel SK \parallel T_3)$. So, in either cases, the proposed scheme resists password guessing attack.

4.4.3 Resists user impersonation attack

As explained in 4.4.1, user identity cannot be obtained by an adversary. In case, he gets hold of a smart card with values $\{CID_i,\ B_i,\ h(.),\ b,\ A_i\}$, he might get CID_i but not PW_i. To impersonate U_i, an adversary has to generate a valid login message $\{T_1,\ D_i,\ CID_i,\ E_i\}$. This is not possible without the knowledge of ID_i and PW_i. As explained in above section, an adversary cannot obtain password of any user U_i even if he eavesdrops the login and authentication messages, $\{T_1,\ D_i,\ CID_i,\ E_i\}$ and $\{CID_i^{new},\ F_i,\ H_i,\ T_3\}$ respectively. So, the proposed scheme withstands user impersonation attack.

4.4.4 Resists server impersonation attack

To impersonate a server S_i, an adversary needs to have knowledge of N_i which is unique for each U_i. The value N_i corresponding to a user U_i is known only to the server S_i. It is not possible to obtain N_i from the login message $\{T_1,\ D_i,\ CID_i,\ E_i\}$. Even though $CID_i = h(ID_i) \oplus N_i$ contains N_i, it is not possible to obtain N_i without the knowledge of

ID_i. This makes it impossible for an adversary to create a valid authentication message $\{CID_i^{new},\ F_i,\ H_i,\ T_3\}$. So, the proposed scheme withstands server impersonation attack.

4.5 Security proof of the proposed scheme using BAN logic

In this, the legitimacy of session keys in the proposed scheme is verified. The constructs used in the proof are mentioned in section 1.4. For this, the messages transmitted between user U_i and server S are written in idealized form as shown below.

Idealized protocol:

$$U \rightarrow S : \langle ID\rangle_{U \overset{A_i}{\leftrightarrow} S}, \langle ID\rangle_{\{U \overset{K_i}{\leftrightarrow} S\}_{ru}}, (U \overset{SK}{\leftrightarrow} S, \{U \overset{A_i}{\leftrightarrow} S\}_{r_s})_{U \overset{A_i}{\leftrightarrow} S}$$

$$S \rightarrow U : (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{A_i}{\leftrightarrow} S\}_{r_u})_{U \underset{\leftrightarrow}{A_i} S}, \langle ID\rangle_{\{U \overset{s2}{\leftrightarrow} S\}_{r_2}}$$

According to the logical postulates, the proposed scheme should satisfy the following goals:

G1. $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$

G2. $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$

The following assumptions are made to achieve the desired goals:

A1. $S \mid\equiv U \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A2. $U \mid\equiv S \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A3. $U \mid\equiv \#(r_u)$

A4. $S \mid\equiv \#(r_s)$

A5. $U \mid\equiv U \overset{A_i}{\leftrightarrow} S$

A6. $S \mid\equiv U \overset{A_i}{\leftrightarrow} S$

Analysis:

P1. Since $U \triangleleft (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{A_i}{\leftrightarrow} S\}_{r_u})_{U \underset{\leftrightarrow}{A_i} S}$, applying message-meaning rule using A5, we obtain $U \mid\equiv S \mid\sim (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{A_i}{\leftrightarrow} S\}_{r_u})$.

P2. From A3 and P1, application of nonce-verification rule yields $U \mid\equiv S \mid\equiv (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{A_i}{\leftrightarrow} S\}_{r_u})$.

P3. From P2 and A5, we can break the conjunction to obtain $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$ (achieves G1).

P4. Since $S \triangleleft (U \overset{SK}{\leftrightarrow} S, \{U \overset{A_i}{\leftrightarrow} S\}_{r_s})_{U \overset{A_i}{\leftrightarrow} S}$, using A6 and applying message-meaning rule, we obtain $S \mid\equiv U \mid\sim (U \overset{SK}{\leftrightarrow} S, \{U \overset{A_i}{\leftrightarrow} S\}_{r_s})$.

P5. From A4 and P4, using nonce-verification rule, we obtain $S \mid\equiv U \mid\equiv (U \overset{SK}{\leftrightarrow} S, \{U \overset{A_i}{\leftrightarrow} S\}_{r_s})$.

P6. Using P5 and A6, we obtain $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$ (achieves G2).

From G1 and G2, it can be observed that both the user U_i and server S believe that the session key $SK = h(h(ID_i^{|}) \parallel r_u \parallel r_s^{|} \parallel C_i)$ is shared between them.

4.6 Performance comparison of the hash functions based proposed scheme

Table 4.1 compares the computational cost of the proposed scheme with scheme proposed by Chen et al. (2018), Ostad Sharif et al. (2019) and Radhakrishnan and Karuppiah (2019). In this table, the number of operations used (hash operations, point multiplication and symmetric key encryption/decryption) in these schemes are counted and tabulated. Execution time comparison is presented in Table 4.2.

Table 4.1 Computational cost comparison with Chen et al.'s scheme

Phase	Chen et al.	Ostad et al.	Radhakrishnan-Karuppiah	Proposed
Registration	$4T_h + 1T_{ed}$	$5T_h + 2T_{ed} + 2T_{pm}$	$7T_h$	$4T_h$
Login	$4T_h$	$6T_h + 2T_{pm}$	$5T_h$	$4T_h$
Authentication	$8T_h + 1T_{ed}$	$12T_h + 2T_{ed} + 3T_{pm}$	$11T_h$	$8T_h$
Total	$16T_h + 2T_{ed}$	$23T_h + 4T_{ed} + 7T_{pm}$	$23T_h$	$16T_h$

From Table 4.1, it can be observed that the proposed scheme uses same number of hash functions as in Chen et al. (2018) and also does not use symmetric key encryption/decryption. In comparison with the scheme in Ostad Sharif et al. (2019), the proposed scheme uses seven hash operations less. Also, the scheme in Radhakrishnan and Karuppiah (2019) requires seven hash operations more than he proposed scheme. Another important observation is that the proposed scheme does not use symmetric key encryption and decryption techniques which is more time consuming. On the other hand, Table 4.2 clearly shows the huge gap between the estimated execution times of proposed and other schemes. Due to this, it can be said that the proposed scheme is robust and efficient in terms of computations.

Table 4.2 Estimated execution time of Chen et al.'s and other schemes(sec)

Phase	Chen et al.	Ostad et al.	Radhakrishnan-Karuppiah	Proposed scheme
Registration	0.0107	0.12615	0.0035	0.0020
Login	0.0020	0.12615	0.0025	0.0020
Authentication	0.0127	0.18923	0.0055	0.0040
Total	0.0254	0.44153	0.0115	0.0080

CHAPTER 5

IMPROVEMENT OF BIOMETRICS-BASED AUTHENTICATION SCHEMES

In this chapter, we focus on biometrics-based or three-factor authentication schemes, meaning schemes use biometrics along with smart cards. The former part studies Jung et al.'s scheme in detail which was proposed in 2017, whereas the latter part focuses on Han et al.'s scheme proposed in 2018. Both the schemes are thoroughly cryptanalyzed and the security flaws found in them are explained. Also, robust schemes have been proposed along with the required security proof that restrict the mentioned flaws. Following this, comparison of various schemes have been presented to highlight the advantages of proposed schemes.

5.1 Analysis and Improvement of Jung et al.'s scheme

For integrated EPR system, Jung et al. (2017) designed an anonymous biometrics-based authentication scheme. They claimed that their scheme overcomes offline password guessing, server spoofing as well as denial-of-service attacks. They use the concept of biohashing (2.3.2.1) in order to conceal the password of a user. We cryptanalyzed their scheme and identified multiple weaknesses which are discussed in this segment.

5.1.1 Review of Jung et al.'s scheme

The details of their scheme are given below.

1. Registration Phase

   Suppose a new user U_i wants to use services from a medical server S_i, he needs to register himself. For registration of a user U_i, the following steps are executed:

   R1. U_i chooses his ID_i, PW_i, engraves his biometrics B_i and computes $RPW_i =$

$h(PW_i \parallel H(B_i))$. He then delivers a request $\{ID_i, RPW_i\}$ to S_i for registration via a secure channel.

R2. On receiving $\{ID_i,\ RPW_i\}$ from U_i, S_i verifies the users' ID_i. Only if it is justifiable, S_i computes $N = h(ID_i \parallel RPW_i)$ and $v = N \oplus K$.

R3. S_i stores the values $\{v,\ H_i,\ h(.)\}$ into SC_i and sends it to U_i.

R4. On obtaining the smart card from S_i, U_i computes $e = h(ID_i \parallel PW_i \parallel H(B_i))$ and saves e in the smart card, SC_i. So, SC_i of U_i now has the values $\{v,\ e,\ H(.),\ h(.)\}$.

2. Login Phase

Whenever a user U_i wishes to utilize resources from S_i, he must insert his smart card into the terminal and enter his ID_i, PW_i and imprint his biometrics B_i. The following steps are executed:

L1. SC_i computes $e^l = h(ID_i \parallel PW_i \parallel H(B_i))$ and checks if it is same as the saved value, e in the SC_i. If this clears, the smart card executes step L2, else it ceases the login process.

L2. SC_i selects a random number r_1, calculates $RPW_i = h(PW_i \parallel H(B_i))$, $N = h(ID_i \parallel RPW_i)$, $DID_i = ID_i \oplus N$, $C_1 = ID_i \oplus r_1$ followed by $C_2 = h(ID_i \parallel N \parallel r_1)$.

L3. SC_i transmits $\{DID_i,\ v,\ C_1,\ C_2\}$ as the login message to S_i over a open medium.

3. Authentication Phase

On obtaining the login message, $\{DID_i,\ v,\ C_1,\ C_2\}$, the server S_i and U_i advance as follows for mutual authentication.

A1. S_i retrieves $ID_i^l = DID_i \oplus v \oplus K$ and verifies it. If it holds, S_i welcomes that request and executes step A2. Otherwise, it rejects the request and closes this phase.

A2. S_i computes $r_1^l = C_1 \oplus ID_i^l$, $C_2^l = h(ID_i^l \parallel (v \oplus K) \parallel r_1^l)$ and verifies whether $C_2^l = C_2$. If this holds, S_i executes step A3, else it terminates this phase.

A3. S_i generates a random number r_2, calculates $a = r_2 \oplus h(r_1^l \parallel C_2^l)$ and $b = h(C_2^l \parallel r_2 \parallel r_1^l)$. It then delivers $\{a,\ b\}$ to U_i as the authentication message via open channel.

A4. On receiving $\{a,\ b\}$, SC_i computes $r_2^l = a \oplus h(r_1 \parallel C_2)$ and $b^l = h(C_2 \parallel r_2^l \parallel r_1)$. It then verifies if b^l and b are equal or not. If $b^l \neq b$, this phase is terminated. If $b^l = b$, the U_i successfully authenticates S_i. U_i then computes $C_3 = h(r_1 \parallel r_2^l \parallel C_2 \parallel h(ID_i \parallel RPW_i))$ and sends the acknowledgment message, $\{C_3\}$ to S_i via a public channel.

A5. S_i computes $C_3^l = h(r_1^l \parallel r_2 \parallel C_2^l \parallel (v \oplus K))$ and checks it with the received C_3. If this holds, S_i successfully authenticates U_i.

A6. Upon authenticating successfully, SC_i and S_i calculate the shared session key, $SK = h(r_1 \parallel r_2^l \parallel a \parallel b \parallel ID_i)$ and $SK = h(r_1^l \parallel r_2 \parallel a \parallel b \parallel ID_i^l)$ respectively.

4. Password Change Phase
Whenever U_i wishes to alter password, the following steps are executed:

P1. After inserting his smart card into the terminal, U_i enters his ID_i, PW_i and imprints his biometrics B_i. SC_i computes $e^l = h(ID_i \parallel PW_i \parallel H(B_i))$ and compares it with the stored e. If $e^l \neq e$, this phase is terminated; else SC_i executes step P2.

P2. U_i enters the new password PW^{new} and SC_i computes $e^{new} = h(ID_i \parallel PW_i^{new} \parallel H(B_i))$.

P3. SC_i then replaces e with e^{new} and now SC_i contains the values $\{v,\ e^{new},\ H(.),\ h(.)\}$. This completes the password change phase.

5.1.2 Security Limitations in Jung et al.'s scheme

In this section, cryptanalysis of Jung et al.'s scheme, based on the assumptions used in 1.3 has been presented. An adversary who wants to attack the system initially registers himself as a legal user using his own ID_a, PW_a and biometrics B_a. A smart card having sssssvalues, $\{v,\ e,\ H(.),\ h(.)\}$ is issued to him. First, he computes $RPW_a = h(PW_a \parallel H(B_a))$ and $N = h(ID_a \parallel RPW_a)$. Using RPW_a and N, he obtains the server S_i's secret key K as $K = v \oplus N$, where v is obtained from the smart card. So, any adversary who registers himself to the server S_i, can easily obtain the servers' secret key K.

5.1.2.1 User anonymity is not preserved

Suppose an adversary eavesdrops a login request message $\{DID_i,\ v,\ C_1,\ C_2\}$ and intercepts it, he can easily compute the users' identity as $ID_i = DID_i \oplus v \oplus K$, where K is computed as explained above. In other words, if an adversary eavesdrops all the login requests, he can obtain identities of all the users. So, the scheme fails to preserve user anonymity.

