Design and Analysis of a Secure Three Factor User Authentication Scheme Using Biometric and Smart Card

by

Hasan Muhammad Kafi

MASTER OF SCIENCE IN INFORMATION AND COMMUNICATIN TECHNOLOGY

Institute of Information and Communication Technology
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

2017

The thesis titled "**Design and Analysis of a Secure Three Factor User Authentication Scheme Using Biometric and Smart Card**" submitted by Hasan Muhammad Kafi, Student Id. 0411312019, and Session April, 2011, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information and Communication Technology on June 17, 2017.

BOARD OF EXAMINERS

1. Dr. Hossen Asiful Mustafa Assistant Professor IICT, BUET, Dhaka	Chairman (Supervisor)
2. Dr. Md. Rubaiyat Hossain Mondal Associate Professor IICT, BUET, Dhaka	Member
3. Prof. Dr. Md. Saiful Islam Director and Professor IICT, BUET, Dhaka	Member (Ex- officio)
4. Dr. Md. Mamun-or-Rashid Professor Dept. of CSE, University of Dhaka, Dhaka-1000	Member (External)

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Hasan Muhammad Kafi

DEDICATION

I dedicate this thesis to the almighty Allah, the most gracious, the most merciful.

Table of Contents

List of Tables and Figures

List of Abbreviations of Technical Symbols and Terms

SPA	Simple Power Analysis
DPA	Differential Power Analysis
B_i	Biometric Key
P_i	Auxiliary Information
MD5	Message Digest Algorithm 5
SHA-1	Secure Hash Algorithm 1
SHA-2	Secure Hash Algorithm 2
SHA-3	Secure Hash Algorithm 3
AES	Advanced Encryption Standard
RAM	Random Access Memory
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDHP	Elliptic Curve Diffie-Hellman Problem
R	The Trusted Registration Center
S_i	The Server
C_i	The User
A_i	The Attacker
ID_i	Identity of the User C_i
PW_i	Password of the User C_i
P, n	Two Large Prime Number
F_p	A Finite Field
$E_p(a, b)$	An Elliptic Curve Defined on Finite Field F_p with Prime Number Order n
P	A point on elliptic curve $E_p(a, b)$ with order n
$h(.)$	A Secure Hash Function
X_s	The Master Secret Key
R_c	A Secret Number Chosen by C_i
R_s	A Secret Number Chosen by S_i
$\parallel$	Message Concatenation Operator
$\oplus$	Exclusive-OR Operator
(Gen, Rep)	Pair of Procedure of a Fuzzy Extractor
K	A Random Number
SK	Shared Secret Key
SID_i	Identity of the Server S_i
PW_{ni}	New Password Chosen by C_i
R_{cont}	Recovery Contact of C_i
R_{n1}, R_{n4}-R_{n8}, W	Secret Random String Chosen by R
R_{n2}	A Secret Random String Chosen by C_i
R_{n3}	A Secret Random String Chosen by S_i
RK_{aes}	AES Private Key of R
SK_{aes}	AES Private Key of S_i
$E_{aes}(.)$	AES Encryption Function

$D_{aes}(.)$	AES Decryption Function
K_{ses}	Session Key
$Stat$	Status Message
Apache	World's Most Used Web Server Software
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JavaScript	A High Level Programming Language
MySql	An Open-Source Relational Database Management System
PHP	HyperText Preprocessor: A Widely-used Open Source General-purpose Scripting Language
Jquery	A Fast, Small, and Feature-rich JavaScript Library
Bootstrap	A Sleek, Intuitive, and Powerful Mobile First Front-end CSS Framework
Font-awesome	A Set of Font Icons, Specially Designed for Websites and Applications

Acknowledgement

In the name of almighty Allah, the most gracious, the most merciful. All the praises and thanks to him for providing me the opportunity to complete this thesis. I would like to express my sincere gratitude to my supervisor and the chairman of the board of examiners Dr. Hossen Asiful Mustafa, Assistant Professor, IICT, BUET for the continuous support of the learning process of this master thesis, for his patience, motivation, and immense knowledge. Furthermore, I would like to convey my thanks to all other examiners of the board, the staffs and the teachers of IICT for the cordial and friendly support during this work. I would like to thank my institute IICT, BUET for giving me this opportunity. I also express my deep gratefulness to so many people who have helped me during this work by their valuable lecture, time, hospitality and cooperation. Last but not the least; I would like to thank my parents, my brother and my wife for supporting me spiritually throughout writing this thesis and my life in general.

Abstract

Password security can no longer provide enough security in the area of remote user authentication. Despite taking numerous attempts to enhance the security of password based system, the attackers are still able to steal passwords. This is mostly due to the user's habit of using password. Most of the users use weak passwords, reuse the same password in several accounts that causes domino effect, store these passwords and reset them frequently. Considering these security drawbacks, researchers are trying to find solution with multifactor remote user authentication system. Some of them have proposed remote user authentication schemes using smart card alongside password. However, some of the schemes have their own drawbacks and are unable to provide proper security to the users. Recently, three factor remote user authentication using biometric and smart card alongside password has drawn a considerable attention of the researchers. Researchers have proposed several remote user authentication schemes. However, most of those schemes have security flaws. They are vulnerable to one or more attacks like user impersonation attack, server masquerading attack, password guessing attack, insider attack, denial of service attack, forgery attack, etc. Moreover, most of them are unable to provide mutual authentication, session key agreement and password, or smart card recovery system. Considering these drawbacks, a secure three factor user authentication scheme using biometric and smart card is proposed in this thesis. Besides registration and authentication, our scheme has mechanisms for password and smart card recovery. Through security analysis, we show that our proposed scheme can overcome drawbacks of existing systems and ensure high security in remote user authentication.

CHAPTER 1
Introduction

Today, most authentication systems rely on passwords. A password is a word or string of characters used for user authentication to prove identity or access approval to gain access to a resource (example: an access code is a type of password), which is to be kept secret from those not allowed access. So, a numerous attempts [1-4] have been taken to protect password from being compromised. However, the attackers are still able to steal password despite those attempts. This is mostly due to the user's habit of using password. Most of the users use weak passwords [5]. They also reuse the same password in several accounts that causes domino effect [6]. According to [5], the average user has 6.5 passwords, each of which is shared across 3.9 different websites. Each user types an average of 8 passwords per day with an average bit-strength of 40.54 bits and has approximate 25 web accounts that require passwords to access. It also mentioned a large number of passwords chosen by the users only contain lowercase letters unless force to do otherwise. Such weak passwords can be guessed by the attacker easily. Increasing password strength may be a solution but it cannot increase the security by a large degree. The strength of password can be increased by mixing up the lowercase letters, uppercase letters, digits, signs and special characters. But, such kinds of passwords are difficult to remember. It reduces the security of a system because i) users might need to store these passwords, ii) users will need to reset them frequently, and iii) users are more likely reuse the same password. According to [5], at least 1.5% of Yahoo users forget their passwords per month.

For these reasons, several methods like password management software, graphical password schemes, cognitive authentication scheme, one time passwords, hardware tokens, phone aided schemes, biometrics, etc, are proposed to replace password [7]. However, none of them provides good usability, deploy-ability and strong security at the same time [7]. In recent years, several smart card based schemes have been proposed [8-10]. But, all of them have several weaknesses.

Recently, biometric and smart card based user authentication schemes along with password have drawn considerable amount of attention of researchers [11-16]. The biometric keys are based on physiological and behavioral characteristics of persons such as fingerprints, faces, irises, hand geometry, and palm-prints, etc. The advantages of using biometric key are given below:

- They cannot be forgotten or lost.
- They cannot be copied or shared easily.
- They are extremely hard to forge or distribute.
- They cannot be guessed easily.
- They prevent non-repudiation.

Considering these advantages, researchers develop their schemes to enhance the security of remote user authentication system. But, all of their schemes have several weaknesses. In particular, the scheme proposed by Li et al. [16] is vulnerable to attacks like user impersonation attack, server masquerading attack, password guessing attack, insider attack, denial of service attack, forgery attack, etc. The other proposed schemes are also vulnerable to one or more of the above mentioned attacks. Additionally, most of them are unable to provide mutual authentication, session key agreement and password, or smart card recovery system.

In this thesis, we propose a three factor user authentication scheme using biometric and smart card that can resist almost all the above mentioned attacks and is also able to provide mutual authentication, session key agreement and password, or smart card recovery system.

1.1 Research Objective

The objective of this thesis is to design and analyze a three factor user authentication system. To achieve this objective, we have the following aims:

- To design a three factor remote user authentication scheme using biometric and smartcard to overcome the weaknesses of existing systems
- To provide a detail security analysis of the proposed scheme
- To show the comparison of the proposed scheme with existing works, and
- To implement the proposed scheme for simulation

Successful completion of this research work will result in a secure three factor remote user authentication scheme using biometric and smartcard.

1.2 Outline of Methodology

At first, we identified the weakness of existing systems. Then, several phases of the proposed scheme is designed such as server registration phase, user registration phase, login and authentication phase, password change phase, password recovery phase, smart card recovery phase, etc. In the proposed scheme, biometric key along with password and smart card for remote user authentication is used.

Biometric cryptosystem and cancelable biometrics represent emerging technologies to release biometric keys as well as provide privacy to biometric templates. Strong biometric keys can be generated from biometric templates using fuzzy extractor. The fuzzy extractor can generate uniform randomness from the templates close to original. Moreover, strong biometric keys can be generated form biometric templates using cancelable biometric technology. Also, biometric keys can be generated from multiple modalities. These technologies can release biometric keys of different bit length. An efficient biometric key generation technique among them can be used in our scheme after further analysis.

Advanced Encryption Standard (AES) is used for providing the security of data in the database and smart card. AES is a symmetric key algorithm where a single key named private key is used for both encryption and decryption purpose. The performance of AES is convincing and it also requires low RAM. Therefore, it can be implemented from 8 bit smart card to high performance computer. The strength of hash function will be used to secure the messages in several phases. The examples of hash function are MD5, SHA-1, SHA-2 family, SHA-3 family, etc. The MD5 and SHA-1 are no longer recommended due to security reason. Either SHA-2 family or SHA-3 family would be used in our scheme.

After designing the authentication scheme, security analysis of the scheme is done to show how the scheme can prevent password guessing attack, secret key stealing, user impersonation attack, server masquerading attack, replay attack, denial of service attack, forgery attack, etc.

Then, we compare our proposed scheme with existing schemes in terms of several security and functionality features. The proposed scheme is simulated using HTML, CSS, JAVASCRIPT, PHP and MYSQL.

1.3 Organization of Thesis

The rest of thesis is organized as follows. In chapter 2, necessary background of the thesis is briefly discussed. Then, chapter 3 describes few related works. An overview of Li et al.'s scheme is given in chapter 4. In chapter 5, our proposed scheme is presented with the security analysis of the proposed scheme in chapter 6. The implementation and simulation of the proposed scheme is discussed in chapter 7. Finally, we draw our conclusion and present the future work scope in chapter 8.