5.1.2.2 Incorrect password change phase

In the password change phase, the smart card computes $e^{new} = h(ID_i \parallel PW_i^{new} \parallel H(B_i))$ after the user enters new password, PW_i^{new}. It then replaces e with e^{new} and now the smart card has the values $\{v,\ e^{new},\ H(.),\ h(.)\}$. But the value, $v = N \oplus K$ remains unchanged and , where $N = h(ID_i \parallel RPW_i)$. Once the password PW_i is changed to PW_i^{new}, RPW_i changes to $RPW_i^{new} = h(PW_i^{new} \parallel H(B_i))$. Accordingly, the v changes to $v^{new} = N \oplus K = (h(ID_i \parallel RPW_i^{new})) \oplus K$. If the user wishes to login, he inserts SC_i and enters his ID_i, PW_i^{new} and imprints his biometrics B_i. After verification of password, smart card transmits the login request message $\{DID_i,\ v,\ C_1,\ C_2\}$ to S_i. S_i retrieves $ID_i^l = DID_i \oplus v \oplus K$ and computes $r_1^l = C_1 \oplus ID_i^l$, $C_2^l = h(ID_i^l \parallel (v \oplus K) \parallel r_1^l)$ and verifies whether $C_2^l = C_2$. But, this verification does not hold since $C_2^l = h(ID_i^l \parallel (v^{new} \oplus K) \parallel r_1^l)$ after the change of password. This is because v is not changed to v^{new} after password change. So, authentication phase is terminated. In other words, user will not be allowed to access services from the server once he changes his password. Therefore, password change phase is not accurate.

5.1.2.3 Vulnerable to user impersonation attack

Suppose an adversary intercepts and obtains a login message $\{DID_i,\ v,\ C_1,\ C_2\}$, he can obtain the identity ID_i as explained above in 5.1.2.1. He computes $N = DID_i \oplus ID_i$. He now chooses a random number r_a, computes $C_{1a} = ID_i \oplus r_a$, $C_{2a} = h(ID_i \parallel N \parallel r_a)$ and sends $\{DID_i,\ v,\ C_{1a},\ C_{2a}\}$ to the server S_i. Upon receiving the message $\{DID_i,\ v,\ C_{1a},\ C_{2a}\}$, S_i obtains $ID_i^l = DID_i \oplus v \oplus K$, where K is the servers' secret key. Then it computes $r_a^l = C_{1a} \oplus ID_i^l$ and checks if $C_2^l = h(ID_i \parallel (v \oplus K) \parallel r_a^l)$ equals C_{2a} or not.This condition clearly holds since ID_i and r_a^l are the same ones which were sent by the adversary. So, an adversary has successfully impersonated the user U_i and hence the scheme cannot withstand user impersonation attack.

5.1.2.4 Insecure session key

If an adversary obtains the login message $\{DID_i,\ v,\ C_1,\ C_2\}$ and authentication message $\{a,\ b\}$, and intercepts them, he can compute ID_i and r_1^l as explained above. Now, he computes $r_2^l = a \oplus h(r_1^l \parallel C_2)$, where C_2 is obtained from the login message. Then he can clearly compute the session key SK as $SK = h(r_1^l \parallel r_2 \parallel C_2 \parallel (v \oplus K) \parallel ID_i)$, where v is obtained from the login message and K is obtained as explained in the beginning of this section. So, the session key is not secure.

5.1.2.5 Vulnerable to replay attack

Since none of the random numbers are recorded on the server side during any session, there is no restriction for repetition of random numbers. So, if an adversary intercepts a login message $\{DID_i,\ v,\ C_1,\ C_2\}$ and replays it, S_i retrieves $ID_i^l = DID_i \oplus v \oplus K$ and computes $r_1^l = C_1 \oplus ID_i^l$, $C_2^l = h(ID_i^l \parallel (v \oplus K) \parallel r_1^l)$ to verify whether $C_2^l = C_2$. This verification holds and S_i selects a random number r_2, computes $a = r_2 \oplus h(r_1^l \parallel C_2^l)$ and $b = h(C_2^l \parallel r_2 \parallel r_1^l)$. It then sends the authentication message $\{a,\ b\}$ to the adversary believing him to be the user U_i. In other words, server S_i will authenticate him without any trouble. So, the scheme cannot withstand replay attack.

5.1.2.6 Vulnerable to server masquerading attack

Suppose a user U_i delivers a login message $\{DID_i,\ v,\ C_1,\ C_2\}$ and adversary intercepts it, he obtains ID_i as explained in 4.1 above. Using ID_i, he computes $r_1^l = DID_i \oplus ID_i$. Then he chooses a random number r_2, computes $a = r_2 \oplus h(r_1^l \parallel C_2^l)$, $b = h(C_2^l \parallel r_2 \parallel r_1^l)$ and sends $\{a,\ b\}$ to the user U_i. On receiving the authentication message $\{a,\ b\}$, U_i obtains $r_2^l = a \oplus h(r_1^l \parallel C_2^l)$ and verifies if $h(C_2 \parallel r_2^l \parallel r_1) = b$ holds or not. This condition clearly holds making U_i believe the sender(adversary) to be the authentic server and actually communicates with the adversary. So, the scheme cannot resist server masquerading attack.

5.1.3 Proposed three-factor scheme

In this section, a robust scheme using biometrics has been proposed. All the computations are explained in detail.

1. Registration Phase

   In registration phase, a new user U_i has to register himself to access services from a trusted medical server S. The steps in this phase are as follows. This phase is demonstrated in Fig 5.1.

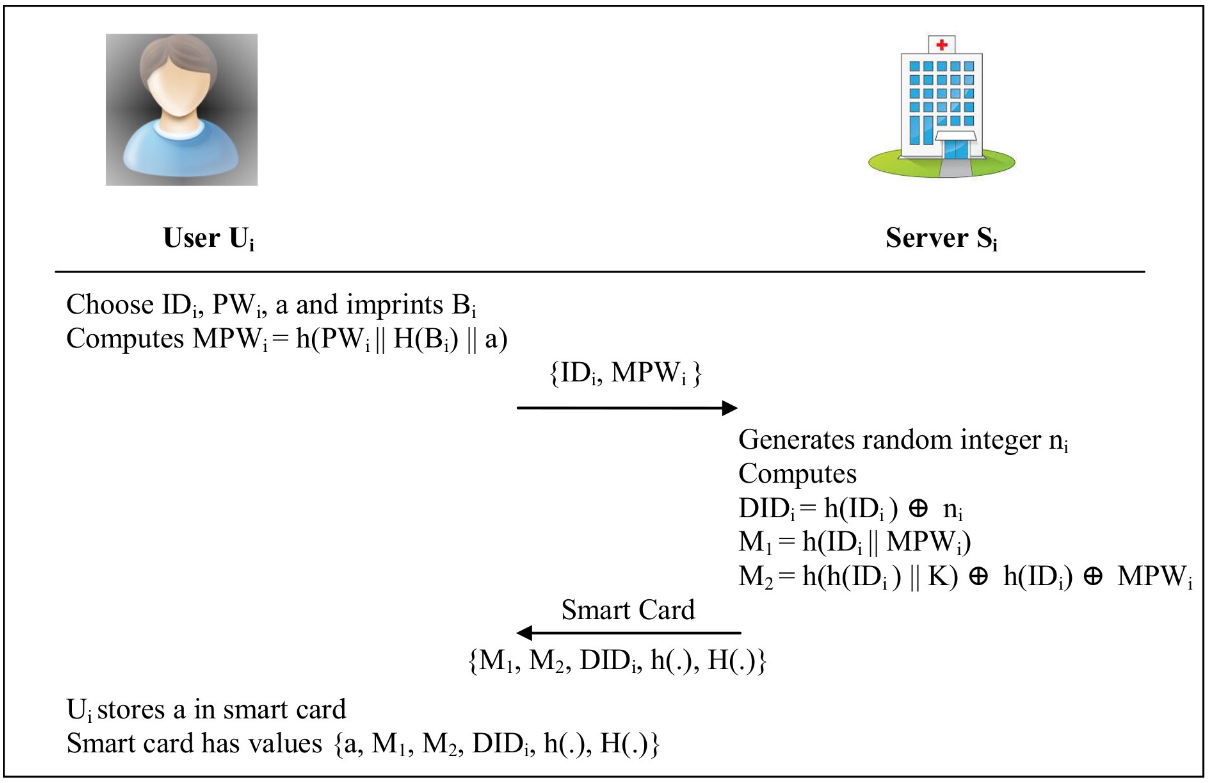
R1. U_i selects his identity ID_i, a password PW_i, biometric B_i and a random number a. U_i computes the masked password $MPW_i = h(PW_i \parallel H(B_i) \parallel a)$ and sends the registration message $\{ID_i,\ MPW_i\}$ to the server S via a secure channel.

R2. On receiving the message $\{ID_i,\ MPW_i\}$ from U_i, S generates a random integer n_i(unique for each U_i) and computes $DID_i = h(ID_i) \oplus n_i$, $M_1 = h(ID_i \parallel$

MPW_i) and $M_2 = h(h(ID_i) \parallel K) \oplus h(ID_i) \oplus MPW_i$, where K is the secret key of the S.

R3. S then issues a smart card to the user U_i containing values $\{DID_i,\ M_1,\ M_2,\ h(.),\ H(.)\}$ via a secure channel.

R4. On receiving the smart card from server, U_i stores a in it and the smart card now has values $\{a,\ DID_i,\ M_1,\ M_2,\ h(.),\ H(.)\}$.



Choose ID_i, PW_i, a and imprints B_i
Computes $MPW_i = h(PW_i \parallel H(B_i) \parallel a)$

$\{ID_i, MPW_i\}$

Generates random integer n_i
Computes
$DID_i = h(ID_i) \oplus n_i$
$M_1 = h(ID_i \parallel MPW_i)$
$M_2 = h(h(ID_i) \parallel K) \oplus h(ID_i) \oplus MPW_i$

Smart Card
$\{M_1, M_2, DID_i, h(.), H(.)\}$

U_i stores a in smart card
Smart card has values $\{a, M_1, M_2, DID_i, h(.), H(.)\}$

Figure 5.1 User registration of the proposed three-factor scheme

2. Login Phase

Suppose any registered user U_i wishes to use services from S, the following steps are performed. This, along with authentication phase is presented in Fig 5.2.

L1. U_i inserts his/her smart card into the card reader of a terminal, inputs $ID_i^{'}$, $PW_i^{'}$ and $B_i^{'}$. The smart card of U_i computes $MPW_i^{'} = h(PW_i^{'} \parallel H(B_i^{'}) \parallel a)$ and $M_1^{'} = h(ID_i^{'} \parallel MPW_i^{'})$. It then verifies if $M_1^{'} = M_1$ holds or not. If it holds, it goes to next step. Otherwise, the login session is terminated.

L2. U_i's smart card generates a random integer, r_1 and acquires the current time stamp T_1. Then it computes $M_3 = M_2 \oplus h(ID_i^{'}) \oplus MPW_i^{'}$, $M_4 = M_3 \oplus r_1$ and $M_5 = h(h(ID_i^{'}) \parallel M_4 \parallel M_3 \parallel T_1)$.

L3. Finally, U_i sends the login request message $\{DID_i, T_1,\ M_4,\ M_5\}$ through a public channel to the server S.

3. Authentication Phase

Upon receiving the login request message $\{DID_i, T_1,\ M_4,\ M_5\}$ from U_i, the server S advances in the following procedure for authentication.

A1. S retrieves the current time stamp T_2 and verifies the freshness of U_i's time stamp T_1 by verifying if $|T_2 - T_1| \leq \delta T$. Also, it does not accept any other request with same credentials in that time interval. If time stamp validity is not verified, the request is rejected; otherwise it obtains $h(ID_i) = DID_i \oplus n_i$ and computes $M_3^{'} = h(h(ID_i) \parallel K)$ and $M_5^{'} = h(h(ID_i) \parallel M_4 \parallel M_3^{'} \parallel T_1)$. It then verifies if $M_5^{'} = M_5$ holds or not. If it holds, U_i is authenticated and authentication process continues. If it does not hold, S rejects the service request message and the authentication phase is terminated.

A2. S generates two random integers, n_i^{new}, r_2 and the time stamp T_3. It then computes $DID_i^{new} = h(ID_i) \oplus n_i^{new}$, $r_1^{'} = M_4 \oplus M_3^{'}$, $M_6 = M_3^{'} \oplus r_2$, $SK = h(h(ID_i) \parallel r_1^{'} \parallel r_2 \parallel M_3^{'})$ and $M_7 = h(M_6 \parallel M_3^{'} \parallel SK \parallel T_3)$. Then it sends the message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$ to the user U_i.

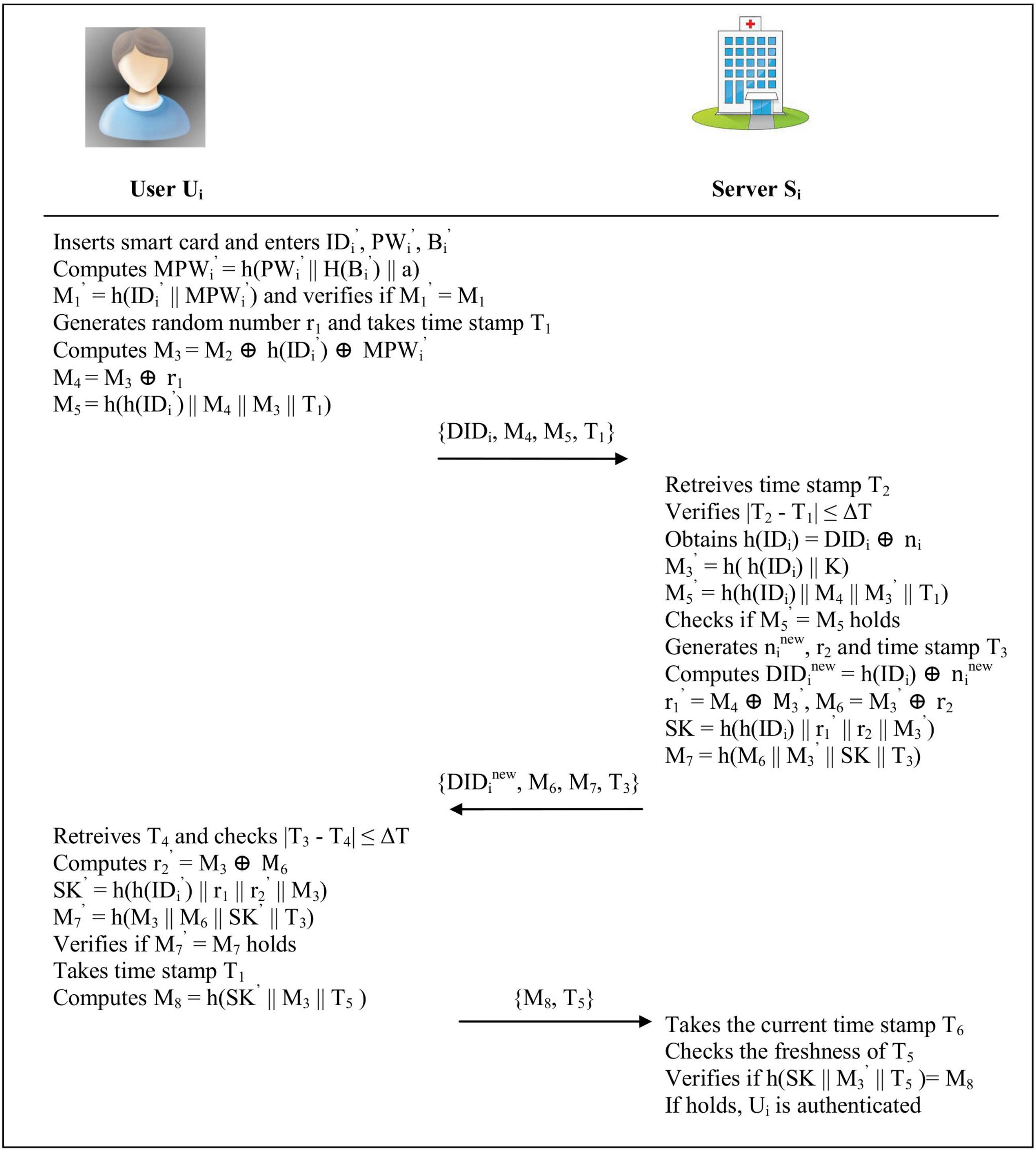
A3. On receiving the message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$ to the user U_i from S, U_i retrieves the current time stamp T_4 and verifies the freshness of S's time stamp T_3 by checking if $|T_3 - T_4| \leq \delta T$. If this does not hold, the request is discarded. Else, it computes $r_2^{'} = M_3 \oplus M_6$, $SK^{'} = h(h(ID_i^{'}) \parallel r_1 \parallel r_2^{'} \parallel M_3)$, $M_7^{'} = h(M_3 \parallel M_6 \parallel SK^{l} \parallel T_3)$ and checks if the condition $M_7^{'} = M_7$ holds or not. If the condition holds, the server S is authenticated and DID_i is replaced by DID_i^{new} in the smart card's memory. Otherwise, the session is terminated.

A4. U_i acquires the current time stamp T_5, computes $M_8 = h(SK^{'} \parallel M_3 \parallel T_5)$ and sends the response message $\{M_8,\ T_5\}$ to S.

A5. On receiving $\{M_8,\ T_5\}$ from U_i, server retrieves the current time stamp T_6 and verifies if $|T_5 - T_6| \leq \delta T$ holds or not. If it does not hold, U_i is not acknowledged; else it computes $M_8^{'} = h(SK \parallel M_3^{'} \parallel T_5)$ and checks if $M_8^{'} = M_8$ holds or not. If it holds, U_i and S believe that they have established the session key SK.

4. Password Change Phase

To modify a password, a user has to insert his smart card in the card terminal and enter his credentials. After this, the following steps are executed.



User U_i

Server S_i

Inserts smart card and enters ID_i', PW_i', B_i'
Computes $MPW_i' = h(PW_i' \| H(B_i') \| a)$
$M_1' = h(ID_i' \| MPW_i')$ and verifies if $M_1' = M_1$
Generates random number r_1 and takes time stamp T_1
Computes $M_3 = M_2 \oplus h(ID_i') \oplus MPW_i'$
$M_4 = M_3 \oplus r_1$
$M_5 = h(h(ID_i') \| M_4 \| M_3 \| T_1)$

$\xrightarrow{\{DID_i, M_4, M_5, T_1\}}$

Retreives time stamp T_2
Verifies $|T_2 - T_1| \leq \Delta T$
Obtains $h(ID_i) = DID_i \oplus n_i$
$M_3' = h(h(ID_i) \| K)$
$M_5' = h(h(ID_i) \| M_4 \| M_3' \| T_1)$
Checks if $M_5' = M_5$ holds
Generates n_i^{new}, r_2 and time stamp T_3
Computes $DID_i^{new} = h(ID_i) \oplus n_i^{new}$
$r_1' = M_4 \oplus M_3'$, $M_6 = M_3' \oplus r_2$
$SK = h(h(ID_i) \| r_1' \| r_2 \| M_3')$
$M_7 = h(M_6 \| M_3' \| SK \| T_3)$

$\xleftarrow{\{DID_i^{new}, M_6, M_7, T_3\}}$

Retreives T_4 and checks $|T_3 - T_4| \leq \Delta T$
Computes $r_2' = M_3 \oplus M_6$
$SK' = h(h(ID_i') \| r_1 \| r_2' \| M_3)$
$M_7' = h(M_3 \| M_6 \| SK' \| T_3)$
Verifies if $M_7' = M_7$ holds
Takes time stamp T_1
Computes $M_8 = h(SK' \| M_3 \| T_5)$

$\xrightarrow{\{M_8, T_5\}}$

Takes the current time stamp T_6
Checks the freshness of T_5
Verifies if $h(SK \| M_3' \| T_5) = M_8$
If holds, U_i is authenticated

Figure 5.2 Login and authentication phases of the proposed three-factor scheme

P1. U_i inserts his/her smart card into the card reader of a terminal, inputs ID_i', PW_i' and B_i'. The smart card of U_i computes $MPW_i' = h(PW_i' \| H(B_i') \| a)$ and $M_1' = h(ID_i' \| MPW_i')$. It then verifies if $M_1' = M_1$ holds or not. If the verification holds, U_i enters a new password PW_i^* and biometric B_i^*; else the request is denied.

P2. The smart card computes $M_1^* = h(ID_i^l \parallel h(PW_i^* \parallel H(B_i^*) \parallel a))$ and $M_2^* = M_2 \oplus MPW_i^{'} \oplus h(PW_i^* \parallel H(B_i^*) \parallel a)$.

P3. Finally, the smart card replaces M_1 and M_2 with M_1^* and M_2^* respectively.

5.1.4 Security analysis of the proposed scheme

In this section, security analysis of the proposed scheme has been discussed. It is shown that this scheme overcomes all the security weaknesses pointed out in Jung et al's scheme.

5.1.4.1 User anonymity is preserved

It can be clearly observed that the identity ID_i of any user is not stored in smart card. During login phase, identity of U_i is sent as dynamic identity DID_i in the login message, $\{DID_i, T_1, M_4, M_5\}$ changes during every session and is replaced by the new identity DID_i^{new}. If an adversary eavesdrops this login message, he cannot obtain ID_i without the knowledge of n_i. Obtaining n_i is impossible since it is unique to every user U_i and is changed during every session by the server. Hence, the user anonymity is preserved in the proposed scheme.

5.1.4.2 Efficient password change phase

In the password change phase of Jung et al.'s scheme, after the user enters the new password, PW_i^{new}, the smart card computes $e^{new} = h(ID_i \parallel PW_i^{new} \parallel H(B_i))$. The smart card then replaces e with e^{new} and now the smart card contains the values $\{v,\ e^{new},\ H(.),\ h(.)\}$. But the value v remains unchanged and $v = N \oplus K$, where $N = h(ID_i \parallel RPW_i)$. Due to this, user U_i will not be able to access services in the next session. But in the proposed scheme, that mistake is rectified by replacing M_1 and M_2 with the newly computed M_1^* and M_2^* respectively. Due to this, the authenticity of the user U_i can be verified by the server S_i during authentication phase without any difficulty.

5.1.4.3 Withstands user impersonation attack

As explained in 5.1.4.1, user identity cannot be obtained by an adversary. In case, he gets hold of a smart card with values $\{a,\ DID_i,\ M_1,\ M_2,\ h(.),\ H(.)\}$, he might get DID_i but not PW_i. To impersonate U_i, an adversary has to generate a valid login message $\{DID_i, T_1,\ M_4,\ M_5\}$. This is not possible without the knowledge of ID_i, PW_i and B_i. As explained in above section, an adversary cannot obtain password of any user U_i even if he eavesdrops the login and authentication messages, $\{DID_i, T_1,\ M_4,\ M_5\}$

and $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$ respectively. So, the proposed scheme withstands user impersonation attack.

5.1.4.4 Secure session key

If an adversary intercepts the login message $\{DID_i, T_1,\ M_4,\ M_5\}$ and authentication message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$, and obtains them, he needs to have the knowledge of ID_i (finding which is impossible as explained in 5.1.4.1), and $M_3^{'}$ to compute the session key, $SK = h(h(ID_i) \parallel r_1^{'} \parallel r_2 \parallel M_3^{'})$. Also, the random numbers $r_1^{'}$ and r_2 are necessary, but it is practically not possible to guess three values, $M_3^{'}$, $r_1^{'}$ and r_2 simultaneously. Hence, session key is secure in the proposed scheme.

5.1.4.5 Withstands replay attack

Replay attack takes place when an adversary intercepts the login message and resends it in order to impersonate the user. In the proposed scheme, if an adversary tries to resend the login message $\{DID_i, T_1,\ M_4,\ M_5\}$, the server first verifies the freshness of the received time stamp by checking if $|T_2 - T_1| \leq \delta T$ holds or not. Intercepting a message from a channel and resending it within that short time interval is not possible. Also, the server rejects any other login request with the same credentials in that time interval. So, even if adversary sends the same message, it will be clearly discarded by the server. Hence, the proposed scheme clearly resists replay attack.

5.1.4.6 Withstands server masquerading attack

To impersonate a server S_i, an adversary needs to have knowledge of n_i which is unique for each U_i. The value n_i corresponding to a user U_i is known only to the server S. It is not possible to obtain n_i from the login message $\{DID_i, T_1,\ M_4,\ M_5\}$. Even though $DID_i = h(ID_i) \oplus n_i$ contains n_i, it is not possible to obtain n_i without the knowledge of ID_i. This makes it impossible for an adversary to create a valid authentication message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$. So, the proposed scheme withstands server impersonation attack.

5.1.4.7 Withstands stolen verifier attack

Stolen verifier attack is an attack wherein an adversary can impersonate a legal user by stealing the verifier of users password stored in the server Chen and Ku (2002). In other words, this attack is possible whenever password verifier of users are stored in the remote server. However, in the proposed scheme, the medical server does not store any data related to password of any user or it does not have verification table. Hence, the proposed scheme easily resists stolen verifier attack.

5.1.4.8 Ensures mutual authentication

In any scheme, it is necessary that user and server can mutually authenticate each other to ensure secure communication. In the proposed scheme, U_i sends the login message $\{DID_i, T_1,\ M_4,\ M_5\}$ from U_i using which the server authenticates U_i as explained in the step A1. On the other hand, server S sends the message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$ to U_i using which U_i authenticates the server which is explained in step A3 of the proposed scheme. Clearly, both the parties authenticate each other and hence the proposed scheme ensures mutual authentication between S and U_i.

5.1.4.9 Withstands password guessing attack

Suppose an adversary gets hold of a smart card with values $\{a,\ DID_i,\ M_1,\ M_2,\ h(.),\ H(.)\}$, he will fail to guess password because $M_1 = h(ID_i \parallel MPW_i)$, where $MPW_i = h(PW_i \parallel H(B_i) \parallel a)$. In the expression of M_1, guessing two unknowns $H(B_i)$ and PW_i simultaneously is infeasible. Also, it is explained in 5.1.4.1 that user anonymity is preserved meaning which an adversary cannot obtain ID_i by any means. If an adversary intercepts the login message $\{DID_i, T_1,\ M_4,\ M_5\}$ during any session, password is still secure in the expressions $M_5 = h(h(ID_i) \parallel M_4 \parallel M_3 \parallel T_1)$, where $M_4 = M_3 \oplus r_1$ and $M_3 = M_2 \oplus h(ID_i^{'}) \oplus MPW_i^{'}$. In these expressions there are multiple unknowns like r_1 (random number chosen by U_i) and $MPW_i = h(PW_i \parallel H(B_i) \parallel a)$. So, password of user is well protected in the login message. In case, adversary intercepts the authentication message $\{DID_i^{new},\ M_6,\ M_7,\ T_3\}$, guessing PW_i is not possible due to the expression $M_7 = h(M_6 \parallel M_3 \parallel SK \parallel T_3)$ in which $SK = h(h(ID_i) \parallel r_1^{'} \parallel r_2 \parallel M_3)$ has two random numbers $r_1^{'}$and r_2. Also, knowledge of $M_3 = M_2 \oplus h(ID_i^{'}) \oplus MPW_i^{'}$ is required to correctly guess the password. But again, guessing multiple unknowns is computationally infeasible. Hence the proposed scheme resists password guessing attack.

5.1.4.10 Withstands insider attack

During registration of a user, if the insider gets information of user's password, he can try to impersonate that user to login other servers using the obtained password. This is possible when any user directly sends password PW_i in the registration request message. In the registration phase of the proposed scheme, user sends $\{ID_i,\ MPW_i\}$ where PW_i is not sent as a plain text but as a hash value in the expression $MPW_i = h(PW_i \parallel H(B_i) \parallel a)$. Hence, the proposed scheme withstands insider attack.