1.4 Summary

In this chapter, we introduce the problem of remote user authentication scheme and recent research trends for remote user authentication. Then, we discussed the objective, methodology and organization of our proposed thesis.

CHAPTER 2
Background

2.1 Types of Attacks on an Authentication System

Internet Engineering Task Force (IETF) defines attack in RFC 2828 as: “an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system”. An attacker can conduct several types of attacks on a user authentication system. Some of them are discussed in the following.

2.1.1 Password Guessing Attack

The password guessing attack [15] is an attack where the attacker tries to guess the victim’s password through several techniques such as dictionary attack. The dictionary attack is a kind of attack where the attacker tries to match the password from a collection of passwords. At first, the attacker tries to break into relatively insecure servers and gather passwords from those servers.

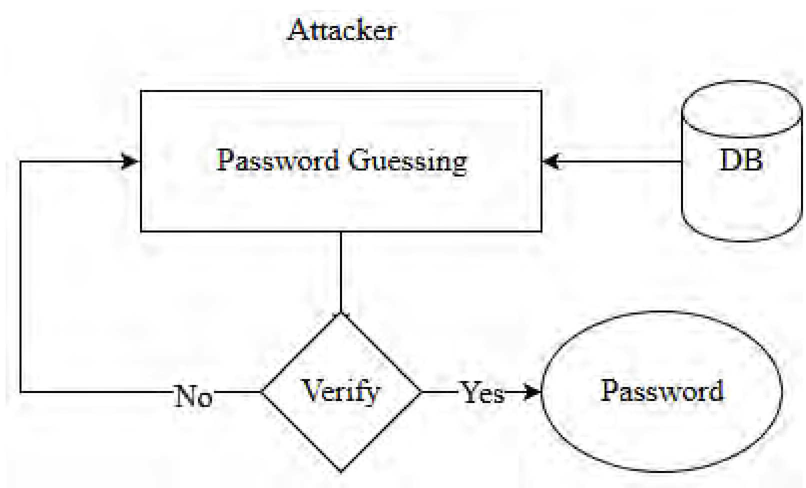

Fig. 2.1. Password guessing attack

By collecting all those passwords, the attacker tries to build a massive database that contains different types of passwords. The attacker uses this database to conduct password guessing attack and may try to login to a server directly using those passwords randomly or use a mathematical approach by collecting any information that contains password or any other methods. Fig. 2.1 shows how a password guessing attack can be launched.

2.1.2 User Impersonation Attack

It is a kind of attack where the attacker acts as a legitimate user and tries to gain access to the server. To impersonate as the legitimate user, an attacker attempts to make a forged login request message which can be authenticated to the server [15]. At first, the attacker collects information that is required to generate a valid login request. He also collects several login messages by eavesdropping into the insecure channel.

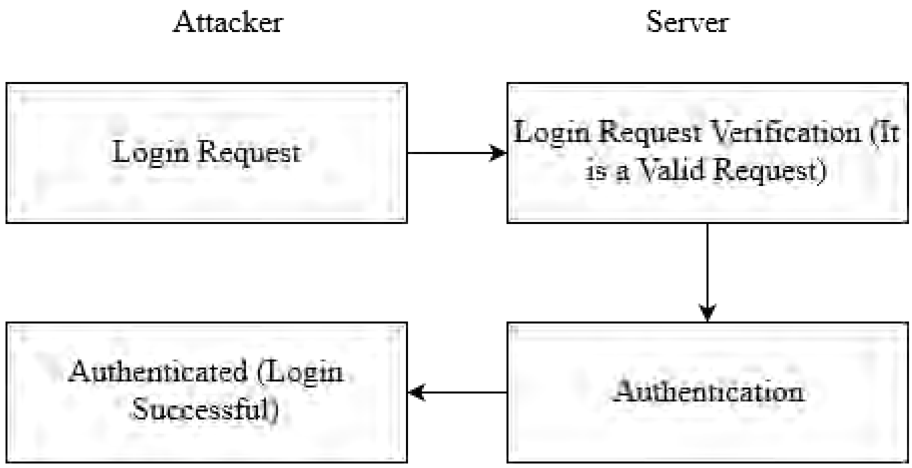

Fig. 2.2. User impersonation attack

The attacker analyzes the login messages and tries to figure out the format or structure of messages. After figuring out the format/structure, he tries to construct a legal login message. Then, this message is used to gain access to the desired server. Fig. 2.2 shows how user impersonation attack can be launched.

2.1.3 Server Masquerading Attack

It is kind of attack where the attacker fakes the identity of server to gain the access of the messages of a legitimate user. To masquerade as the legitimate server, an attacker attempts to make a forged reply message which can be masqueraded to the user when receiving the user's login request message [15].

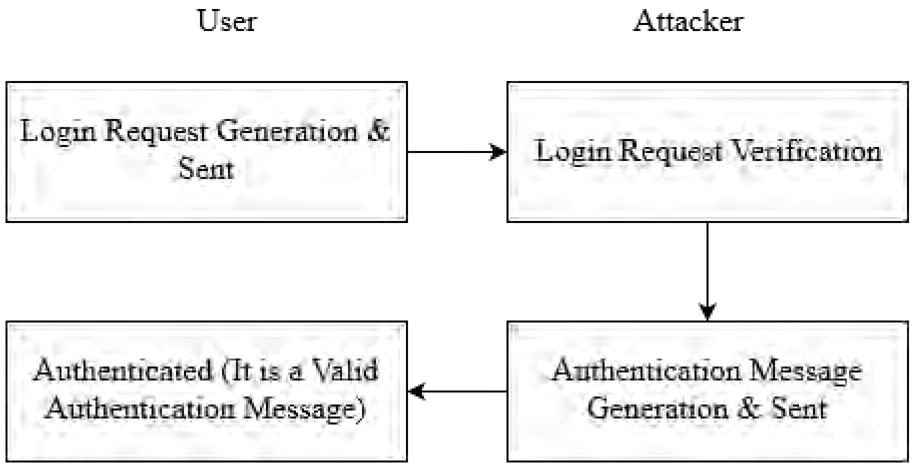

Fig. 2.3. Server masquerading attack

At first, the attacker collects information that is required to generate a valid authentication message. He also collects several authentication messages by eavesdropping into the insecure channel. Then, he analyzes the authentication messages and tries to figure out the format or structure of them. After figuring out the format/structure, he tries to construct a legal authentication message and then, uses this message when received a login request from a user. Fig. 2.3 shows how server masquerading attack can be launched.

2.1.4 Forgery Attack

It is a kind of attack where the attacker forges into an insecure channel and tries to gather useful information from there. He can easily collect several login and authentication messages by simply eavesdropping into the insecure channel which is used for message interchange. If the attacker somehow can analyze or reuse these messages, then he can forge a valid login request [16]. Fig. 2.4 shows how messages can be collected to conduct forgery attack.

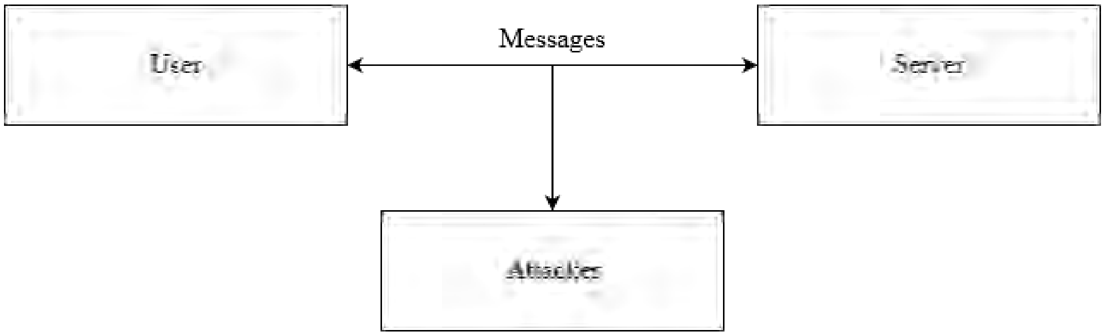

Fig. 2.4. Forgery attack

2.1.5 Replay Attack

It is a kind of attack where the attacker tries to gain access to the server by using old login and authentication messages. An attacker may attempt to pretend to be a valid user to login to the server by sending messages previously transmitted by a legal user [16].

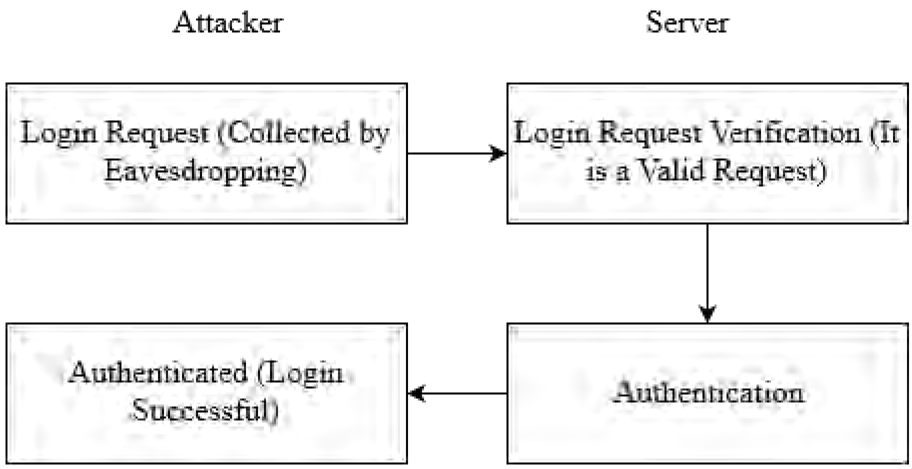

Fig. 2.5. Replay attack

The attacker collects several messages including login and authentication messages by eavesdropping into the insecure channel. Then, he directly uses these messages to send valid login request and tries to gain access to the server. Fig. 2.5 shows how replay attack can be launched.

2.1.6 Insider Attack

An insider attack is a malicious attack conducted on an authentication system by a person with authorized system access. If the users store their valuable information directly to the server, then the insider of the server may directly obtain that valuable information. Thus, the insider of the server as an attacker can impersonate as the legal user to access the user's other accounts in other server if the user uses the same password for the other accounts [15].

2.2 Attack on Smart Card

An attacker can access the information of a smart card by using power analysis attack [17-18]. There are two types of power analysis attack. They are Simple Power Analysis (SPA) attack and Differential Power Analysis (DPA) attack.

The SPA involves visually interpreting power traces, or graphs of electrical activity of a device over time. As the device performs different operations, variations in power consumption occur. For example, different instructions performed by a microprocessor will have differing power consumption profiles. Even if the magnitude of the variations in power consumption is small, standard digital oscilloscopes can easily show the data-induced variations. An attacker can use these data to learn necessary information.

The DPA attack is more advanced form of power analysis which allows an attacker to compute the intermediate values by statistically analyzing data collected from multiple operations. The attack exploits biases varying power consumption of microprocessors or other hardware while performing operations. DPA attacks have signal processing and error correction properties which can extract secrets from measurements which contain too much noise to be analyzed using SPA. Using DPA, an adversary can obtain information by analyzing power consumption measurements from multiple operations performed by a vulnerable smart card or other device.