5.1.5 Security proof using BAN logic

Idealized protocol:

$$U \rightarrow S : \langle ID \rangle_{U \overset{M_1}{\leftrightarrow} S}, \langle ID \rangle_{\{U \overset{M_1}{\leftrightarrow} S\}_{r_1}}, (U \overset{SK}{\leftrightarrow} S, \{U \overset{M_1}{\leftrightarrow} S\}_{r2})_{U \overset{M_1}{\leftrightarrow} S}$$
$$S \rightarrow U : (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{M_1}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{M_1} S}, \langle ID \rangle_{\{U \overset{M_1}{\leftrightarrow} S\}_{r_2}}$$

According to the logical postulates, it suffices to prove the following goals:

G1. $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$

G2. $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$

For proof, the following assumptions are made:

A1. $S \mid\equiv U \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A2. $U \mid\equiv S \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A3. $U \mid\equiv \#(r_1)$

A4. $S \mid\equiv \#(r_2)$

A5. $U \mid\equiv U \overset{M_1}{\leftrightarrow} S$

A6. $S \mid\equiv U \overset{M_1}{\leftrightarrow} S$

Analysis:

P1. Since $U \triangleleft (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{M_1}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{M_1} S}$, applying message-meaning rule using A5, we obtain $U \mid\equiv S \mid\sim (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{M_1}{\leftrightarrow} S\}_{r_1})$.

P2. From A3 and P1, application of nonce-verification rule yields $U \mid\equiv S \mid\equiv (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{M_1}{\leftrightarrow} S\}_{r_1})$.

P3. From P2 and A5, we can break the conjunction to obtain $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$ (G1 is achieved).

P4. Since $S \triangleleft (U \overset{SK}{\leftrightarrow} S, \{U \overset{M_1}{\leftrightarrow} S\}_{r_2})_{U \overset{M_1}{\leftrightarrow} S}$, using A6 and applying message-meaning rule, we obtain $S \mid\equiv U \mid\sim (U \overset{SK}{\leftrightarrow} S, \{U \overset{M_1}{\leftrightarrow} S\}_{r_2})$.

P5. From A4 and P4, using nonce-verification rule, we obtain $S \mid\equiv U \mid\equiv (U \overset{SK}{\leftrightarrow} S, \{U \overset{M_1}{\leftrightarrow} S\}_{r_2})$.

P6. Using P5 and A6, we obtain $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$ (G2 is achieved).

From G1 and G2, it can be observed that both the user U_i and server S believe that the session key $SK = h(h(ID_i) \parallel r_1^{'} \parallel r_2 \parallel M_3^{'}) = h(h(ID_i^{'}) \parallel r_1 \parallel r_2^{'} \parallel M_3)$ is shared between them.

5.1.6 Performance comparison

In this section, a detailed comparison of the proposed scheme with Jung et al. (2017), Chen et al. (2018) and Zhang et al. (2018) has been made in terms of computational cost, execution time and performance.

Table 5.1 Computational cost comparison with Jung et al.'s scheme

Phase	Jung et al.	Chen et al.	Zhang et al.	Proposed
Registration	$3T_h$	$4T_h + 1T_{ed}$	$5T_h$	$5T_h$
Login	$4T_h$	$4T_h$	$7T_h$	$4T_h$
Authentication	$9T_h$	$8T_h + 1T_{ed}$	$14T_h$	$8T_h$
Total	$16T_h$	$16T_h + 2T_{ed}$	$26T_h$	$17T_h$

Table 5.1 shows the computational cost comparison. It can be observed that the proposed scheme uses one hash function more than the schemes proposed by Jung et al. and Chen et al. but nine hash functions less than Zhang et al.'s scheme. In addition to this, scheme designed in Chen et al. (2018) uses symmetric key encryption which is more time-consuming when compared to hash functions. But, there is no symmetric key encryption in the proposed scheme.

Table 5.2 Estimated execution time of Jung et al.'s and other schemes(sec)

Phase	Jung et al.	Chen et al.	Zhang et al.	Proposed
Registration	0.0015	0.0107	0.0025	0.0025
Login	0.0020	0.0020	0.0035	0.0020
Authentication	0.0045	0.0127	0.0070	0.0040
Total	0.0080	0.0254	0.0130	0.0085

Table 5.2 presents the comparison of the execution time of the proposed scheme with schemes by Jung et al. (2017), Chen et al. (2018) and Zhang et al. (2018) based on the experiment described in section 2.4. From the table, it is observed that the proposed

scheme uses 0.005s more than Jung et al.'s scheme but this difference is compensated by the performance which is presented in Table 5.3. On the other hand, the schemes designed by Chen et al. and Zhang et al. require 0.0169s and 0.0045s more than the time required by the proposed scheme respectively.

Table 5.3 Comparison of security properties with Jung et al.'s scheme

Security Properties	Jung et al.	Chen et al.	Zhang et al.	Proposed
Provides user anonymity	No	No	Yes	Yes
Resists user impersonation attack	No	No	Yes	Yes
Resists stolen-verifier attack	Yes	Yes	No	Yes
Resists replay attack	No	Yes	Yes	Yes
Secure session key	No	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes
Resists password guessing attack	Yes	No	Yes	Yes
Resists server masquerading attack	No	No	Yes	Yes
Resists insider attack	Yes	Yes	Yes	Yes
Correct password change phase	No	Yes	Yes	Yes

Table 5.3 compares the security properties. It can be noticed that Jung et al.'s scheme has many weaknesses like no user anonymity, user impersonation, server masquerading, insecure session key and inefficient password change mechanism. Also, Mir and Nikooghadam's scheme does not preserve user anonymity. Chen et al.'s scheme fails to preserve user anonymity and cannot withstand user impersonation, password guessing and server masquerading attacks. Zhang et al.'s scheme cannot resist stolen verifier attack since it maintains identity-password verifier table in the remote database.

From all the three tables, it can be observed that the schemes in Jung et al. (2017) and Chen et al. (2018) with one less hash operation than the proposed scheme and using symmetric key encryption are incapable to overcome all the security weaknesses. On the other hand, in spite of using more computations than the proposed scheme, the scheme in Zhang et al. (2018) still fail to withstand all attacks. Based on the observations, it is clear that with comparable number of computations, the proposed scheme overcomes all the mentioned security weaknesses. So, it can be said that the proposed scheme is efficient as well as secure.

5.2 Cryptanalysis and Improvement of Han et al.'s scheme

This part studies another biometric-based scheme designed by Han et al. (2018). In that, they have used ECC concepts along with biometrics. They claim that their scheme preserves user anonymity and resists impersonation attack. But, after cryptanalysis of their scheme, certain security flaws have been identified. They have been explained in detail. To rectify those issues, a robust biometrics-based scheme is proposed. The proposed scheme is analyzed informally as well as using BAN logic. Following this, comparison is presented based on computation and performance to provide an overall picture of the proposed scheme.

5.2.1 Review of Han et al.'s scheme

In this segment, a detailed review of the scheme proposed by Han et al. (2018) is presented. All the phases are explained below.

1. Registration Phase
   To register to a server S_i, any user U_i undertakes the following steps:

   R1. U_i selects identity ID_i, password PW_i, biometrics B_i and generates a random number r. He computes $MP_i = PW_i \oplus H(B_i) \oplus r$ and sends the registration request message $\{ID_i,\ MP_i\}$ to S_i via a secure channel.

   R2. S_i computes $AID_i = h(ID_i \parallel x)$, $K_i = h(AID_i)$ and $V_i = AID_i \oplus MP_i$. Then it generates a random number a to compute $CID_i = E_x(ID_i \parallel a)$ and issues a smart card to U_i with values $\{K_i,\ V_i,\ CID_i,\ h(.),\ H(.)\}$.

   R3. Once the smart card is received, U_i computes $R_i = r \oplus h(ID_i \parallel PW_i \parallel H(B_i))$ and stores R_i in his smart card.
   This completes the registration phase.

2. Login Phase
   In this phase, a legal user U_i with a smart card executes the following steps:

   L1. U_i inserts his smart card into the card reader, enters his identity ID_i, PW_i and biometrics B_i. The smart card computes $r = R_i \oplus h(ID_i \parallel PW_i \parallel H(B_i))$, $MP_i = PW_i \oplus H(B_i) \oplus r$ and $AID_i = V_i \oplus MP_i$. SC_i then checks whether $h(AID_i) = K_i$ holds or not. If it holds, step L2 is executed; else the process is terminated.

   L2. Smart card generates a random nonce $d_u \in Z_p$ and takes the current time stamp T_1 to compute $D = d_uP$, $M_1 = AID_i \oplus D$ and $M_2 = h(AID_i \parallel D \parallel T_1)$. The smart card then transmits the login request message $\{M_1,\ M_2,\ CID_i,\ T_1\}$ to S_i.

3. Authentication Phase

In this phase, smart card of U_i and S_i authenticate each other and then generate a session key, using which they communicate with each other. On receiving $\{M_1,\ M_2,\ CID_i,\ T_1\}$ from U_i, S_i executes the following steps.

A1. S_i takes the current time stamp T_2 to check the freshness of T_1 by verifying whether $|T_1 - T_2| \leq \delta T$ holds or not. If true, S_i retrieves ID_i by decrypting CID_i and computes $AID_i = h(ID_i \parallel x)$. Then it computes $D = AID_i \oplus M_1$ and verifies if $M_2 = h(AID_i \parallel D \parallel T_1)$ holds or not. If it holds, S_i generates random numbers a^l and $d_s \in Z_p$, and computes $E = d_sP$, $CID_i^l = E_x(ID_i \parallel a^l)$, $M_3 = AID_i \oplus E$, $SK = h(AID_i \parallel d_s(D) \parallel CID_i)$ and $M_4 = h(CID_i^l \parallel SK \parallel E \parallel T_3)$, where T_3 is the current time stamp. Then S_i sends the authentication message $\{M_3,\ M_4,\ CID_i^l,\ T_3\}$ to U_i.

A2. On receiving $\{M_3,\ M_4,\ CID_i^l,\ T_3\}$ from S_i, SC_i checks the freshness of T_3. Then it computes $E = M_3 \oplus AID_i$, $SK = h(AID_i \parallel d_u(E) \parallel CID_i)$, $M_4^l = h(CID_i^l \parallel SK \parallel E \parallel T_3)$ and checks whether $M_4^l = M_4$ holds or not. If not, the session is terminated. If it holds, SC_i replaces CID_i with CID_i^l and taking the current time stamp T_4, it computes $M_5 = h(E \parallel SK \parallel T_4)$. Then it sends the message $\{M_5,\ T_4\}$ to S_i.

A3. S_i checks the validity of T_4 and verifies if $h(E \parallel SK \parallel T_4) = M_5$ holds or not. If it holds, S_i authenticates U_i and accepts SK as the session key.

4. Password Change Phase

Suppose U_i wants to alter his password, the following steps are performed.

P1. U_i inserts his smart card and inputs his ID_i, PW_i and B_i.

P2. Smart card computes $r = R_i \oplus h(ID_i \parallel PW_i \parallel H(B_i))$, $MP_i = PW_i \oplus H(B_i) \oplus r$, $AID_i = V_i \oplus MP_i$ and checks if $h(AID_i)$ equals K_i. If they are unequal, the session is closed. If it holds, U_i inputs a new password PW_i^{new}, biometrics B_i^{new} and a new random number r^{new}.

P3. Smart card of U_i computes $MP_i^{new} = PW_i^{new} \oplus H(B_i)^{new} \oplus r^{new}$, $V_i^{new} = AID_i \oplus MP_i^{new}$ and $R_i^{new} = r^{new} \oplus h(ID_i \parallel PW_i^{new} \parallel H(B_i^{new}))$. Finally, it replaces R_i and V_i by R_i^{new} and V_i^{new} respectively.

5.2.2 Security flaws in Han et al's Scheme

Security flaws were found in Han et al.'s scheme. These flaws were revealed by crypt-analyzing their scheme using the assumptions mentioned in 1.3. Each of these issues are elaborated in this part.

Obtaining the master key, x of S_i

Initially, an adversary registers as a legitimate user in the EMR system with his ID_a, PW_a, biometrics B_a and random number r_a. He computes $MP_a = PW_a \oplus H(B_a) \oplus r_a$ and sends $\{ID_a,\ MP_a\}$ to S_i for registration. The server provides a smart card to the adversary having the parameters $\{K_a,\ V_a,\ H(.),\ h(.),\ CID_a\}$ where $AID_a = h(ID_a \parallel x)$, $K_a = h(AID_a)$, $V_a = AID_a \oplus MP_a$ and $CID_a = E_x(ID_a \parallel a_v)$ where a_v is a random number generated by the server S_i. Then he computes $R_a = r_a \oplus h(ID_a \parallel PW_a \parallel H(B_a))$ and stores R_a in the smart card. Now he chooses a value x^l and computes $h(h(ID_a \parallel x^l))$. Then he checks if that value equals K_a or not. If yes, he has predicted x correctly; otherwise he continues the same procedure by choosing different values for x until he obtains the correct value.

5.2.2.1 No user anonymity

Assume that an adversary acquires the smart card of any U_i. This card has values $\{CID_i,\ K_i,\ V_i,\ h(.),\ H(.)\}$. As explained above, he has knowledge of x. He obtains AID_i by repeatedly guessing and checking for different values of AID_i in the expression $K_i = h(AID_i)$. Once he has guessed the correct value of AID_i, he randomly chooses an ID^l and computes $h(ID^l \parallel x)$. Then he verifies whether the computed value equals AID_i or not. If they are equal, adversary has rightly guessed ID_i. If not, he checks for more values of ID_i until he has guessed the correct ID_i. So, the scheme fails to preserve user anonymity.

5.2.2.2 User impersonation attack

Suppose an adversary, who is also a registered user, has the ID_i of a legal user, U_i and also his stored smart card values $\{CID_i,\ K_i,\ V_i,\ h(.),\ H(.)\}$. Since he is registered, he has knowledge of the value P which is fixed during registration. He selects a random number e and computes $D = eP$, $M_1 = AID_i \oplus D$, where $AID_i = h(ID_i \parallel x)$ and $M_2 = h(AID_i \parallel D \parallel T_a)$ where T_a is the present time stamp. He then sends the message $\{M_1,\ M_2,\ CID_i,\ T_a\}$ (CID_i is obtained from the smart card) to S_i. Upon receiving this message, S_i checks the freshness of T_a and computes $AID_i = h(ID_i \parallel x)$, $D = AID_i \oplus M_1$, $M_2 = h(AID_i \parallel D \parallel T_a)$ for verification. Following that, it chooses a random number d_s to compute $E = d_sP$, $CID_i^l = E_x(ID_i \parallel a^l)$, $M_3 = AID_i \oplus E$, $SK = h(AID_i \parallel d_s(D) \parallel CID_i)$ and $M_4 = h(CID_i^l \parallel SK \parallel E \parallel T_a^l)$, where T_a^l is the current time stamp. Then it sends

$\{M_3,\ M_4,\ CID_i^l,\ T_a^l\}$ to the adversary assuming him to be the legal user. So, the adversary has impersonated the legal user successfully. Hence, the scheme cannot prevent user impersonation attack.

5.2.2.3 Server impersonation attack

As explained in 5.1.2.1, an adversary has user identity ID_i from the smart card using which he computes $AID_i = h(ID_i \parallel x)$. Suppose he records the login message $\{M_1,\ M_2,\ CID_i,\ T_1\}$ of U_i, he obtains D as $D = AID_i \oplus M_1$. Then he chooses random numbers a^l and d_{as} to compute $E = d_{as}P$, $CID_i^l = E_x(ID_i \parallel a^l)$, $M_3 = AID_i \oplus E$, $SK = h(AID_i \parallel d_{as}(D) \parallel CID_i)$ and $M_4 = h(CID_i^l \parallel SK \parallel E \parallel T_2)$ where T_2 is the current time stamp. Then he sends the authentication message $\{M_3,\ M_4,\ CID_i^l,\ T_2\}$ to the user U_i. On receiving this, smart card of U_i retrieves the current time stamp T_3 and verifies the freshness of T_2. Then it computes $E = M_3 \oplus AID_i$, $SK = h(AID_i \parallel d_u(E) \parallel CID_i)$, $M_4^l = h(CID_i^l \parallel SK \parallel E \parallel T_3)$ and checks whether $M_4^l = M_4$. This condition is satisfied and taking the current time stamp T_4, the smart card computes $M_5 = h(E \parallel SK \parallel T_4)$. Then it sends the acknowledgment message $\{M_5,\ T_4\}$ to adversary believing him to be the authentic server. Hence, the scheme cannot withstand server impersonation attack.

5.2.2.4 Man-in-the-middle attack

This attack takes place when an adversary acts as a gateway in the communicating channel. The attacker (that is, the "man in the middle") intercepts traffic from the source and forwards it to the destination, thus gaining the ability to modify messages and insert new ones without either party realizing it (Callegati et al., 2009). It has been explained clearly in 5.2.2.2 and 5.2.2.3 that their scheme is vulnerable to user impersonation as well as server impersonation attacks. In other words, an adversary can alter messages anytime whenever he impersonates a user or server without much difficulty. So, the scheme is easily vulnerable to man-in-the-middle attack.

5.2.3 Biometrics-based proposed scheme

This section presents the proposed scheme. This scheme uses biometrics and hash functions. All the phases are explained in detail.

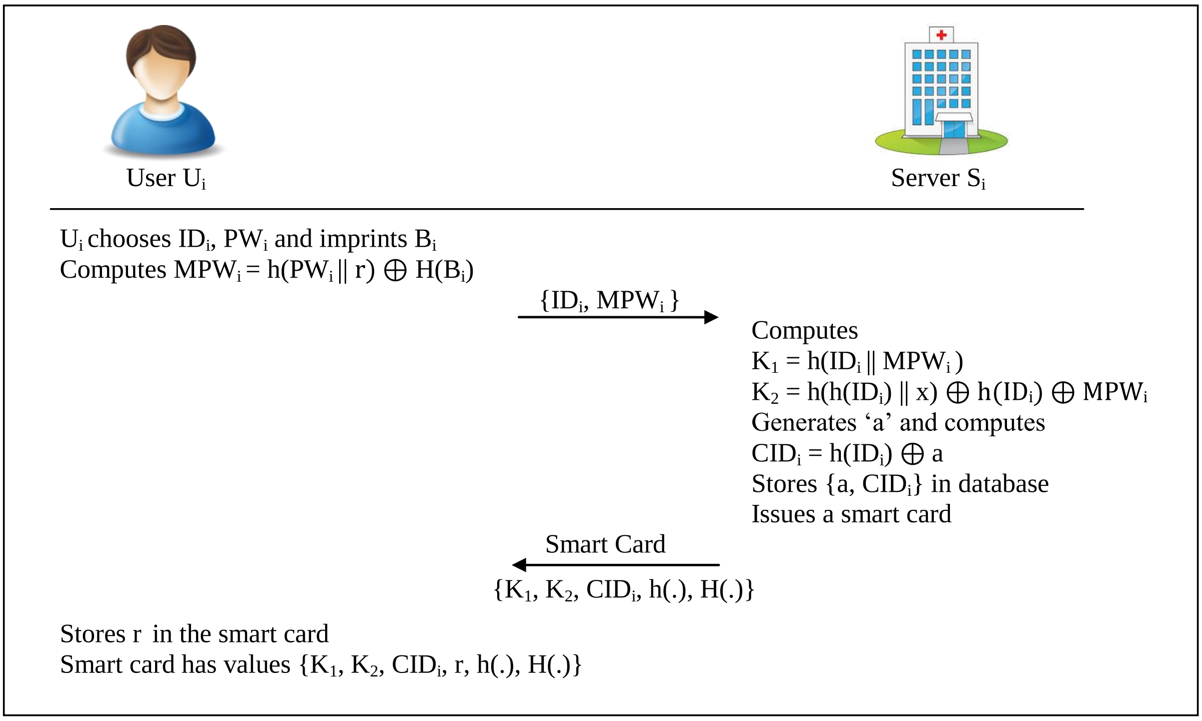
1. Registration Phase
   If any new user U_i wants to register to a remote medical server S_i, the following steps are executed. Fig 5.3 gives the schematic representation of this phase.

R1. U_i chooses his identity ID_i, password PW_i and biometrics B_i. He computes the masked password, $MPW_i = h(PW_i \parallel r) \oplus H(B_i)$, where r is a random number chosen by U_i, and sends the registration request message $\{ID_i,\ MPW_i\}$ to S_i via a secure channel.

R2. S_i computes $K_1 = h(ID_i \parallel MPW_i)$ and $K_2 = h(h(ID_i) \parallel x) \oplus h(ID_i) \oplus MPW_i$ where x is its master key. Then it generates a random number a to compute $CID_i = h(ID_i) \oplus a$. It then stores $\{a,\ CID_i\}$ in its database and issues a smart card to U_i which has the values $\{K_1,\ K_2,\ CID_i,\ h(.),\ H(.)\}$.

R3. Upon receiving SC_i, U_i stores r in it. So, the values in the smart card are $\{r,\ K_1,\ K_2,\ CID_i,\ h(.),\ H(.)\}$.

This completes the registration phase.



U_i chooses ID_i, PW_i and imprints B_i
Computes $MPW_i = h(PW_i \parallel r) \oplus H(B_i)$

Computes
$K_1 = h(ID_i \parallel MPW_i)$
$K_2 = h(h(ID_i) \parallel x) \oplus h(ID_i) \oplus MPW_i$
Generates ‘a’ and computes
$CID_i = h(ID_i) \oplus a$
Stores $\{a, CID_i\}$ in database
Issues a smart card

Stores r in the smart card
Smart card has values $\{K_1, K_2, CID_i, r, h(.), H(.)\}$

Figure 5.3 Registration phase of biometrics based proposed scheme

2. Login Phase

Suppose a user U_i wants to login to a server S_i, the smart card executes the following steps:

L1. U_i installs his smart card into the card reader, enters his identity ID_i^l, PW_i^l and biometrics B_i^l. The smart card computes $MPW_i^l = h(PW_i^l \parallel r) \oplus H(B_i)$, $K_1^l = h(ID_i^l \parallel MPW_i^l)$ and checks whether it equals K_1 or not. If it holds, step L2 is executed; else the process is terminated because either the entered identity ID_i^l or password PW_i^l does not match with the original identity ID_i or password PW_i of U_i. Local password verification takes place in this step.

L2. Smart card computes generates a random nonce r_1 and takes the current time stamp T_1 to compute $M_1 = K_2 \oplus h(ID_i)^l) \oplus MPW_i^l$, $M_2 = M_1 \oplus r_1$ and $M_3 = h(h(ID_i^l) \parallel M_1 \parallel M_2 \parallel T_1)$. It then delivers the login request message $\{T_1,\ M_2,\ M_3,\ CID_i\}$ to S_i.

3. Authentication Phase

The following steps are executed by the server S_i on receiving $\{T_1,\ M_2,\ M_3,\ CID_i\}$ from U_i. Figure 5.4 represents all the steps of login and authentication phases of the proposed scheme.

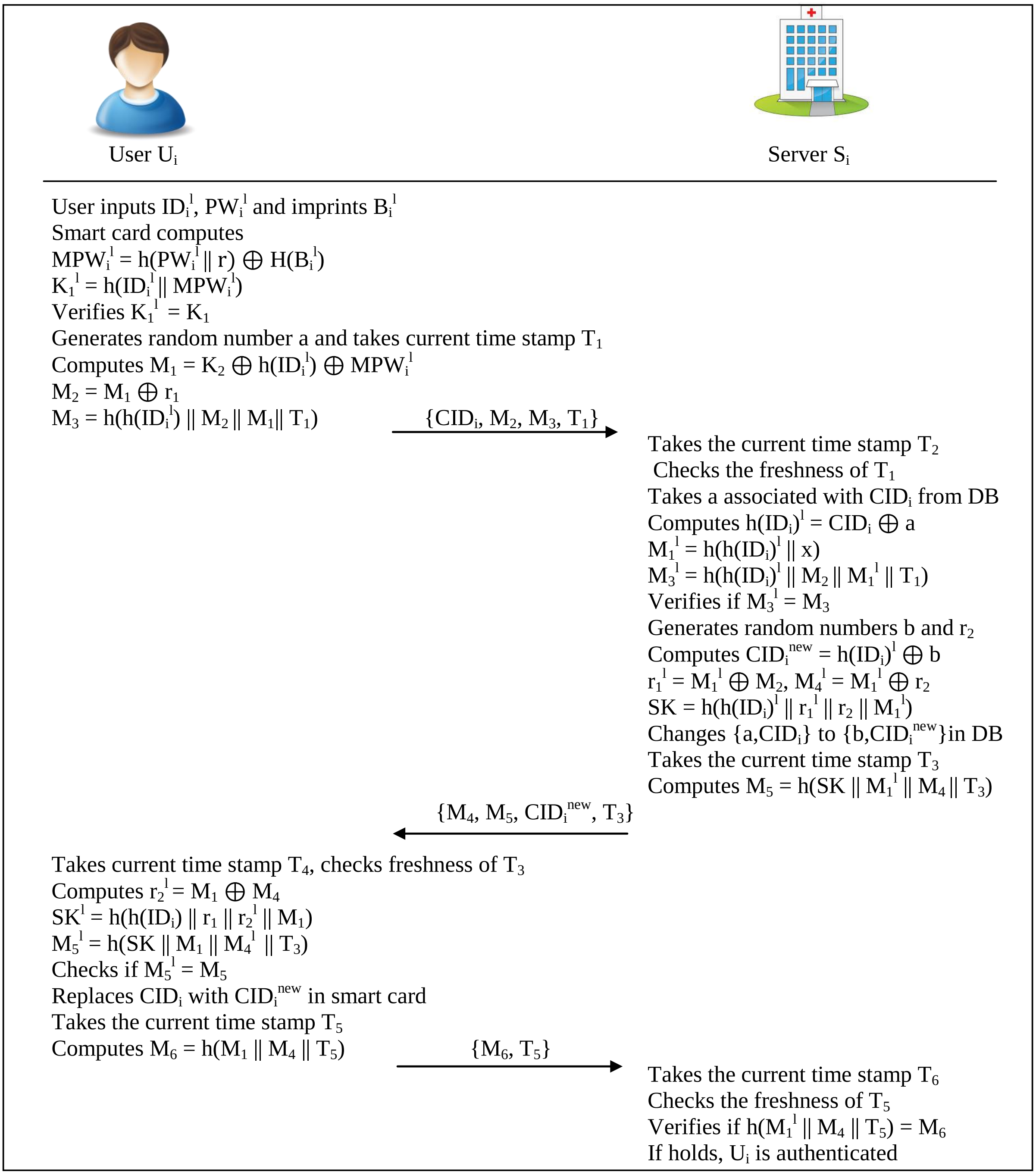
A1. S_i takes the current time stamp T_2 to check the freshness of T_1 by verifying whether $|T_1 - T_2| \leq \delta T$ holds or not. If it does not hold, the session is ceased since the time stamp received from the login message does not come within the required threshold value, thereby resisting replay attack. If true, S_i takes the value a associated with CID_i obtained from the login message and computes $h(ID_i)^l = CID_i \oplus a$, $M_1^l = h(h(ID_i)^l \parallel x)$ and $M_3^l = h(h(ID_i)^l \parallel M_2 \parallel M_1^l \parallel T_1)$. Then it verifies if $M_3^l = M_3$ holds or not. If it holds, S_i generates random numbers b and r_2, and computes $r_1^l = M_2 \oplus M_1^l$, $CID_i^{new} = h(ID_i) \oplus b$ and $M_4 = M_1^l \oplus r_2$. Then it updates the replaces $\{CID_i,\ a\}$ in its database with $\{CID_i^{new},\ b\}$. It further computes $SK = h(h(ID_i^l) \parallel r_1^l \parallel r_2 \parallel M_l^l)$ and $M_5 = h(SK \parallel M_1^l \parallel M_4 \parallel T_3)$, where T_3 is the current time stamp. Then S_i sends the authentication message $\{M_4,\ M_5,\ CID_i^{new},\ T_3\}$ to U_i.