2.3 Biometric Key Generation

Recently, the use of biometric keys in cryptography is increased dramatically. These keys can be generated from biometric templates. Biometric cryptosystem and cancelable biometrics represent emerging technologies to release biometric keys as well as provide privacy to biometric templates [19]. A few of those technologies are discussed in the following.

2.3.1 Biometric Key Generation Using Fuzzy Extractor

The researchers' in [20] claimed that strong biometric keys can be generated from biometric templates using fuzzy extractor. The fuzzy extractor can generate uniform randomness from the templates close to original. This scheme provides two functions:

$B_i = Gen(P_i, R_i)$……………………………………………………………(2.1)
$R_i = Rep(P_i, B_{ci})$…………………………………………………………...(2.2)

Where, B_i is the biometric, B_{ci} is the biometric very close to B_i, R_i is the generated randomness and P_i is the auxiliary information. Both P_i and encrypted information need not to be kept secret because there is no way to decrypt the encrypted information without B_{ci}.

2.3.2 Efficient Cancelable Biometric Key Generation

The researchers' in [21] proposed an efficient cancelable biometric key generation scheme for cryptographic use. This scheme has three steps that involve feature extraction, generation of secured feature matrix and key generation from feature matrix. This scheme can generate 256 bit keys from fingerprint templates.

2.3.3 Non-invertible Biometric Key Generation

An efficient approach for non-invertible cryptographic key generation from cancelable fingerprint biometrics is proposed in [22]. This scheme consists of feature extraction, generation of transformed points by using one way function, utilization of the points to generate cancelable template and use this template to release unique key.

2.3.4 Biometric Key Generation from Multiple Modalities

The researchers' in [23] proposed a cryptographic key generation scheme from multiple biometric modalities. At first, the features, minutiae points and texture properties are extracted from fingerprint and iris images. Then, the features are fused to generate template and this template is used to generate 256 bit key.

2.4 Hash Function

A cryptographic hash function is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size. This is designed to also be a one-way function, that is, a function which is infeasible to invert [24]. The main properties of a standard hash function are fast computation, infeasible to invert, small change in message changes the hash value extensively and infeasible to find the same hash for different messages. The example of hash function are MD5 [25], SHA-1 [26], SHA-2 family [27], SHA-3 family [28], etc. The MD5 and SHA-1 are no longer recommended due to security reason. Either SHA-2 family or SHA-3 family should be used to generate message digest.

2.5 Advanced Encryption Standard (AES)

AES is a symmetric key algorithm where a single key, named private key, is used for both encryption and decryption purpose [29-30]. It is also known as Rijndael [29]. It is based on a design principle known as substitution-permutation network [31]. It is a block cipher with three different key lengths. They are 128, 192 and 256 bit. The degree of security relies on key length. For instance, top secret information requires either 192 bit or 256 bit key length. There are 10 rounds for 128 bit key, 12 rounds for 192 bit key and 14 rounds for 256 bit key. Each round consists of several processing steps that include substitution, transposition and mixing of the input plain text and finally, to transform it into the final output of cipher text. Despite the different key length, AES operates on 128 bits plain text block and generates 128 bits cipher text. The performance of it is very convincing. It is a high speed algorithm and also requires low memory. Therefore, it can be implemented from 8 bit smart card to high performance computer.

2.6 Summary

In this chapter, we discussed several types of attacks on a user authentication system like user impersonation attack, server masquerading attack, password guessing attack, insider attack, denial of service attack, replay attack, forgery attack, etc. Moreover, we also discussed how an attacker can access the information of a smart card by using power analysis attack. Features like biometric key, AES encryption and decryption, secure hash function are also discussed here.

CHAPTER 3
Related Works

A highly secure system has to deal with high level of security risk. So, it requires a highly secure authentication system. To ensure that type of security, the multifactor user authentication system comes into account. Now-a-days, the biometric and smart card based user authentication schemes along with password have drawn a considerable amount of attention among the researchers [11-16]. Some of them are discussed in the following.

In 2010, an efficient biometrics-based remote user authentication scheme using smart card [11] was proposed by C.T. Li and M.S. Hwang. They claimed that the computation cost of their work was relatively low compared with other related schemes. The proposed scheme is based on smart card, one way hash function, and biometric verification. They claimed that their scheme can resist masquerading attack, replay attack, parallel session attack and also provide the security of information stored within the smart card. Moreover, it enables the user to change their password freely, provides mutual authentication between the remote server and the user, doesn't need to store any password or identity tables, doesn't require any synchronized clock and provides non-repudiation. They also claimed that their scheme is very efficient compared to other schemes. There are three phases in this scheme. They are registration phase, login phase, and authentication phase. During registration phase, the user needs to provide his identity, biometric and password. The registration center processes the information provided by the user and provides a smart card to the user. The user uses this smart card during login phase. The mutual authentication between the user and the remote server occurs in authentication phase. The user can easily change his password without informing the registration center. They provided a security analysis to show how much security their scheme can provide. They also provided a performance comparison and a functionality comparison with other schemes.

In 2011, analysis and improvement on an efficient biometric based remote user authentication scheme using smart cards [14] was proposed by A.K. Das. He claimed that [11] has design flaws in login and authentication phase, password change phase, and verification of biometrics using hash function. He provided an improved scheme that covered up these flaws. The scheme has four phases; they are registration phase, login phase, authentication phase, and password change phase. During registration phase, the user needs to provide his identity, biometric and password. The registration center processes the information provided by the user and provides a smart card to the user. The user uses this smart card during login phase. To prevent replay attack and man-in-the-middle attack, the server stores identity and a value to its database and compares it with the next calculated value. The mutual authentication between the user and the remote server occurs in authentication phase. The user can freely change password by providing both old and new passwords. The authors also provided a security analysis to show that their scheme can cover up the security flaws of [11]. A performance comparison among other schemes and proposed scheme is also shown by means of efficiency.

Y. An, in [15], revealed the security flaws of [14]. He showed that the proposed scheme in [14] is vulnerable to user impersonation attack, server masquerading attack, password guessing attack, and insider attack and also it cannot provide mutual authentication. He showed security analysis and proposed enhancements of an effective biometric-based remote user authentication scheme using smart cards to overcome the security weaknesses of [14] while preserving all their merits. The enhanced scheme is divided into three phases; they are registration phase, login phase, and authentication phase. During registration phase, the user needs to choose a random number and provide his identity, biometric and password. He sends the identity, password and biometric exclusive-ORed by a chosen random number to the registration center. The registration center then processes the information provided by the user and provides a smart card to the user. The user uses this smart card during login phase. The mutual authentication between the user and the remote server occurs in authentication phase. He does not provide any password change phase or password recovery phase or smart card recovery phase. He also provided a security analysis to show that his scheme can overcome the security flaws of [14]. He claimed that his scheme can prevent user impersonation attack, server masquerading attack, password guessing attack, insider attack, and provide mutual authentication. He also provided a security comparison among the related schemes and the enhanced scheme.

In 2013, Li et al. conducted a detail analysis on [15] and revealed some weaknesses e.g., the scheme is vulnerable to denial of service attack, forgery attack, does not provide session key agreement, etc. Li et al. also proposed their robust biometrics based remote user authentication scheme with session key agreement using elliptical curve cryptography [16] to overcome these weaknesses. The biometric verification of this scheme relies on fuzzy extractor and the security of itself relies on one way hash function, Elliptic Curve Discrete Logarithm Problem (ECDLP) and Elliptic Curve Diffie-Hellman Problem (ECDHP). A fuzzy extractor can reliably extract nearly uniform randomness from the biometric input. The extractor is error tolerant in the sense that it can give same output under the help of auxiliary information if the input is reasonably close to the original input. The scheme is divided into four phases; they are registration phase, login phase, authentication and key agreement phase, and password change phase. During registration phase, the user needs to choose a random number and provide his identity and password. He needs to imprint his biometric template at the fuzzy extractor to generate the biometric key. Then, he sends the identity, biometric key and hash of password exclusive-ORed by chosen random number to the registration center. The registration center then processes the information provided by the user and provides a smart card to the user. The user uses this smart card during login phase. The session key generation and the mutual authentication between the user and the remote server occur during authentication phase. The user can freely change password by providing both old and new passwords, identity and biometric key. They also provided a security analysis to show that their scheme can cover up the security flaws of [15]. They claimed that their scheme can provide the security of secret key, session key agreement, proper biometric authentication, quick detection of unauthorized login, proper mutual authentication, prevent forgery attack, stolen smart card attack and replay attack.

3.1 Summary

A highly secure system requires secure authentication system to deal with high level of security risk. To ensure that type of security, multi-factor user authentication system comes into account. Researchers of [11-16] are trying to develop biometric and smart card based user authentication schemes along with password to ensure that type of security. In this chapter, we discussed about few of such schemes. Despite their claims, the schemes suffer from various weaknesses.

CHAPTER 4
Overview of Li et al.'s Scheme

In 2013, Li et al. proposed robust biometrics based remote user authentication scheme with session key agreement using elliptical curve cryptography [16]. In this chapter, we discuss the scheme briefly and present a security analysis to identify weaknesses of the scheme. The notations used in this scheme are shown in Table 4.1.

4.1 Review of the Li et al.'s Scheme

According to the literature [20], a fuzzy extractor can reliably extract nearly uniform randomness R_i from the biometric input B_i; the extraction is error-tolerant in the sense that R_i will be the same under the help of auxiliary information P_i even if the input changes, as long as it remains reasonably close to the original. A fuzzy extractor is a pair of procedure (Gen, Rep) such that:

$$Gen(B_i) = (R_i, P_i) \quad (4.1)$$

$$Rep(B_{ci}, P_i) = R_i \quad (4.2)$$

Where, B_{ci} is the reasonably close to B_i.

Initially, the R chooses an elliptic curve equation $E_p(a, b)$ and a base point P with the order n over $E_p(a, b)$, and publishes the parameters $(E_p(a, b), n, P)$. It also chooses a secret key X_s and distributes it to the server S_i through a secure channel.

There are four phases in this scheme: registration phase, login phase, authentication and key agreement phase, and password change phase.

4.1.1 Registration Phase

During this phase the registration center R and the user C_i have to perform the following steps:

Table 4.1. Notations used in Li et al.'s scheme

Notation	Description
R	Trusted registration center
S_i	Server
C_i	User
A_i	An attacker
ID_i	Identity of the user C_i
PW_i	Password of the user C_i
B_i	Biometric template of the user C_i
P, n	Two large prime numbers
F_p	A finite field
$E_p(a, b)$	An elliptic curve defined on finite field F_p with prime number order n
P	A point on elliptic curve $E_p(a, b)$ with order n
$h(.)$	A secure hash function
X_s	The master secret key
$\parallel$	Message concatenation operation
$\oplus$	Exclusive-OR operation

I. Registration Request

The user C_i provides his ID_i, PW_i, B_i at the fuzzy extractor and a random number K. The C_i sends ID_i, B_i, RPW_i to the registration center R via a secure channel.

$$RPW_i = h(PW_i \parallel K) \quad (4.3)$$

II. Data Processing

The R computes e_i, f_i, r_i and R_i using (4.5), (4.4), (4.6) and (4.1) respectively.

$$f_i = h(ID_i \parallel R_i) \quad (4.4)$$

$$e_i = h(ID_i \parallel X_s) \oplus h(f_i \parallel RPW_i) \quad (4.5)$$

$$r_i = h(ID_i \parallel RPW_i) \quad (4.6)$$

III. Card Preparation and Delivery

The R stores $(e_i, f_i, r_i, P_i, h(.))$ on the C_i's smart card and sends it to the C_i via a secure channel.

IV. Finalization

The C_i enters K into the smart card. The registration phase is illustrated in Fig. 4.1.

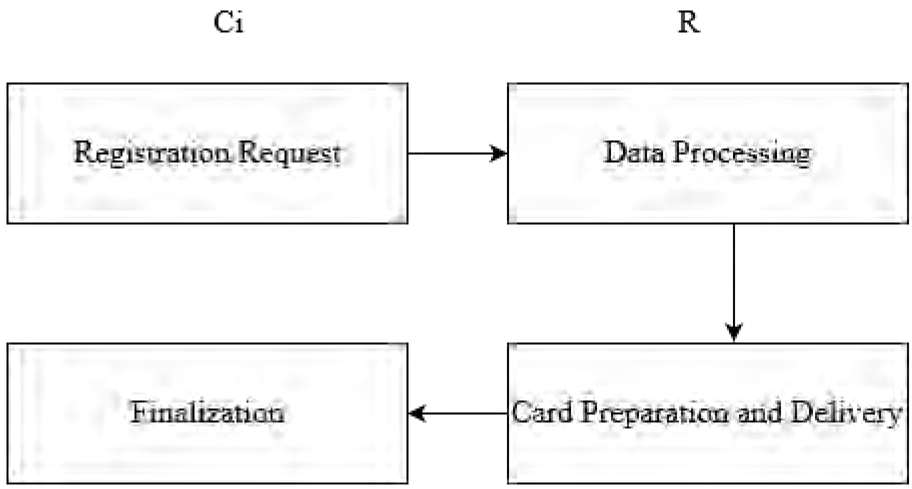

Fig. 4.1. Registration phase of Li et al.'s scheme which involves the user C_i and the registration center R

4.1.2 Login Phase

During this phase the user C_i performs the following steps:

I. Biometric Verification

The C_i inserts the smart card to card reader and also provides ID_i, PW_i, B_i to a specific device with fuzzy extractor and generates R_i using (4.2). Then, smart card computes f_{ci} by placing provided ID_i and calculated R_i at (4.4). If $f_{ci} = f_i$, then the user C_i passes the biometric verification and continues the following steps. Otherwise, the session is terminated.

II. Password Verification

The smart card computes RPW_i and r_{ci} using (4.3) and by placing provided ID_i and calculated RPW_i at (4.6) respectively. It checks whether $r_{ci} = r_i$ or not. If they are equal, then ID_i and PW_i are verified and smart card performs the next step. Otherwise, the session is terminated.

III. Login Request

The smart card computes M_1, M_2 and M_3 using (4.7), (4.8) and (4.9) respectively.

$$M_1 = e_i \oplus h(f_i \parallel RPW_i) \qquad (4.7)$$
$$M_2 = aP \text{ where } a \epsilon Z_n^* \qquad (4.8)$$
$$M_3 = h(M_1 \parallel M_2) \qquad (4.9)$$

The C_i sends login request $\{ID_i, M_2, M_3\}$ to the server S_i.

4.1.3 Authentication and Session Key Agreement Phase

During this phase the user C_i and the server S_i performs the following steps:

I. User ID Validation

The S_i checks the format of ID_i.

II. Login Request Verification

If ID_i is valid, then the S_i computes M_4 and M_{c3} using following equations:

$$M_4 = h(ID_i \parallel X_s) \qquad (4.10)$$
$$M_{c3} = h(M_4 \parallel M_2) \qquad (4.11)$$

It checks whether $M_3 = M_{c3}$ or not. If they are equal, the S_i accepts the login request message and the validity of the user C_i is authenticated by the server S_i. Otherwise, the session is terminated.

III. Mutual Authentication Request

The server S_i computes M_5 and M_6 using (4.12) and (4.13) respectively.

$M_5 = bP$ where $b \epsilon Z_n^*$ ……………………………………………………………..(4.12)

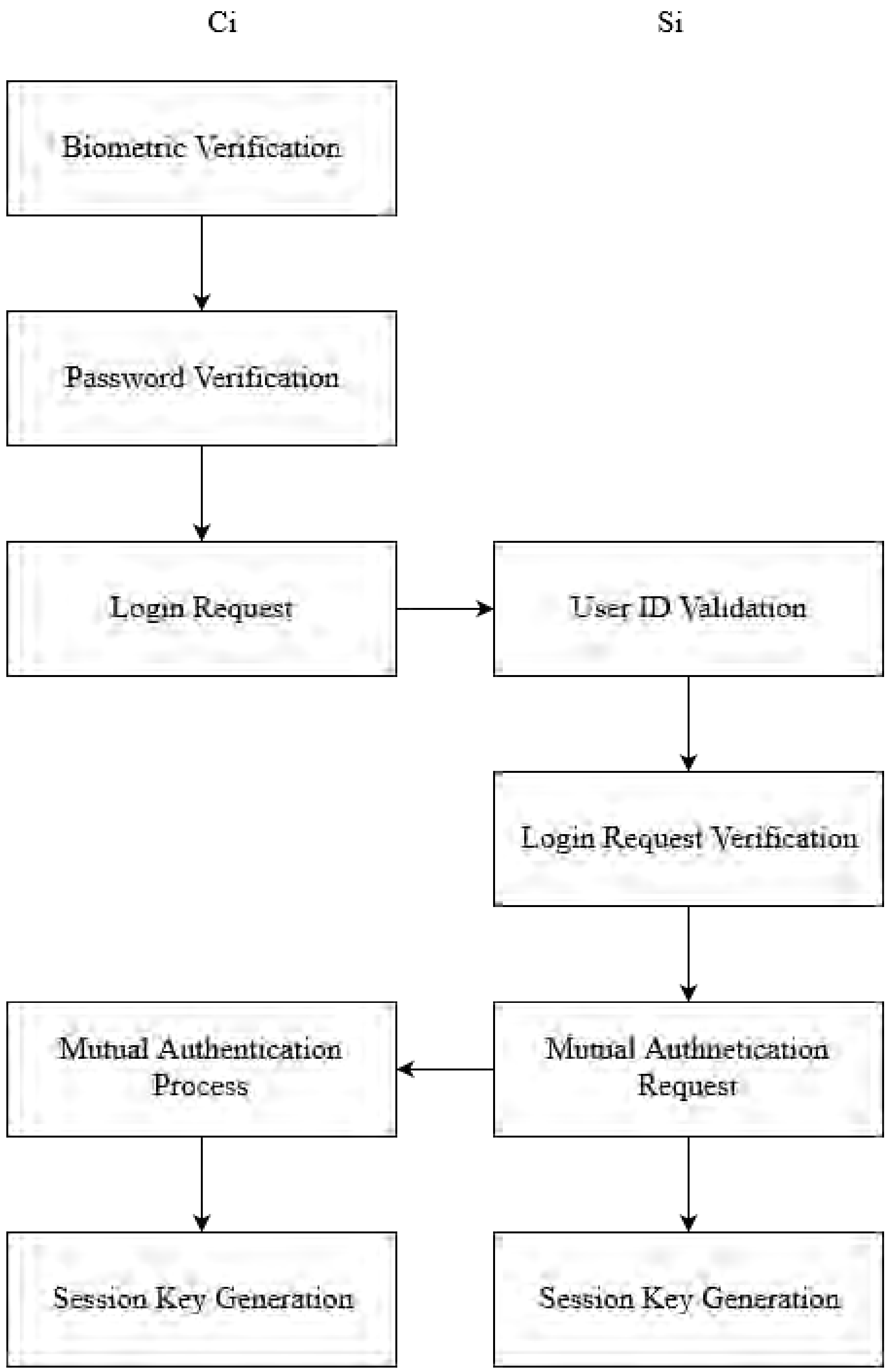

Fig. 4.2. Login phase and authentication & session key agreement phase of Li et al.'s scheme which involves the user C_i and the server S_i

$$M_6 = h(M_4 \parallel M_2 \parallel M_5) \quad (4.13)$$

It sends the mutual authentication message $\{M_5, M_6\}$ to the user C_i.

IV. Mutual Authentication Process

After receiving the reply, the user C_i checks whether $M_6 = M_{c6}$ or not. The M_{c6} is calculated as follow:

$$M_{c6} = h(M_1 \parallel M_2 \parallel M_5) \quad (4.14)$$

If they are equal, then the server S_i is authenticated by the user C_i and mutual authentication is completed.

V. Session Key Generation

The user C_i and the server S_i compute a shared key using (4.15).

$$SK = h(aM_5) = h(bM_2) = h(abP) \quad (4.15)$$

It is used for future confidential communication. The login phase and authentication & session key agreement phase is illustrated in Fig. 4.2.

4.1.4 Password Change Phase

During this phase the user C_i performs the following steps:

I. Biometric Verification

The C_i inserts the smart card to card reader and also provides ID_i, PW_i, B_i to a specific device with fuzzy extractor and generates R_i (4.2). Then, smart card computes f_{ci} by placing provided ID_i and calculated R_i at (4.4) and compares it with f_i which is stored in the smart card. If $f_{ci} = f_i$, then the user C_i passes the biometric verification and continues the following steps.

II. Password Verification

The smart card computes RPW_i and r_{ci} using (4.3) and by placing provided ID_i and calculated RPW_i at (4.6) respectively and checks whether $r_{ci} = r_i$ or not. If they are equal, then ID_i and PW_i are verified and smart card performs the next step. The user inputs his new password PW_{ni}.