A2. On receiving $\{M_4,\ M_5,\ CID_i^{new},\ T_3\}$ from S_i, smart card of U_i takes the time stamp T_4 and checks the freshness of T_3. Then it computes $r_2^l = M_1 \oplus M_4$, $SK^l = h(h(ID_i) \parallel r_1 \parallel r_2^l \parallel M_1)$, $M_5^l = h(SK^l \parallel M_1 \parallel M_4^l \parallel T_3)$ and checks whether $M_5^l = M_5$ holds or not. If not, the session is terminated. If it holds, smart card of U_i replaces CID_i with CID_i^{new} and takes the current time stamp T_5 to compute $M_6 = h(M_l \parallel M_4 \parallel T_5)$. Then it sends the message $\{M_6,\ T_5\}$ to S_i.

A3. S_i checks the validity of T_5 and verifies if $h(M_1^l \parallel M_4 \parallel T_5) = M_6$ holds or not. If it holds, S_i authenticates U_i and accepts $SK = SK^l$ as the session key.

4. Password Change Phase

Suppose U_i wishes to modify his password or update biometrics, the following



User inputs ID_i^l, PW_i^l and imprints B_i^l
Smart card computes
$MPW_i^l = h(PW_i^l \| r) \oplus H(B_i^l)$
$K_1^l = h(ID_i^l \| MPW_i^l)$
Verifies $K_1^l = K_1$
Generates random number a and takes current time stamp T_1
Computes $M_1 = K_2 \oplus h(ID_i^l) \oplus MPW_i^l$
$M_2 = M_1 \oplus r_1$
$M_3 = h(h(ID_i^l) \| M_2 \| M_1 \| T_1)$

$\xrightarrow{\{CID_i, M_2, M_3, T_1\}}$

Takes the current time stamp T_2
Checks the freshness of T_1
Takes a associated with CID_i from DB
Computes $h(ID_i)^l = CID_i \oplus a$
$M_1^l = h(h(ID_i)^l \| x)$
$M_3^l = h(h(ID_i)^l \| M_2 \| M_1^l \| T_1)$
Verifies if $M_3^l = M_3$
Generates random numbers b and r_2
Computes $CID_i^{new} = h(ID_i)^l \oplus b$
$r_1^l = M_1^l \oplus M_2$, $M_4^l = M_1^l \oplus r_2$
$SK = h(h(ID_i)^l \| r_1^l \| r_2 \| M_1^l)$
Changes $\{a, CID_i\}$ to $\{b, CID_i^{new}\}$in DB
Takes the current time stamp T_3
Computes $M_5 = h(SK \| M_1^l \| M_4 \| T_3)$

$\xleftarrow{\{M_4, M_5, CID_i^{new}, T_3\}}$

Takes current time stamp T_4, checks freshness of T_3
Computes $r_2^l = M_1 \oplus M_4$
$SK^l = h(h(ID_i) \| r_1 \| r_2^l \| M_1)$
$M_5^l = h(SK \| M_1 \| M_4^l \| T_3)$
Checks if $M_5^l = M_5$
Replaces CID_i with CID_i^{new} in smart card
Takes the current time stamp T_5
Computes $M_6 = h(M_1 \| M_4 \| T_5)$

$\xrightarrow{\{M_6, T_5\}}$

Takes the current time stamp T_6
Checks the freshness of T_5
Verifies if $h(M_1^l \| M_4 \| T_5) = M_6$
If holds, U_i is authenticated

Figure 5.4 Login-authentication phases of biometrics based proposed scheme

computations are performed.

P1. U_i inserts his smart card and inputs his ID_i^l, PW_i^l and B_i^l.

P2. The smart card computes $MPW_i^l = h(PW_i^l \| r) \oplus H(B_i)$, $K_1^l = h(ID_i^l \| MPW_i^l)$ and checks whether it equals K_1 or not. If they are not equal, the session is aborted. If equal, U_i enters a fresh password PW_i^{new} and biometrics B_i^{new}.

P3. SC_i computes $MPW_i^{new} = h(PW_i^{new} \parallel r) \oplus H(B_i)^{new}$, $K_1^{new} = h(ID_i \parallel MPW_i^{new})$ and $K_2^{new} = K_2 \oplus MPW_i \oplus MPW_i^{new}$. Finally, it replaces K_1 and K_2 by K_1^{new} and K_2^{new} respectively.

This completes the password change phase.

5.2.4 Security analysis

This section describes the analysis of the proposed scheme. It explains in detail how the proposed scheme is resistant to known security attacks.

5.2.4.1 Preserves user anonymity

It can be clearly observed that identity ID_i of any user U_i is not stored in his smart card. ID_i is not sent directly in the login message $\{T_1, M_2, M_3, CID_i\}$ to S_i as well as the authentication message $\{M_4, M_5, CID_i^{new}, T_3\}$ to U_i. Suppose an adversary gets hold of a smart card and is able to obtain CID_i from it, ID_i still remains unrevealed since $CID_i = h(ID_i) \oplus a$ where the value a is stored in the database and can be known only to the server. So, the proposed scheme preserves user anonymity.

5.2.4.2 Secure against password guessing attack

If an adversary has a smart card with values $\{r, K_1, K_2, CID_i, h(.), H(.)\}$, guessing password is not possible from these values. This is because $K_1 = h(ID_i) \parallel MPW_i$ and $K_2 = h(h(ID_i) \parallel x) \oplus h(ID_i) \oplus MPW_i$. In the first expression, ID_i is not revealed since user anonymity is preserved as explained in 5.2.4.1 and MPW_i has two unknowns (PW_i and B_i) guessing which is computationally infeasible. So password cannot be guessed using smart card values. If adversary eavesdrops login and authentication messages, $\{T_1, M_2, M_3, CID_i\}$ to S_i and $\{M_4, M_5, CID_i^{new}, T_3\}$ to U_i, password is secure. This is valid since the login message values M_2 and M_3 are hashed together with MPW_i and random number r_1. Guessing two unknowns out of a hash value is not possible. Also, from the authentication message, the knowledge of M_4 and M_5 will not reveal PW_i because in $M_4 = M_1^l \oplus r_2$, both the values are not known to the adversary. In $M_5 = h(SK \parallel M_1^l \parallel M_4 \parallel T_3)$, the therm containing PW_i is M_1^l which cannot be known by the adversary. In either case, the proposed scheme resists password guessing attack.

5.2.4.3 Secure against user impersonation attack

User impersonation attack is possible when an adversary is able to create a valid login request and communicate with the server posing to be the legal user. In the proposed scheme, the login message has four values, $\{T_1, M_2, M_3, CID_i\}$. Suppose an adversary eavesdrops this message, he cannot create another valid request since $M_2 = M_1 \oplus r_1$ (where M_1 and r_1 are unknowns) and $M_3 = h(h(ID_i^l) \parallel M_1 \parallel M_2 \parallel T_1)$ (where M_1 is not known). Computing M_1 is not possible since it requires the knowledge of K_2 which, in turn requires the values of users PW_i and B_i and servers master key x, both of which cannot be guessed simultaneously. Assume that an adversary gets hold of a smart card with values $\{r, K_1, K_2, CID_i, h(.), H(.)\}$, he might intercept CID_i, K_1 and K_2 but these values will not contribute in creating a valid login request without the knowledge of MPW_i. So, the proposed scheme resists user impersonation attack.

5.2.4.4 Secure against server impersonation attack

Server impersonation is a process wherein an adversary masquerades a server and communicates with a user making him believe that he is communicating with the authentic server. In the proposed scheme, to impersonate a server, an adversary has to generate a valid authentication message $\{M_4, M_5, CID_i^{new}, T_3\}$. Suppose he has eavesdropped and recorded the previous authentication messages, he cannot use the same values because of the usage of time stamp. To compute correct M_5, adversary needs the session key, $SK = h(h(ID_i)^l \parallel r_1^l \parallel r_2 \parallel M_1^l)$. But computing SK is computationally infeasible as guessing three unknowns r_1^l, r_2 and M_1^l simultaneously is totally impractical. Therefore, server impersonation attack is clearly resisted.

5.2.4.5 Secure against man-in-the-middle attack

This attack is possible when an adversary eavesdrops the messages sent over the communication channel and possibly alters the messages making the user and server believe they are communicating directly to each other. It is described in 5.2.4.3 and 5.2.4.4 that the proposed scheme resists user impersonation and server impersonation attacks respectively. Hence, it is clear that an adversary cannot perform man-in-the-middle attack in the proposed scheme.

5.2.4.6 Secure against stolen-verifier attack

Many a times, the remote server stores hashed passwords to verify the user. Stolen verifier attack takes place when an adversary directly impersonates a user by stealing the saved password verifier Madhusudhan and Mittal (2012). That is, this attack can take place whenever the password verification table is stored in the database during the registration process. But in the proposed scheme, users password is not stored in clear text or hashed form. Also, no verification table is stored in the database. Hence, there is no chance for an adversary to execute a stolen verifier attack in the proposed scheme.

5.2.4.7 Withstands replay attack

Replay attack is when an adversary eavesdrops a login message and sends the same message to server and, the server believes it to be valid login message. To avoid this, the proposed scheme makes use of time stamps and random numbers which keep changing during every interaction. If an adversary does eavesdrop the login request $\{T_1,\ M_2,\ M_3,\ CID_i\}$, he cannot resend the message as a valid request because when the message is replayed, the server S_i first checks the freshness of T_1 by checking $|T_1 - T_*| \leq \delta T$, where T_* is the time stamp when S_i receives the message the replayed message. But because of the threshold value δT, the message will be discarded. Hence, the proposed scheme clearly resists replay attack.

5.2.4.8 Ensures a secure session key

Security of session key in an important part of authentication because user and server communicate with each other using the generated session key. In the proposed scheme, the session key is given by $SK = h(h(ID_i^l) \parallel r_1^l \parallel r_2 \parallel M_l^l)$. This expression is a hash of multiple values, $h(ID_i)$, r_1^l, r_2 and M_1^l. None of the four values are directly available from the smart card, or login and authentication request messages. Using brute force attack to guess four unknowns is practically impossible. As explained in 5.2.4.1, the identity ID_i of U_i cannot be revealed by the adversary. Also, keeping track of the random numbers r_1 and r_2 cannot help the adversary in computing the session key since they change during every session between user and server. So, the session key is secure in the proposed scheme.

5.2.4.9 Provides mutual authentication

From Fig 5.4, it can be clearly noted that U_i and S_i mutually confirm each others' identity. After receiving the login request message $\{T_1,\ M_2,\ M_3,\ CID_i\}$ from U_i, S_i retrieves a from its database, computes $h(ID_i)^l$, M_1^l and M_3^l. It then verifies if M_3^l is same as the received M_3. If they are equal, S_i authenticates U_i.On the other hand, when S_i sends the authentication request message $\{M_4,\ M_5,\ CID_i^{new},\ T_3\}$ to U_i, smart card of U_i computes r_2^l, SK^l and M_5^l. Then it checks if the computed value M_5^l is same as the received M_5. Only if they are same, U_i authenticates S_i. Hence, mutual authentication takes place between the user and server in the proposed scheme.

5.2.4.10 Secure against privileged insider attack

During registration phase, if the user sends his password in a clear text format, the other-side-party can take that password to execute other security attacks for different applications. This can be termed as privileged insider attack. To prevent this, U_i first computes the masked password, $MPW_i = h(PW_i \parallel r) \oplus H(B_i)$ and sends this value instead of sending a clear-text password. Because of the Biometrics B_i and random number r, the insider cannot obtain the real password PW_i. Hence, the proposed scheme withstands privileged insider attack. %endenumerate

5.2.5 Security proof using BAN logic

In this proof, we show that the session key generated during a session is secure, meaning which the communication followed using that session key is safe. The symbols used below are explained in section 1.4.

Idealized protocol:

$U \rightarrow S : \langle ID \rangle_{U \overset{MPW_i}{\leftrightarrow} S}, \langle ID \rangle_{\{U \overset{K_1}{\leftrightarrow} S\}_{r_1}}, (U \overset{SK}{\leftrightarrow} S, \{U \overset{K_1}{\leftrightarrow} S\}_{r_2})_{U \overset{K_1}{\leftrightarrow} S}$

$S \rightarrow U : (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{K_1}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{K_1} S}, \langle ID \rangle_{\{U \overset{K_1}{\leftrightarrow} S\}_{r_2}}$

According to the logical postulates, the following goals should be achieved:

G1. $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$

G2. $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$

To achieve the desired goals, the following assumptions are made:

A1. $S \mid\equiv U \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A2. $U \mid\equiv S \mid\Rightarrow U \overset{SK}{\leftrightarrow} S$

A3. $U \mid\equiv \#(r_1)$

A4. $S \mid\equiv \#(r_2)$

A5. $U \mid\equiv U \overset{K_1}{\leftrightarrow} S$

A6. $S \mid\equiv U \overset{K_1}{\leftrightarrow} S$

Proof:

P1. As $U \triangleleft (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{K_1}{\leftrightarrow} S\}_{r_1})_{U \underset{\leftrightarrow}{K_1} S}$, using message-meaning rule using A5, we see that $U \mid\equiv S \mid\sim (U \overset{SK}{\leftrightarrow} S,\ \{U \overset{K_1}{\leftrightarrow} S\}_{r_1})$.

P2. Now using A3 with P1, application of nonce-verification rule gives $U \mid\equiv S \mid\equiv (U \overset{SK}{\leftrightarrow} S, \{U \overset{K_1}{\leftrightarrow} S\}_{r_1})$.

P3. From A5 and using A2, we can break the conjunction resulting in $U \mid\equiv S \mid\equiv U \overset{SK}{\leftrightarrow} S$.
This proves G1.

P4. Since $S \triangleleft (U \overset{SK}{\leftrightarrow} S, \{U \overset{K_1}{\leftrightarrow} S\}_{r_2})_{U \overset{K_1}{\leftrightarrow} S}$, applying message-meaning rule and using A6, we have $S \mid\equiv U \mid\sim (U \overset{SK}{\leftrightarrow} S, \{U \overset{K_1}{\leftrightarrow} S\}_{r_2})$.

P5. With A4 and P4, using nonce-verification rule, we obtain $S \mid\equiv U \mid\equiv (U \overset{SK}{\leftrightarrow} S, \{U \overset{K_1}{\leftrightarrow} S\}_{r_2})$.

P6. Now from P5 and A6, by breaking the conjunction, we obtain $S \mid\equiv U \mid\equiv U \overset{SK}{\leftrightarrow} S$.
This proves G2.
From G1 and G2, it is proved that both the user U_i and server S believe that the session key $SK = h(h(ID_i)^l \parallel r_1^l \parallel r_2 \parallel M_1^l) = h(h(ID_i) \parallel r_1 \parallel r_2^l \parallel M_1)$ is shared securely between them.

5.2.6 Performance and computational cost comparison

A detailed comparison of Han et al. (2018) with the proposed scheme, Mir et al. (2015) and Amin and Biswas (2015) is presented in this section.

Table 5.4 Computational cost comparison with Han et al.'s scheme

Phase	Han et al.	Mir et al.	Amin et al.	Proposed scheme
Registration	$4T_h + 1T_{ed}$	$6T_h$	$5T_h$	$4T_h$
Login	$4T_h + 1T_{pm}$	$7T_h$	$6T_h + 1T_{ed}$	$4T_h$
Authentication	$8T_h + 2T_{ed} + 1T_{pm}$	$12T_h$	$9T_h + 1T_{ed}$	$8T_h$
Total	$16T_h + 3T_{ed} + 2T_{pm}$	$25T_h$	$20T_h + 2T_{ed}$	$16T_h$

Table 5.4 compares the schemes based on computational cost. Since execution time of concatenation and XOR operations is very less in comparison to that of hash functions and symmetric key encryption/decryption, they have been ignored. It can be seen that the proposed scheme uses same number of hash operations as the scheme in Han et al. (2018) but it neither uses symmetric key encryption nor elliptic curve point multiplication. In comparison with the scheme proposed in Amin and Biswas (2015), the proposed scheme has four hash operations less than their scheme. In case of the scheme

in Mir et al. (2015), the proposed scheme has nine less hash operations thereby reducing the number of required computations.

Table 5.5 Execution time comparison with Han et al.'s scheme(s)

Phase	Han et al.	Mir et al.	Amin et al.	Proposed scheme
Registration	0.01070	0.0030	0.00250	0.002
Login	0.06508	0.0035	0.06608	0.002
Authentication	0.08448	0.0060	0.06758	0.004
Total	0.16026	0.0125	0.13616	0.008

Table 5.5 demonstrates the comparison of execution time of the schemes. Based on the experiment stated in section 2.4, the estimated time of the proposed scheme is computed and compared with other schemes. This clearly shows that the execution time of the proposed scheme is 0.15226s less than the scheme proposed by Han et al.'s scheme. Also, the proposed scheme uses 0.12816s less than Amin et al.'s scheme for execution. In addition to this, the execution time is 0.0045s less than that of Mir et al.'s scheme. With the three compared schemes, it is clear that the proposed scheme takes comparatively very less time to execute, thereby increasing the efficiency of the scheme.

Table 5.6 Performance comparison with Han et al.'s scheme

Security Properties	Han et al.	Mir et al.	Amin et al.	Proposed
Provides user anonymity	No	No	Yes	Yes
Resists user impersonation attack	No	Yes	No	Yes
Resists stolen-verifier attack	Yes	Yes	Yes	Yes
Resists replay attack	Yes	Yes	No	Yes
Secure session key	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes
Resists password guessing attack	Yes	Yes	No	Yes
Resists server impersonation attack	No	Yes	Yes	Yes
Resists privileged insider attack	Yes	Yes	Yes	Yes
Resists man-in-the-middle attack	No	Yes	Yes	Yes

Performance comparison is presented in Table 5.6. From this table, it is clear that the scheme proposed in Han et al. (2018) cannot overcome several attacks. In spite of using symmetric key encryption with hash operations, it is not able to withstand known security attacks. Also, the scheme proposed in Amin and Biswas (2015) is not able to resist few attacks. Mir et al. (2015) also fails to preserve user anonymity. On the other hand, the proposed scheme resists all security weaknesses with relatively less number of computations as can be seen from Table 5.4 and Table 5.5. This proves the performance efficiency of the proposed scheme.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions

The main focus of this chapter is analysis of authentication schemes, in particular the schemes used for TMIS. But prior to this, a detailed explanation is given on TMIS and various security issues. Then, we have considered two-factor as well as three-factor schemes. A wide range of these schemes have been studied (including those using chaotic maps, symmetric key encryption/decryption, ECC). After extensive study of these, a survey was conducted on password practices of several known websites to get a clear picture of the procedure involved in password based authentication. Based on this, weak areas in present practices regarding the choice of passwords of users and various security questions in case the user forgets password were discussed. To overcome these type of issues, several guidelines for improvement have been given. These topics are discussed in chapters 1 and 2.

The core part of this study focuses on various authentication schemes in TMIS. The evolution of authentication methods have been discussed in length for a better understanding of the subject. Several schemes have been studied and based on various factors, this study is divided into three main sections.

Chapter 3 deals with a scheme using chaotic maps. In particular, Li et al.'s scheme is cryptanalyzed and security weaknesses are discussed. These issues are addressed in the proposed scheme. To support this statement, the proposed scheme is analyzed and security proof is provided using BAN logic. Also, comparisons of computations and security properties are presented. It can be seen that the proposed scheme is way more secure than Li et al.'s scheme.

In chapter 4, security of Chen et al.'s scheme is discussed. This is a two-factor scheme which uses hash operations as well as symmetric key encryption technique. This scheme is cryptanalyzed and security flaws were revealed. These are discussed in this chapter, To overcome these flaws, hash functions based scheme is proposed. The security analysis of this scheme shows that it overcomes the flaws found in Chen et al.'s scheme. Computational cost and estimated execution time comparison is presented. This comparison clearly shows the efficiency of the proposed scheme.

Following this, three-factor (biometrics-based) schemes are studied in chapter 5. Since the use of biometrics his increasing day by day, we have studied two schemes in this chapter. In the first part, Jung et al.'s scheme is studied thoroughly. The weaknesses found in their scheme are highlighted and explained in detail. Further, those weaknesses have been rectified in the proposed scheme. To prove this, security analysis and performance comparison is provided. These show that the proposed scheme outsmarts Jung et al.'s scheme. The second part of this chapter discusses another three-factor scheme which is proposed by Han et al.. Even though it uses biometrics, it is different from Jung et al.'s scheme, since Han et al.'s scheme uses ECC as well as symmetric key encryption technique, whereas Jung et al.'s scheme only uses hash operations and biometrics. By cryptanalysis, certain security attacks were disclosed in Han et al.'s scheme. The proposed scheme is designed in such a way that an adversary cannot launch those attacks. This is proved in the security analysis of the proposed scheme. In addition to this, the performance comparison also shows that the proposed scheme is efficient and is suitable for practical implementation.

6.2 Future scope

In future, it is aimed to study Wireless Body Area Network (WBAN) which can help provide better services in TMIS. WBAN provides a platform for inexpensive and continuous health monitoring over the Internet through real-time updates of patients medical records. For example, a BAN in a significant place of a patient can alert the hospital before that patient gets a heart attack by measuring changes in their vital signs. A standard WBAN consists of sensors, a processor, a transceiver and a battery. The main requirements of a WBAN are listed below.

- *Data Quality*: Quality of medical data should be of high standard to ensure best decision-making based on this data.

- *Data Management*: Since huge amount of data is generated by BAN, maintenance and management of these data is crucial.

- *Sensor Validation*: All the sensor readings should be validated to reduce false alarm generation as much as possible.

- *Data Consistency*: Vital patient datasets may be spread over a number of nodes. All this data stored on multiple devices need to be collected and analyzed to upgrade the quality of patient care.

- *Security*: It is of utmost importance to provide secure and limited access to patients sensitive data.

- *Interoperability*: The system must be scalable and must be able to transfer data seamlessly across networks.

- *System Devices*: The sensors must be light-weight, power efficient, low on complexity and easy to use. On the other hand, storage devices must have remote storage options and access to external analysis, tools over the Internet.

- *Cost*: Low-cost BAN are highly expected to ensure that these services are made available to all classes of the society.

- *Constrained Deployment*: WBAN should be wearable and should be able to perform its task without the user realizing it. However, it is equally important that the system should not modify the daily activities of the user.

- *Consistency*: Since a users medical conditions are involved, it is necessary that the sensor readings are accurate to avoid any mishaps.

All the aforementioned factors are necessary, security is of high priority since this is an area that involves networks. Since very little work has been done in this area of WBAN, lot of effort would be required to guarantee secure and accurate WBAN transmission. In addition to this, it is equally important that secure medical data of different users does not get mixed up. Since WBANs are resource-constrained, commonly used security solutions might not be applicable to these. Exploring this area can be a major breakthrough in the field of telemedicine.

Also, we plan to extend our study to blockchain technology. A blockchain is a distributed ledger system that allows transactions globally without third party verification. Key features of this technology are:

- *Decentralized*: There is no central authority that can control the network. In other words, data can be stored, accessed and updated by anyone connected to the network.

- *Transparency*: Stored data is transparent to potential users.

- *Immutable*: Records stored on a a blockchain are reserved forever; data cannot be altered.
- *Anonymity*: Identity of a user remains undisclosed since data is transferred between nodes.
- *Security*: Data stored on the blockchain is hashed cryptographically making it impossible to tamper with.
- *Consensus-driven*: Each block on the network is verified using a consensus algorithm that allows for decision-making for the group of active nodes on the network.

Since records stored on a blockchain are transparent and medical data of a user needs to be kept confidential, finding secure applications for blockchain based healthcare will also be a topic of interest due to the complex structure of blockchain.

BIBLIOGRAPHY

Amin, R. and Biswas, G. (2015). An improved rsa based user authentication and session key agreement protocol usable in tmis. *Journal of Medical Systems*, 39(8):1–14.

Anderson, J. G. (2007). Social, ethical and legal barriers to e-health. *International journal of medical informatics*, 76(5):480–483.

Arora, S., Yttri, J., and Nilsen, W. (2014). Privacy and security in mobile health (mhealth) research. *Alcohol research: current reviews*, 36(1):143–151.

Arshad, H. and Nikooghadam, M. (2014). Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. *Journal of medical systems*, 38(12):1–12.

Arya, K. and Vidwansh, A. (2015). A robust authentication scheme for telecare medicine information systems. *International Journal of Computer Applications*, 123(6):5–10.

Awasthi, A. K. and Srivastava, K. (2013). A biometric authentication scheme for telecare medicine information systems with nonce. *Journal of medical systems*, 37:9964(1–4).

Breaux, T. and Antón, A. (2008). Analyzing regulatory rules for privacy and security requirements. *IEEE transactions on software engineering*, 34(1):5–20.

Burrows, M., Abadi, M., and Needham, R. M. (1989). A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 426(1871):233–271.

Callegati, F., Cerroni, W., and Ramilli, M. (2009). Man-in-the-middle attack to the https protocol. *IEEE Security & Privacy*, 7(1):78–81.

Chang, C. C., Lee, J. S., Lo, Y. Y., and Liu, Y. (2017). A secure authentication scheme for telecare medical information systems. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pages 303–312. Springer.

Chaturvedi, A., Mishra, D., and Mukhopadhyay, S. (2013). Improved biometric-based three-factor remote user authentication scheme with key agreement using smart card. In *International Conference on Information Systems Security*, pages 63–77. Springer.

Chaturvedi, A., Mishra, D., and Mukhopadhyay, S. (2017). An enhanced dynamic id-based authentication scheme for telecare medical information systems. *Journal of King Saud University-Computer and Information Sciences*, 29(1):54–62.

Chen, C. L., Lee, C. C., and Hsu, C. Y. (2012). Mobile device integration of a fingerprint biometric remote authentication scheme. *International Journal of Communication Systems*, 25(5):585–597.

Chen, C. M. and Ku, W. C. (2002). Stolen-verifier attack on two new strong-password authentication protocols. *IEICE Transactions on communications*, 85(11):2519–2521.

Chen, C. M., Xiang, B., Ke, E. W., Wu, T. Y., and Lin, J. C. W. (2018). Improvement of an anonymous and lightweight authentication scheme for tmis. *Journal of Applied Mathematics and Physics*, 6(01):18.

Cohn, S. P. (2006). Privacy and confidentiality in the nationwide health information network. *Online at http://www. ncvhs. hhs. gov/060622lt. htm.*

Council, N. R., Committee, S. S. S., et al. (1990). *Computers at risk: safe computing in the information age*. National Academies Press.

Das, A. K. (2015). A secure and robust password-based remote user authentication scheme using smart cards for the integrated epr information system. *Journal of medical systems*, 39(3):1–14.

Debiao, H., Jianhua, C., and Rui, Z. (2012). A more secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems*, 36(3):1989–1995.

Devaney, R. L., Siegel, P. B., Mallinckrodt, A. J., and McKay, S. (1993). A first course in chaotic dynamical systems: theory and experiment. *Computers in Physics*, 7(4):416–417.