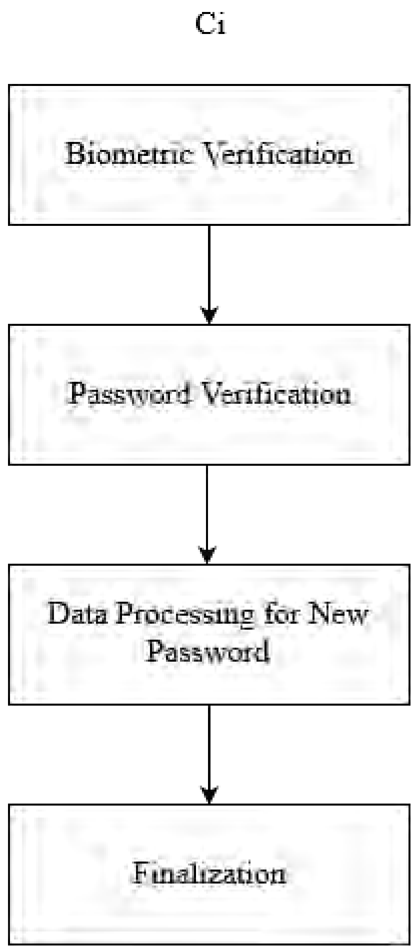

Fig. 4.3. Password change phase of Li et al.'s scheme which involves the user C_i

III. Data Procession for New Password

The smart card computes RPW_{ni} and r_{ni} using (4.3) and (4.6) respectively and by replacing PW_i by PW_{ni} and RPW_i by RPW_{ni}. The e_{ni} is calculated as follow:

$$e_{ni} = e_i \oplus h(f_i \parallel RPW_i) \oplus h(f_i \parallel RPW_{ni}) \qquad (4.16)$$

IV. Finalization

The smart card replaces e_i and r_i by e_{ni} and r_{ni} respectively to complete the phase. Fig. 4.3 shows the password change phase of Li et al.'s scheme.

4.2 Security Analysis of the Li et al.'s Scheme

The security weaknesses of Li et al.'s scheme are discussed in the following. We assume that the attacker A_i can control the insecure channel.

4.2.1 Password Guessing Attack Using Stolen Smart Card

If the A_i can manage to steal the smart card, then he can manage to extract the information from the card by examining the power consumption signal as discussed in section 2.2. The A_i also can manage the ID_i by capturing one of the login request messages. When the attacker manages to achieve the information r_i, ID_i, K, $h(.)$, he can conduct the password guessing attack as follow:

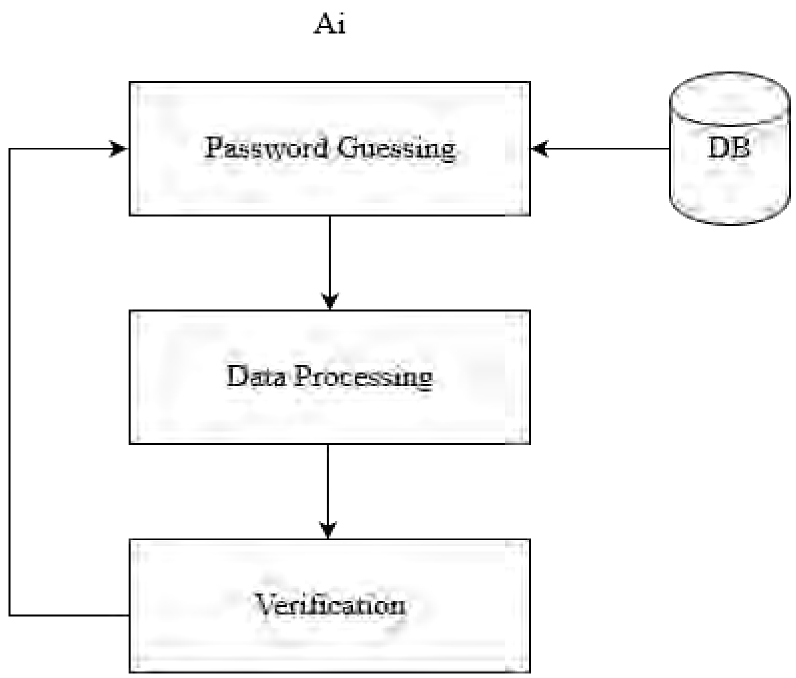

Fig. 4.4. Password guessing attack on Li et al.'s scheme which involves the attacker A_i

I. Password Guessing by the Attacker

Consider that the A_i selects a password PW_{ai} from a massive database of passwords collected by him.

II. Data Processing by the Attacker

Then, the A_i computes RPW_{ai} and r_{ai} using (4.3) and (4.6) respectively and by replacing PW_i by PW_{ai} and RPW_i by RPW_{ai}.

III. Verification by the Attacker

The A_i checks whether $r_i = r_{ai}$ or not. If they are equal, the selected password is correct. Otherwise, repeat the steps.

4.2.2 User impersonation Attack

User impersonation attack can be launched after password guessing attack using stolen smart card. From previous discussion, we know that the attacker A_i has $(e_i, f_i, r_i, P_i, h(.), K)$ from smart card as well as the password PW_{ai} from password guessing attack and ID_i by capturing one of the login request messages or simply using shoulder surfing technique. Additionally, the parameters $(E_p(a,b), n, P)$ which are published by the registration center R, are stored in the smart card or in a public domain. So, the A_i can manage to gather these parameters. Now, the A_i can perform the following steps and try to login to the remote server S_i.

I. Login Request by the Attacker

The A_i computes RPW_{ai} using (4.3) and by replacing PW_i by PW_{ai}. It also calculates M_1, M_2 and M_3 using (4.7), (4.8) and (4.9) respectively and by replacing RPW_i by RPW_{ai}. It sends $\{ID_i, M_2, M_3\}$ to the server S_i.

II. User ID Validation by the Server

The S_i receives the message sent by the A_i and verifies ID_i.

III. Login Request Verification by the Server

If ID_i is valid, then the S_i computes M_4 and M_{c3} from (4.10) and (4.11) respectively. It checks whether $M_3 = M_{c3}$ or not. Because they are equal, the server S_i authenticates the A_i as valid user.

IV. Mutual Authentication Request by the Server

The server S_i computes M_5 and M_6 using (4.12) and (4.13) respectively and sends the mutual authentication message $\{M_5, M_6\}$ to the attacker A_i.

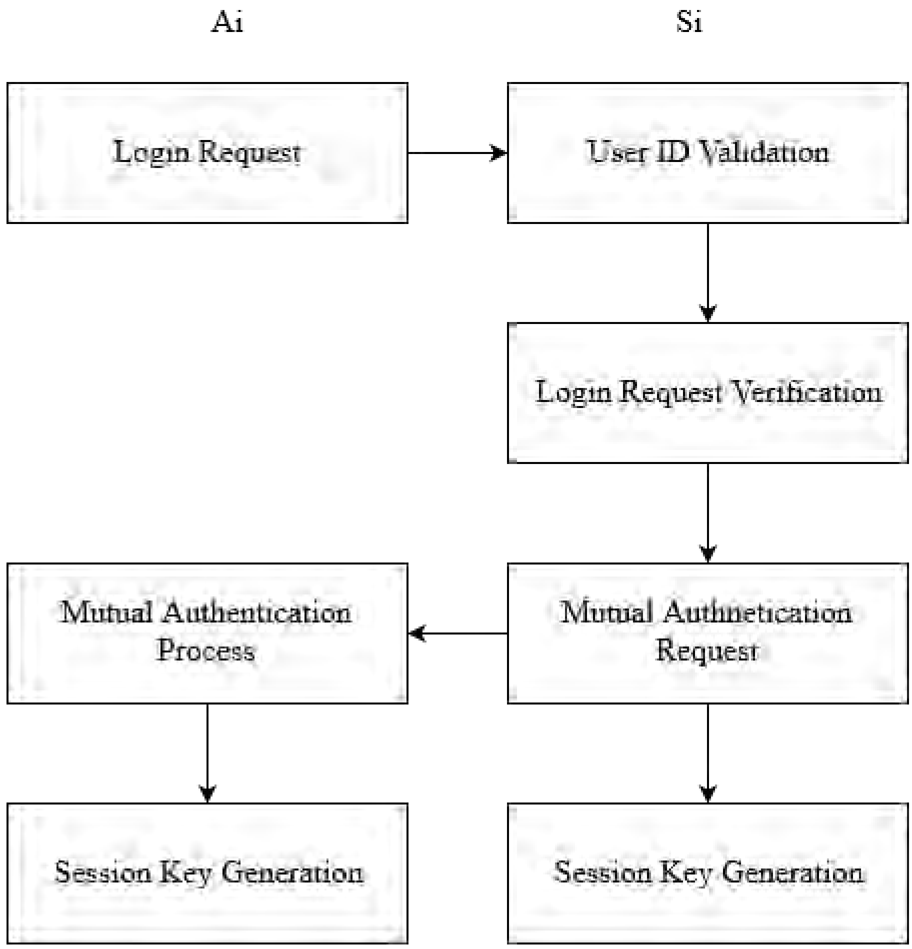

Fig. 4.5. User impersonation attack on Li et al.'s scheme which involves the attacker A_i and the server S_i

V. Mutual Authentication Process by the Attacker

After receiving the reply, the attacker A_i calculates M_{c6} using (4.14) and checks whether $M_6 = M_{c6}$ or not. If they are equal, then the server S_i is authenticated by the attacker A_i and mutual authentication is completed.

VI. Session Key Generation by the Attacker and the Server

The attacker A_i and the server S_i compute shared key SK using (4.15) and use it for future confidential communication. Fig. 4.5 shows the user impersonation on Li et al.'s scheme.

4.2.3 Security of the Secret Key

The secret key X_s remains stored in the server. Generally, server stores this type of information in a database or in a file. Since X_s will be unique for every user, the server S_i has to maintain a mapping of ID_i and X_s. According to the discussion in [6], the information stored in the server could be compromised. Therefore, this scheme is unable to provide the security of secret key.

4.2.4 Server Masquerading Attack

If the attacker A_i can manage to steal the secret key X_s as discussed in previous section, then it can lunch attack as follow:

I. Login Request by the User

The C_i computes RPW_i, M_1, M_2 and M_3 using (4.3), (4.7), (4.8) and (4.9), and sends $\{ID_i, M_2, M_3\}$ to the A_i (because the A_i is masquerading as server).

II. Login Request Verification by the Attacker

The A_i receives the message sent by the C_i and computes M_4 and M_{c3} using (4.10) and (4.11) respectively and checks whether $M_3 = M_{c3}$ or not. Because they are equal, the A_i authenticates the C_i as valid user.

III. Mutual Authentication Request by the Attacker

The A_i computes M_5 and M_6 using (4.12) and (4.13) respectively and sends the mutual authentication message $\{M_5, M_6\}$ to the user C_i.

IV. Mutual Authentication Process by the User

After receiving the reply, the user C_i calculates M_{c6} using (4.14) and checks whether $M_6 = M_{c6}$ or not. If they are equal, then the A_i (because the C_i believes the A_i as server) is authenticated by the user C_i and mutual authentication is completed.