Doel, K. (2013). Scary logins: Worst passwords of 2012 and how to fix them. *SplashData, 2012*.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.

Furnell, S. (2005). Authenticating ourselves: will we ever escape the password? *Network Security*, 2005(3):8–13.

Gehringer, E. F. (2002). Choosing passwords: security and human factors. In *IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No. 02CH37293)*, pages 369–373. IEEE.

Haller, N. (1994). The s/key (tm) one-time password system. In *Symposium on Network and Distributed System Security*, pages 151–157.

Han, L., Tan, X., Wang, S., and Liang, X. (2018). An efficient and secure three-factor based authenticated key exchange scheme using elliptic curve cryptosystems. *Peer-to-Peer Networking and Applications*, 11(1):63–73.

Hsu, C. L. (2004). Security of chien et al.'s remote user authentication scheme using smart cards. *Computer Standards & Interfaces*, 26(3):167–169.

Identifiable, P. (2012). Password hacks show major sites are vulnerable. *Computer Fraud & Security*, 2012(6):1 – 3.

Irshad, A., Sher, M., Nawaz, O., Chaudhry, S. A., Khan, I., and Kumari, S. (2017). A secure and provable multi-server authenticated key agreement for tmis based on amin et al. scheme. *Multimedia Tools and Applications*, 76(15):16463–16489.

Islam, S. H. and Biswas, G. (2013). Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modelling*, 57(11-12):2703–2717.

Jiang, Q., Chen, Z., Li, B., Shen, J., Yang, L., and Ma, J. (2018). Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. *Journal of Ambient Intelligence and Humanized Computing*, 9(4):1061–1073.

Jiang, Q., Ma, J., Ma, Z., and Li, G. (2013). A privacy enhanced authentication scheme for telecare medical information systems. *Journal of Medical Systems*, 37(1):9897(1–8).

Jung, J., Kang, D., Lee, D., and Won, D. (2017). An improved and secure anonymous biometric-based user authentication with key agreement scheme for the integrated epr information system. *PloS one*, 12(1):e0169414(1–26).

Kahn, D. (1996). The codebreakers: The story of secret writing, revised ed. *New York: Scribner*.

Kang, D., Lee, D., Cho, S., Jung, J., and Won, D. (2017). Cryptanalysis and improvement of robust authentication scheme for telecare medicine information systems. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, page 18. ACM.

Khan, M. K., Kumari, S., and Gupta, M. K. (2014). More efficient key-hash based fingerprint remote authentication scheme using mobile device. *Computing*, 96(9):793–816.

Kocarev, L. and Lian, S. (2011). *Chaos-based cryptography: theory, algorithms and applications*, volume 354. Springer Science & Business Media.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential power analysis. In *Advances in cryptology—CRYPTO'99*, pages 789–789. Springer.

Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772.

Lee, N. Y. and Chen, J. C. (2005). Improvement of one-time password authentication scheme using smart cards. *IEICE transactions on communications*, 88(9):3765–3767.

Lee, T. F., Chang, I. P., Lin, T. H., and Wang, C. C. (2013). A secure and efficient password-based user authentication scheme using smart cards for the integrated epr information system. *Journal of medical systems*, 37(3):9941(1–7).

Li, C. T., Lee, C. C., Weng, C. Y., and Chen, S. J. (2016). A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-healthcare systems. *Journal of medical systems*, 40(11):233(1–10).

Li, C. T., Weng, C. Y., Lee, C. C., and Wang, C. C. (2015). A hash based remote user authentication and authenticated key agreement scheme for the integrated epr information system. *Journal of medical systems*, 39(11):144.

Li, X., Wu, F., Khan, M. K., Xu, L., Shen, J., and Jo, M. (2018). A secure chaotic map-based remote authentication scheme for telecare medicine information systems. *Future Generation Computer Systems*, 84:149–159.

Liu, W., Xie, Q., Wang, S., and Hu, B. (2016). An improved authenticated key agreement protocol for telecare medicine information system. *SpringerPlus*, 5(1):555(1–16).

Lovis, C., Baud, R. H., and Scherrer, J.-R. (1998). Internet integrated in the daily medical practice within an electronic patient record. *Computers in biology and medicine*, 28(5):567–579.

Lu, Y., Li, L., Peng, H., Xie, D., and Yang, Y. (2015a). Robust and efficient biometrics based password authentication scheme for telecare medicine information systems using extended chaotic maps. *Journal of medical systems*, 39(6):1–10.

Lu, Y., Li, L., Peng, H., and Yang, Y. (2015b). An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. *Journal of Medical Systems*, 39(3):1–8.

Madhusudhan, R. and Mittal, R. (2012). Dynamic id-based remote user password authentication schemes using smart cards: A review. *Journal of Network and Computer Applications*, 35(4):1235–1248.

Madhusudhan, R. and Nayak, C. S. (2018). An assessment of website user authentication mechanisms. *International Journal of Computing & Information Sciences*, 14(3):14.

Madhusudhan, R. and Nayak, C. S. (2018). An improved user authentication scheme for telecare medical information systems. In *2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 100–105.

Madhusudhan, R. and Nayak, C. S. (2019). A robust authentication scheme for telecare medical information systems. *Multimedia Tools and Applications*, 78(11):15255–15273.

Madhusudhan, R. and Nayak, C. S. (2020). An improved user authentication scheme for electronic medical record systems. *MULTIMEDIA TOOLS AND APPLICATIONS*.

Masuda, N. and Aihara, K. (2002). Cryptosystems with discretized chaotic maps. *Ieee transactions on circuits and systems i: fundamental theory and applications*, 49(1):28–40.

Messerges, T. S., Dabbish, E. A., and Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE transactions on computers*, 51(5):541–552.

Mir, O., van der Weide, T., and Lee, C. C. (2015). A secure user anonymity and authentication scheme using avispa for telecare medical information systems. *Journal of medical systems*, 39(9):89.

Mishra, D., Mukhopadhyay, S., Kumari, S., Khan, M. K., and Chaturvedi, A. (2014). Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce. *Journal of medical systems*, 38(5):1–11.

Mohammed, L., Ramli, A. R., Prakash, V., Daud, M. B., et al. (2004). Smart card technology: past, present, and future. *International Journal of The Computer, the Internet and Management*, 12(1):12–22.

Morris, R. and Thompson, K. (1979). Password security: A case history. *Communications of the ACM*, 22(11):594–597.

Nikooghadam, M. and Zakerolhosseini, A. (2012). Secure communication of medical information using mobile agents. *Journal of medical systems*, 36(6):3839–3850.

Ostad Sharif, A., Abbasinezhad Mood, D., and Nikooghadam, M. (2019). A robust and efficient ecc-based mutual authentication and session key generation scheme for healthcare applications. *Journal of medical systems*, 43(1):10.

Othman, S. B., Trad, A., and Youssef, H. (2014). Security architecture for at-home medical care using wireless sensor network. In *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 304–309. IEEE.

Pinkas, B. and Sander, T. (2002). Securing passwords against dictionary attacks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 161–170. ACM.

Preneel, B. (1993). *Analysis and design of cryptographic hash functions*. PhD thesis, Citeseer.

Qiu, S., Xu, G., Ahmad, H., and Wang, L. (2018). A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems. *IEEE Access*, 6:7452–7463.

Radhakrishnan, N. and Karuppiah, M. (2019). An efficient and secure remote user mutual authentication scheme using smart cards for telecare medical information systems. *Informatics in Medicine Unlocked*, 16:100092.

Rankl, W. and Effing, W. (2004). *Smart card handbook*. John Wiley & Sons.

Safran, C. and Goldberg, H. (2000). Electronic patient records and the impact of the internet. *International Journal of Medical Informatics*, 60(2):77–83.

Schechter, S., Brush, A. B., and Egelman, S. (2009). It's no secret. measuring the security and reliability of authentication via "secret" questions. In *2009 30th IEEE Symposium on Security and Privacy*, pages 375–390. IEEE.

Siddiqui, Z., Abdullah, A. H., Khan, M. K., and Alghamdi, A. S. (2016). Cryptanalysis and improvement of 'a secure authentication scheme for telecare medical information system'with nonce verification. *Peer-to-Peer Networking and Applications*, 9(5):841–853.

Stallings, W. (2006). *Cryptography and network security, 4/E*. Pearson Education India.

Stocksdale, G. (1998). Nsa glossary of terms used in security and intrusion detection. *SANS Institute Resources, http://www. sans. org/newlook/resources/glossary. htm.*

Sutrala, A. K., Das, A. K., Odelu, V., Wazid, M., and Kumari, S. (2016). Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems. *Computer methods and programs in biomedicine*, 135:167–185.

Takeda, H., Matsumura, Y., Kuwata, S., Nakano, H., Sakamoto, N., and Yamamoto, R. (2000). Architecture for networked electronic patient record systems. *International journal of medical informatics*, 60(2):161–167.

Tan, Z. (2014). A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. *Journal of medical systems*, 38(3):16.

Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M., and Sands, D. Z. (2006). Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association*, 13(2):121–126.

Tsai, C. S., Lee, C. C., and Hwang, M. S. (2006). Password authentication schemes: Current status and key issues. *IJ Network Security*, 3(2):101–115.

Tzu, C. Y., Hsiao Yun, S., and Hwang, J. J. (2002). A secure one-time password authentication scheme using smart cards. *IEICE Transactions on Communications*, 85(11):2515–2518.

Uslu, A. M. and Stausberg, J. (2008). Value of the electronic patient record: an analysis of the literature. *Journal of biomedical informatics*, 41(4):675–682.

van Ginneken, A. M. (2002). The computerized patient record: balancing effort and benefit. *International journal of medical informatics*, 65(2):97–119.

Wayman, J., Jain, A., Maltoni, D., and Maio, D. (2005). *An introduction to biometric authentication systems*. Springer.

Wazid, M., Zeadally, S., Das, A. K., and Odelu, V. (2016). Analysis of security protocols for mobile healthcare. *Journal of medical systems*, 40(11):229.

Wei, J., Hu, X., and Liu, W. (2012). An improved authentication scheme for telecare medicine information systems. *Journal of medical systems*, 36(6):3597–3604.

Wen, F. (2014). A more secure anonymous user authentication scheme for the integrated epr information system. *Journal of medical systems*, 38(5):42.

Wen, F. and Guo, D. (2014). An improved anonymous authentication scheme for telecare medical information systems. *Journal of medical systems*, 38(5):26(1–7).

Wu, F. and Xu, L. (2013). Security analysis and improvement of a privacy authentication scheme for telecare medical information systems. *Journal of medical systems*, 37(4):9958(1–9).

Wu, F., Xu, L., Kumari, S., and Li, X. (2015). A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. *Computers & Electrical Engineering*, 45:274–285.

Wu, Z. Y., Chung, Y., Lai, F., and Chen, T. S. (2012a). A password-based user authentication scheme for the integrated epr information system. *Journal of medical systems*, 36(2):631–638.

Wu, Z. Y., Lee, Y. C., Lai, F., Lee, H. C., and Chung, Y. (2012b). A secure authentication scheme for telecare medicine information systems. *Journal of medical systems*, 36(3):1529–1535.

Xie, Q., Zhang, J., and Dong, N. (2013). Robust anonymous authentication scheme for telecare medical information systems. *Journal of medical systems*, 37(2):9911.

Xiong, H., Tao, J., and Yuan, C. (2017). Enabling telecare medical information systems with strong authentication and anonymity. *IEEE Access*, 5:5648–5661.

Yeh, H. L., Chen, T. H., Hu, K. J., and Shih, W. K. (2013). Robust elliptic curve cryptography-based three factor user authentication providing privacy of biometric data. *IET Information Security*, 7(3):247–252.

You, I. and Jung, E. S. (2006). A light weight authentication protocol for digital home networks. In *International Conference on Computational Science and Its Applications*, pages 416–423. Springer.

Zhang, L., Zhang, Y., Tang, S., and Luo, H. (2018). Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement. *IEEE Transactions on Industrial Electronics*, 65(3):2795–2805.

Zhou, X. and Kalker, T. (2010). On the security of biohashing. In *Media Forensics and Security II*, volume 7541, page 75410Q. International Society for Optics and Photonics.

Zhu, Z. (2012). An efficient authentication scheme for telecare medicine information systems. *Journal of medical systems*, 36(6):3833–3838.

PUBLICATIONS

Journals

1. Madhusudhan, R., and Chaitanya S. Nayak. "An improved user authentication scheme for electronic medical record systems." Multimedia Tools and Applications 79.29-30 (2020): 22007-22026.

2. Madhusudhan, R., and Chaitanya S. Nayak. "A robust authentication scheme for telecare medical information systems." Multimedia Tools and Applications 78(11) (2019): 15255-15273.

3. Madhusudhan, R., and Chaitanya S. Nayak. "An Assessment of Website User Authentication Mechanisms." International Journal of Computing & Information Sciences 14(3) (2018): 14.

4. R. Madhusudhan and Chaitanya S. Nayak, "An Improved Three-factor User Authentication Scheme for TMIS", Cryptologia (*Communicated*).

Conference

Madhusudhan, R., and Chaitanya S. Nayak. "An Improved User Authentication Scheme for Telecare Medical Information Systems." In 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), pp. 100-105. IEEE, 2018.

BIODATA

Name	:	Chaitanya Sadanand Nayak
Email	:	chaitanyasnayak19@gmail.com
Date of Birth	:	19th Nov 1991.
Permanent address	:	Chaitanya Sadanand Nayak, D/o Sadanand Nayak , 5-42(9)(1), "Ranga Nivas", Kalyan Nagar 4th cross, Kukkikatte, Udupi-576 101.

Educational Qualifications :

Degree	**Year**	**Institution / University**
B.Sc. Computer Science	2012	Dr. G. Shankar GWFGC & PG Study Centre, Udupi. Mangalore University.
M.Sc. Mathematics	2014	Dr. G. Shankar GWFGC & PG Study Centre, Udupi. Mangalore University.