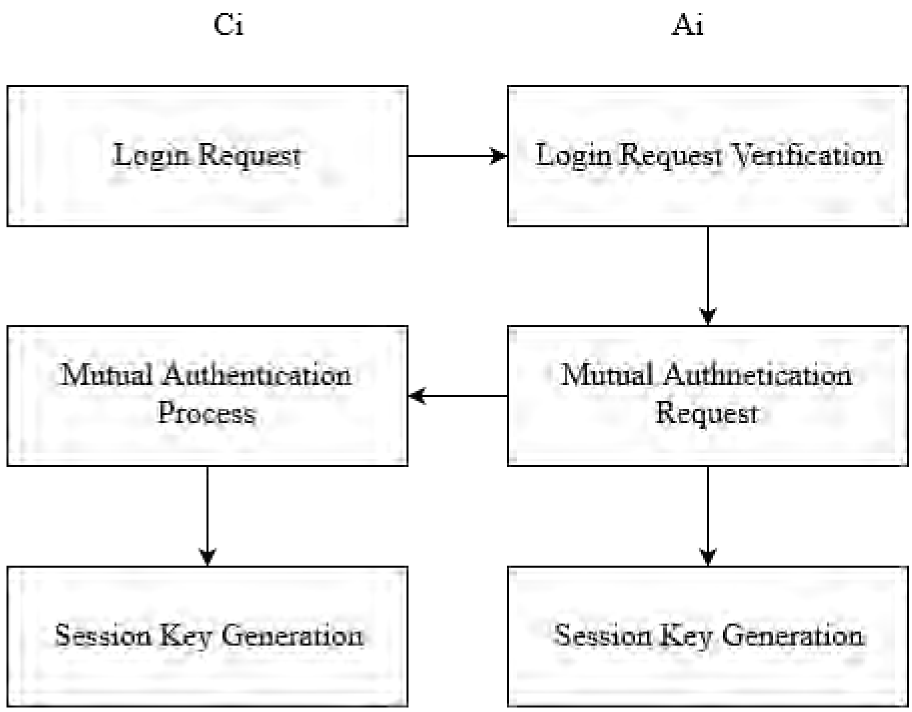

Fig. 4.6. Server masquerading attack on Li et al.'s scheme which involves the user C_i and the attacker A_i

V. Session Key Generation by the User and the Attacker

The user C_i and the A_i compute a shared key SK using (4.15) and use it for future confidential communication. The server masquerading attack on Li et al.'s scheme is illustrated in Fig. 4.6.

4.2.5 No Password or Smart Card Recovery Phase

There is no password or smart card recovery phase in this scheme. According to the discussion of [5], the user C_i can forget his password. If he forgets his password there is no way he ever can get logged in. Moreover, if the attacker A_i somehow can manage to steal the smart card of the user C_i, then he will not be able to recover the smart card.

4.2.6 Fails to Provide Mutual Authentication

According to [15], if authentication scheme is insecure against user impersonation attack and server masquerading attack, the authentication schemes cannot provide mutual authentication between the user and the remote server. Therefore, this scheme fails to provide mutual authentication according to the discussion in section 4.2.2 and 4.2.4.

4.3 Summary

In this chapter, we discussed about Li et al.'s scheme and presented security analysis of this scheme. The scheme has four phases: registration phase, login phase, authentication and key agreement phase, and password change phase. It uses fuzzy extractor to generate biometric key and elliptical curve cryptography to exchange session key between the user and the server. Though they tried to provide a robust scheme, their scheme is unable to provide security against password guessing attack, user impersonation attack, server key stealing, server masquerading attack, etc. Additionally, it is unable to provide mutual authentication and password, or smart card recovery system.

CHAPTER 5

Proposed Scheme for Remote User Authentication

Our proposed scheme consists of six phases that includes server registration phase, user registration phase, login and authentication phase, password change phase, password recovery phase and smart card recovery phase. The notations used in our proposed scheme are given in Table 5.1. We have few assumptions under which our proposed scheme worked properly as discussed below.

5.1 Assumptions

- User ID ID_i is unique but not a secret
- Server ID SID_i is unique but not a secret
- Password PW_i is secret but it may not be unique
- Biometric key B_i is unique and very hard to be copied, shared and distributed
- AES keys are secret and the attacker A_i cannot steal them
- Smart card can be stolen and information stored in smart card can be revealed
- The attacker A_i cannot manage to steal smart card, password and biometric key at the same time
- The attacker A_i cannot take over the secure channel
- The attacker A_i has control over the insecure channel

Our proposed scheme works under these assumptions.

5.2 Proposed Scheme

All six phases of our proposed scheme are discussed as follow:

5.2.1 Server Registration Phase

During the server registration phase, the server S_i and the registration center R need to perform the following steps:

Table 5.1. Notations used in our proposed scheme

Notation	Description
C_i	User
S_i	Server
R	Trusted registration center
A_i	An attacker
ID_i	Identity of the user C_i
SID_i	Identity of the server S_i
PW_i	Password of the user C_i
PW_{ni}	New password chosen by the user C_i
B_i	Biometric key of the user C_i
R_{cont}	Recovery contact of the user C_i
$h(.)$	A secure hash function
K_s	The master secret key for server
X_s	The master secret key for user
$W, R_{n1} - R_{n8}$	Secret random strings
RK_{aes}	The AES key of the trusted registration center R
SK_{aes}	The AES key of the server S_i
$E_{aes}(.)$	Encryption function for AES
$D_{aes}(.)$	Decryption function for AES
K_{ses}	Session key
$Stat$	Status message
$\|\|$	Message concatenation operation
$\oplus$	Exclusive-OR operation

I. Registration Request by the Server

The server sets $Stat = Register$ and sends $\{SID_i, Stat\}$ to the registration center R.

II. Secret Generation and Reply by the Registration Center

The R receives $\{SID_i, Stat\}$ from the S_i. If $Stat = Register$, then it checks in to its database. If the server S_i is already registered, then it discards the process. Otherwise, it chooses a secret random string R_{n1} and generates K_s using (5.1).

$$K_s = h(SID_i \parallel R_{n1}) \qquad (5.1)$$

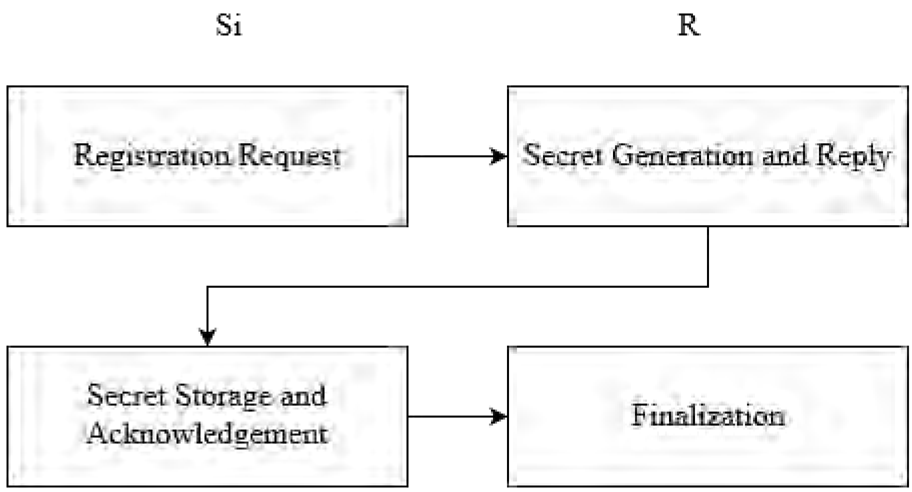

Fig. 5.1. Server registration phase of proposed scheme which involves the server S_i and the registration center R

The R sets $Stat = Accept$ and sends $\{SID_i, K_s, Stat\}$ to the S_i through a secure channel.

III. Secret Storage and Acknowledgement by the Server

The S_i receives $\{SID_i, K_s, Stat\}$ from the R and checks SID_i. If SID_i matches with its own and $Stat = Accept$, then it calculates EK_s using (5.2).

$$EK_s = Eaes(K_s, SK_{aes}) \qquad (5.2)$$

It stores EK_s in its database. It sets $Stat = Ack$ and sends $\{SID_i, Stat\}$ to the R.

IV. Finalization by the Registration Center

After receiving the acknowledgement $\{SID_i, Stat\}$ from the S_i, the R calculates HK_s as follow:

$$HK_s = E_{aes}(K_s, RK_{aes}) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (5.3)$$

It stores $\{SID_i, HK_s\}$ in its database.

5.2.2 User Registration Phase

During user registration phase, the user C_i, the registration center R and the server S_i need to perform the following steps:

I. Registration Request by the User

The user C_i needs to choose his user identification ID_i, password PW_i, recovery contact R_{cont}, collect server identification SID_i which is published publicly and imprint his biometrics in a specific device which can generate biometric key B_i form biometrics. He also calculates BP_i as follow:

$$BP_i = h(PW_i \parallel B_i) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots .(5.4)$$

Then, he sets $Stat = Register$ and sends $\{ID_i, BP_i, SID_i, R_{cont}, Stat\}$ to the registration center R through a secure channel.

II. Registration Request by the Registration Center

The registration center R receives the message $\{ID_i, BP_i, SID_i, R_{cont}, Stat\}$ from the C_i. If $Stat = Register$ and SID_i is already registered, then it generates a secret random string W and calculates TX_s using (5.5).

$$TX_s = W \parallel BP_i \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots .(5.5)$$

The R sends $\{ID_i, SID_i, TX_s, Stat\}$ to the server S_i through a secure channel.

III. Secret Generation, Storage and Reply by the Server

The server S_i receives $\{ID_i, SID_i, TX_s, Stat\}$ from the R. It verifies SID_i. If the verification passes, then it proceeds; otherwise, it discards the request. If the verification passes and $Stat = Register$, then it calculates K_s, X_s and SX_i using (5.6), (5.7) and (5.8) respectively.

$K_s = D_{aes}(EK_s, SK_{aes})$ ………………………………………………………… (5.6)
$X_s = h(K_s \parallel TX_s)$ ………………………………………………………… (5.7)
$SX_i = E_{aes}(X_s, SK_{aes})$ ………………………………………………………… (5.8)

Then, it stores $\{ID_i, SX_i\}$ into its database, sets $Stat = Complete$ and sends $\{ID_i, SID_i, Stat\}$ to the registration center R through a secure channel.

IV. Card Preparation and Delivery by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the S_i. It verifies ID_i and SID_i. If verification passes, then it checks the value of $Stat$. If $Stat = Complete$, then it confirms that the registration process in the server has completed. Then, it calculates K_s and TC_s using (5.9) and (5.10) respectively.

$K_s = D_{aes}(HK_s, RK_{aes})$ ………………………………………………………… (5.9)
$TC_s = K_s \parallel W$ ………………………………………………………… (5.10)

It stores $\{ID_i, SID_i, BP_i, TC_s\}$ to the smart card and distributes it to the C_i through a secure channel.

V. Card Receive and Acknowledgement by the User

The C_i receives smart card from the R. Then, he puts the smart card into a card reader. The C_i checks ID_i and SID_i, and if they are correct, then he provides PW_i and imprints his biometrics to the specific device to generate biometric key B_i. Then, the C_i calculates BP_{ci} by putting PW_i and B_i at (5.4) and compares whether $BP_i = BP_{ci}$ or not. If the verification passes, then he accepts the card, sets $Stat = Accept$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel.

$$QX_i = E_{aes}(TC_s, B_i) \quad (5.11)$$

The user C_i calculates QX_i using (5.11) and replaces TC_s in the smart card. He also removes BP_i from the card. If the verification fails, he rejects the card, sets $Stat = Reject$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R and discards the process.

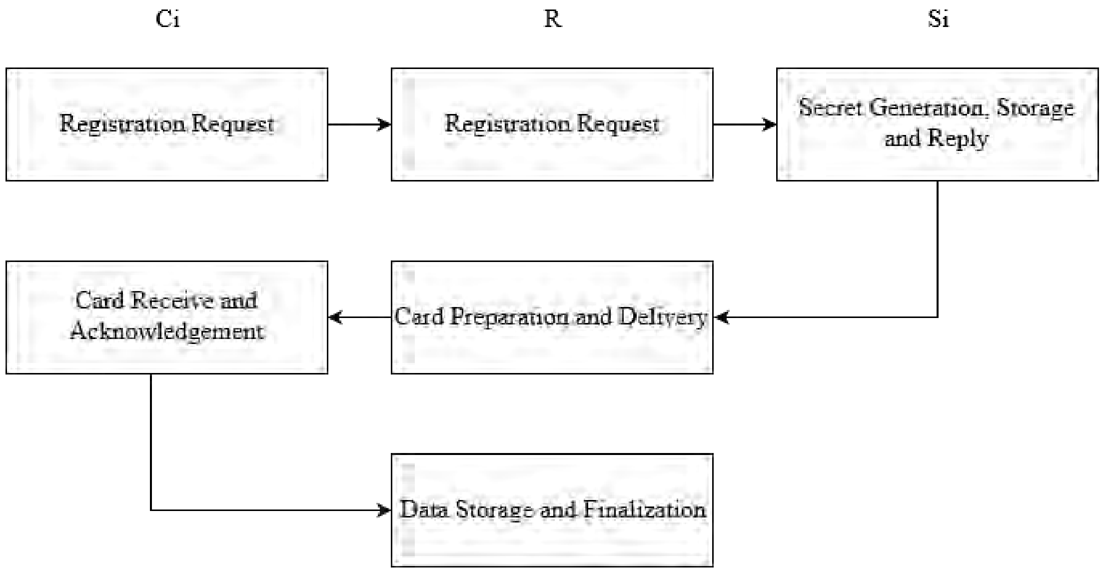

Fig. 5.2. User registration phase of proposed scheme which involves the user C_i, the registration center R and the server S_i

VI. Data Storage and Finalization by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the C_i. It verifies ID_i and SID_i. If verification passes, then it checks $Stat$. If $Stat = Accept$, then it confirms that the card has reached to the designated user. Then, it calculates R_{cov}, EX_i and UX_i as follow:

$$R_{cov} = E_{aes}(R_{cont}, RK_{aes}) \quad (5.12)$$

$$EX_i = E_{aes}(TX_s, RK_{aes}) \quad (5.13)$$

$$UX_i = E_{aes}(TC_s, RK_{aes}) \quad (5.14)$$

It stores $\{ID_i, SID_i, UX_i, EX_i, R_{cov}\}$ to its database. If $Stat = Reject$ or if the verification fails, then it discards the process and sets $Stat = Deregister$ and sends the S_i a message $\{ID_i, SID_i, Stat\}$ to deregister the user through a secure channel.

When S_i receives such message, then it deletes the corresponding data from its database. The user registration phase is illustrated in Fig. 5.2.

5.2.3 Login and Authentication Phase

During this phase, the user C_i and the server S_i needs to perform the following steps:

I. Login Request by the User

The user C_i inserts his smart card into the card reader. He also provides his ID_i, PW_i and imprints his biometrics to a specific device to generate biometric key B_i. Then, the smart card verifies ID_i. If the verification fails, then he terminates the session. Then, he calculates BP_i, TC_s and X_s using (5.3), (5.15) and (5.16) respectively.

$$TC_s = D_{aes}(QX_i, B_i) \ldots \quad (5.15)$$
$$X_s = h(TC_s \parallel BP_i) \ldots \quad (5.16)$$
$$M_1 = h(X_s \parallel R_{n2}) \ldots \quad (5.17)$$
$$M_2 = h(ID_i \parallel X_s) \oplus R_{n2} \ldots \quad (5.18)$$

It generates a secret random string R_{n2} and calculates M_1 and M_2 using (5.17) and (5.18) respectively. Then, he sets $Stat = Login$ and sends $\{ID_i, SID_i, M_1, M_2, Stat\}$ to the server S_i.

II. Verification and Mutual Authentication Request by the Server

The server S_i receives the login message $\{ID_i, SID_i, M_1, M_2, Stat\}$ from the user C_i. It verifies ID_i of the message with stored ID_i and SID_i with its server id. If the verification fails, then it terminates the session. If the verification passes and $Stat = Login$, then it proceeds. It calculates X_s, R_{n2} and M_3 using (5.19), (5.20) and (5.17) respectively.

$$X_s = D_{aes}(SX_i, SK_{aes}) \ldots \quad (5.19)$$
$$R_{n2} = M_2 \oplus h(ID_i \parallel X_s) \ldots \quad (5.20)$$
$$M_5 = h(ID_i \parallel X_s \parallel R_{n2}) \oplus R_{n3} \ldots \quad (5.21)$$

Then, it compares whether $M_1 = M_3$ or not. If they are not equal, it terminates the session. Otherwise, the user is authenticated. It generates a secret random string R_{n3}. It calculates M_4 by replacing R_{n2} with R_{n3} at (5.17) and M_5 using (5.21). Then, it sets $Stat = Auth$ and sends $\{ID_i, SID_i, M_4, M_5, Stat\}$ to the user C_i.

III. Mutual Authentication and Acknowledgement by the User

The user C_i receives the message $\{ID_i, SID_i, M_4, M_5, Stat\}$ from the S_i. He verifies ID_i and SID_i of the message. If the verification fails, then he terminates the session. If the verification passes and $Stat = Auth$, then he proceeds. He calculates R_{n3} using (5.22) and M_6 by replacing R_{n2} with R_{n3} at (5.17).

$$R_{n3} = M_5 \oplus h(ID_i \parallel X_s \parallel R_{n2}) \quad \ldots\ldots (5.22)$$

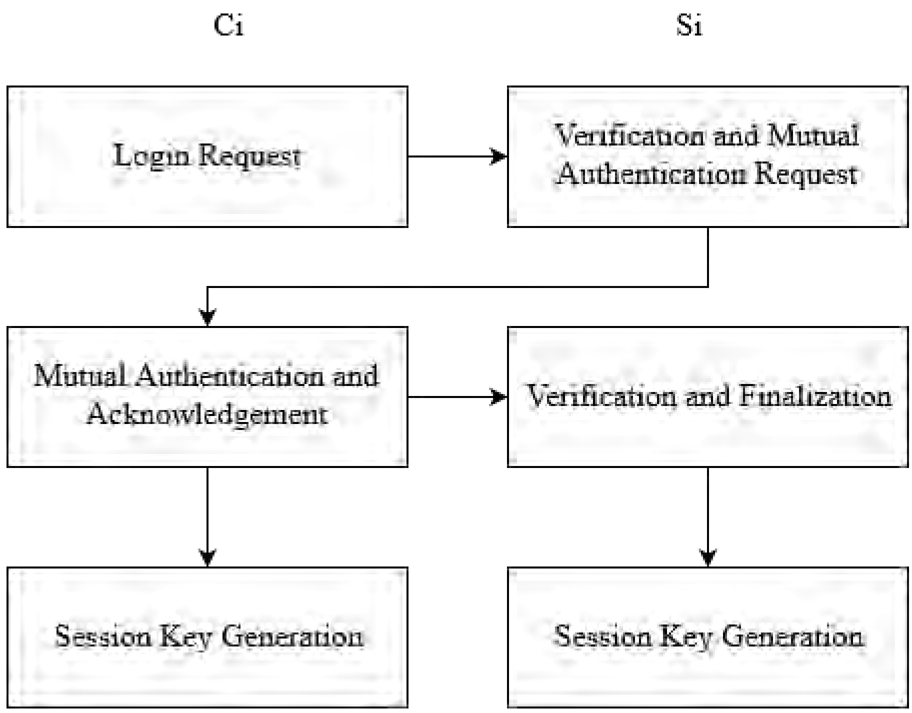

Fig. 5.3. Login and authentication phase of proposed scheme which involves the user C_i and the server S_i

$$M_7 = h(X_s \parallel R_{n2} \parallel R_{n3}) \quad \ldots\ldots (5.23)$$

He compares whether $M_4 = M_6$ or not. If they are not equal, then he terminates the session. Otherwise, the server is authenticated. He calculates M_7 using (5.23) and sets $Stat = Auth$ and sends $\{ID_i, SID_i, M_7, Stat\}$ to the server S_i. If the session is terminated, then he sends login request again.

IV. Verification and Finalization by the Server

The server S_i receives $\{ID_i, SID_i, M_7, Stat\}$ from the C_i. Then, it checks ID_i and SID_i. If ID_i is desired user id, SID_i is desired server id and $Stat = Auth$, then it calculates M_8 by putting X_s, R_{n2} and R_{n3} at (5.23). It compares whether $M_7 = M_8$ or not. If they are not equal, then it discards the message and terminates the session. Otherwise, the authentication is completed.

V. Session Key Generation by the User and the Server

The S_i and the C_i both calculate the session key for further secret communication. The session key is calculated as follow:

$$K_{ses} = h(R_{n2} \parallel R_{n3}) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .. (5.24)$$

The login and authentication phase is illustrated in Fig. 5.3.

5.2.4 Password Change Phase

During password change phase the user C_i, the R and the S_i has to perform the following steps:

I. Password Change Request by the User

The user C_i inserts his smart card into the card reader. He also provides his ID_i, PW_i and imprints his biometrics to a specific device to generate biometric key B_i. Then, the smart card verifies ID_i. If the verification fails, then it discards the process. Otherwise, it calculates BP_i, TC_s, X_s and TCX_s using (5.4), (5.15), (5.16) and (5.25) respectively.

$$TCX_s = TC_s \oplus X_s \ldots \ldots (5.25)$$

He also provides a new password PW_{ni}. Then the smart card calculates BP_{ni} by replacing PW_i with PW_{ni} at (5.4), sets $Stat = Passchange$ and sends $\{ID_i, TCX_s, BP_{ni}, SID_i, Stat\}$ to the R through a secure channel.

II. Secret Change Request by the Registration Center

The R receives the message $\{ID_i, TCX_s, BP_{ni}, SID_i, Stat\}$ from the C_i. It verifies ID_i and SID_i with its database. If verification fails, then it discards the request. If the verification passes and $Stat = Passchange$, then it calculates TC_s, K_s, TX_s and X_s using (5.26), (5.9), (5.27) and (5.28) respectively and X_{cs} by putting calculated K_s and TX_s at (5.7).

$$TC_s = D_{aes}(UX_i, RK_{aes}) \ldots \ldots (5.26)$$
$$TX_s = D_{aes}(EX_i, RK_{aes}) \ldots \ldots (5.27)$$
$$X_s = TCX_s \oplus TC_s \ldots \ldots (5.28)$$

It compares whether $X_{cs} = X_s$ or not. If they are not equal, then the request is discarded. If they match, then it sets $Stat = Passchange$, chooses a secret random string R_{n4}, calculates TX_{ns} by replacing W and BP_i with R_{n4} and BP_{ni} respectively at (5.5) and sends $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ to the server S_i through a secure channel. If the process is discarded, then the R sets $Stat = Fail$ and sends failure message $\{ID_i, Stat\}$ to the user C_i. If the C_i receives the failure message, then he sends the password change request again.

III. Secret Change and Reply by the Server

The server S_i receives $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ from the R. It verifies ID_i and SID_i. If the verification passes, then it proceeds. Otherwise, it discards the request. It calculates K_s and X_s using (5.6) and (5.19) respectively and X_{cs} by putting calculated K_s and TX_s at (5.7) and compares whether $X_s = X_{cs}$ or not. If they are not equal, then it discards the request. If they match, then it calculates X_{ns} and SX_{ni} by replacing

TX_s with TX_{ns} and X_s with X_{ns} at (5.7) and (5.8) respectively. It replaces SX_i with SX_{ni} in its database. Then, it sets $Stat = Complete$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel. If the process is discarded, then it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R. If the R receives the failure message, then it sends the secret change request again.

IV. Smart Card Update Request by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the S_i. It verifies ID_i and SID_i. If verification passes and $Stat = Complete$, then it assumes that the update process in the server has completed. Otherwise, if the verification fails, then it discards the process.

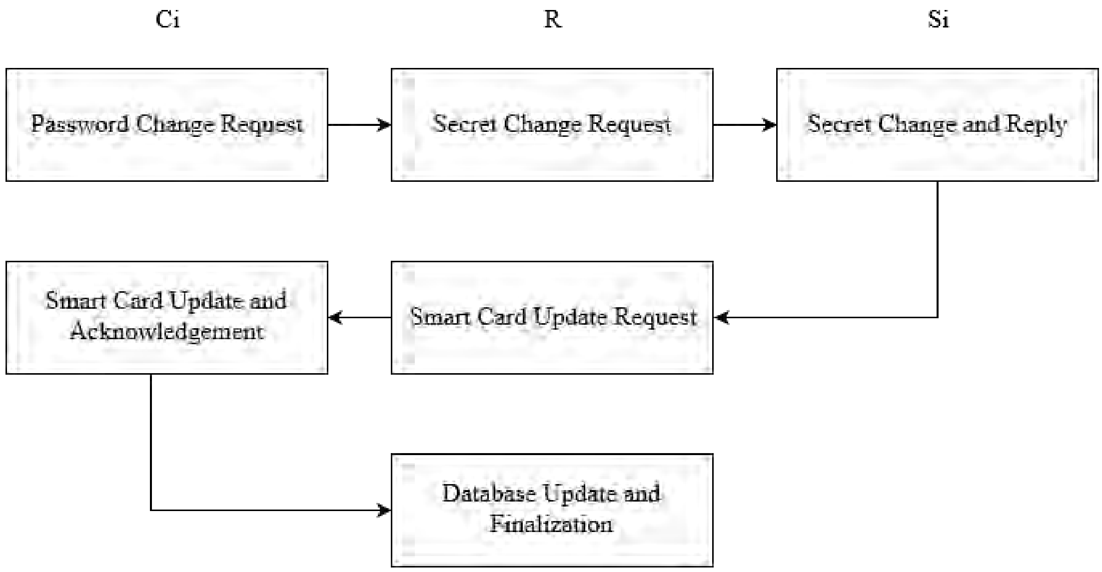

Fig. 5.4. Password change phase of proposed scheme which involves the user C_i, the registration center R and the server S_i

When it has the confirmation of the completion of the server update, then it calculates TC_{ns} by replacing W with R_{n4} at (5.10) and TC_s using (5.26), sets $Stat = Complete$ and sends $\{ID_i, SID_i, TC_{ns}, TC_s, Stat\}$ to the user C_i through a secure channel. If the process is discarded, it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i. If the S_i receives this failure message, then it simply revert the database change. The R sends the secret change request to the S_i again.

V. Smart Card Update and Acknowledgement by the User

The user C_i receives the message $\{ID_i, SID_i, TC_{ns}, TC_s, Stat\}$ from the R. Then, he checks ID_i and SID_i. If they are correct and $Stat = Complete$, then he calculates TC_{cs} using (5.15) and compares whether $TC_s = TC_{cs}$ or not. If the verification passes, then he sets $Stat = Complete$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel. Then, he calculates QX_{ni} by replacing TC_s with TC_{ns} at (5.11) and replaces QX_i with QX_{ni} in the smart card. If any of the verification fails, then he rejects the reply. If the reply is rejected, he discards the process, sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R and retries password change again. If the R receives the failure message, then it sends the failure message to the S_i to revert the database change and the S_i does accordingly.

VI. Database Update and Finalization by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If verification passes and $Stat = Complete$, then it confirms that the card has updated and it calculates UX_{ni} and EX_{ni} by replacing TC_s with TC_{ns} and TX_s with TX_{ns} at (5.14) and (5.13) respectively. It also replaces UX_i with UX_{ni} and EX_i with EX_{ni} in its own database. Otherwise, if the verification fails, it discards the process. The R sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i and the C_i to revert their changes and they act accordingly. The password change phase is illustrated in Fig. 5.4.

5.2.5 Password Recovery Phase

If the user C_i forgets his password, then he, the registration center R and the server S_i have to perform the following steps:

I. Password Recovery Request by the User

The user C_i needs to provide his user identification ID_i and collect server identification SID_i which is published publicly. Then, he sets $Stat = Recovery$ and sends the message $\{ID_i, SID_i, Stat\}$ to the R through a secure channel.

II. User Verification Request by the Registration Center

The R receives the message $\{ID_i, SID_i, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If verification passes and $Stat = Recovery$, then it proceeds. Otherwise, if the verification process fails, it terminates the process. It generates a secret random string R_{n5} and calculates R_{cont} as follow:

$$R_{cont} = D_{aes}(R_{cov}, RK_{aes}) \quad (5.29)$$

It sets $Stat = Verify$ and sends a message $\{ID_i, SID_i, R_{n5}, Stat\}$ to the recovery contact (R_{cont}) of the C_i through a secure channel.

III. Verification Reply by the User

The user C_i receives the message $\{ID_i, SID_i, R_{n5}, Stat\}$ from the R. Then he checks ID_i and SID_i. If they are correct and $Stat = Verify$, then he proceeds. He sets $Stat = Verify$, chooses a new password PW_{ni}, calculates BP_{ni} by replacing PW_i with PW_{ni} at (5.4) and sends $\{ID_i, SID_i, R_{n5}, BP_{ni}, Stat\}$ to the R through a secure channel. Otherwise, if the verification fails, he simply discards the message and sends the recovery request again.

IV. Finalization of Verification and Secret Change Request by the Registration Center

The R receives the message $\{ID_i, SID_i, R_{n5}, BP_{ni}, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If the verification passes, and $Stat = Verify$ then it verifies R_{n5}. If it holds, then it confirms that the user C_i is valid. Otherwise, if any of the verifications fails, it discards the message. If the user C_i is valid, then it chooses a secret random string R_{n6}, calculates TX_s using (5.27) and TX_{ns} by replacing W and BP_i with R_{n6} and BP_{ni} respectively at (5.5). Then, it sets $Stat = Recovery$ and sends $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ to the server S_i through a secure channel.

V. Secret Change and Reply by the Server

The server S_i receives $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ from the R. It verifies ID_i and SID_i. If the verification passes, then it proceeds; otherwise, it discards the request. It calculates K_s and X_s using (5.6) and (5.19) respectively and X_{cs} by putting calculated K_s and TX_s at (5.7) and compares whether $X_s = X_{cs}$ or not. If they are not equal, then it discards the request. If they match, then it calculates X_{ns} and SX_{ni} by replacing TX_s with TX_{ns} and X_s with X_{ns} at (5.7) and (5.8) respectively and replaces SX_i with SX_{ni} in its database. Then, it sets $Stat = Done$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel.

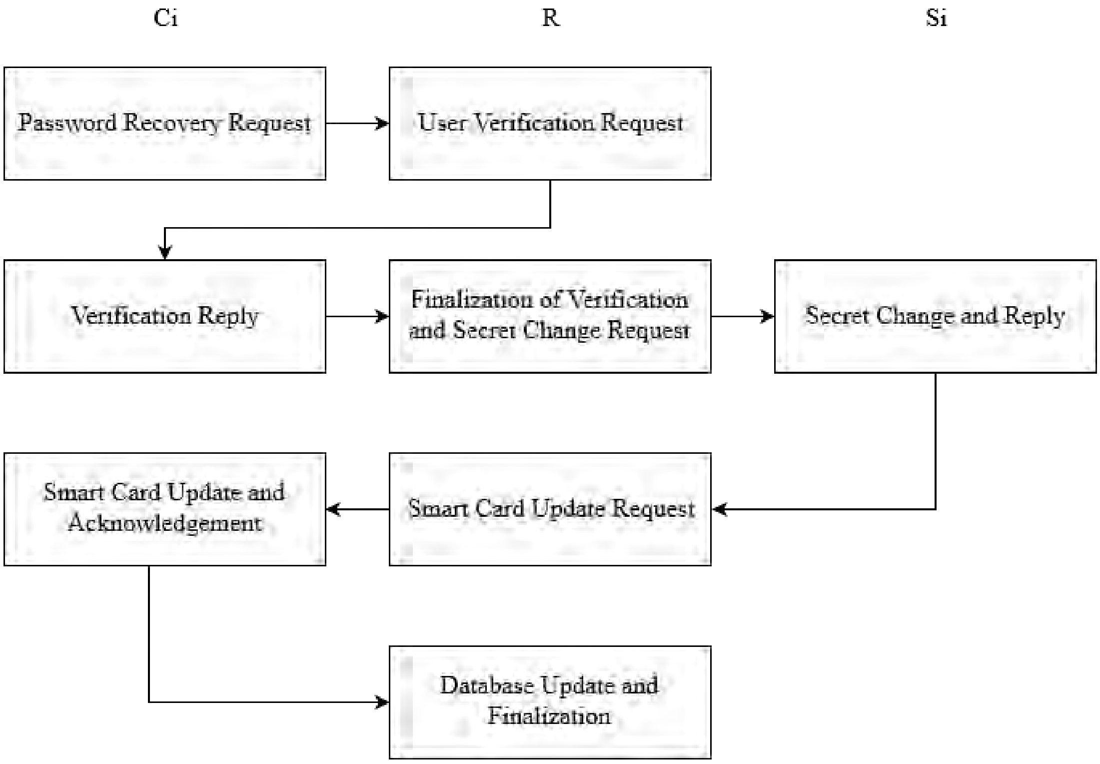

Fig. 5.5. Password recovery phase of proposed scheme which involves the user C_i, the registration center R and the server S_i

If the process is discarded, then it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R. If the R receives the failure message, then it sends the secret change request again.

VI. Smart Card Update Request by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the S_i. It verifies ID_i and SID_i. If verification passes and $Stat = Done$, then it assumes that the update process in the server has completed. Otherwise, if the verification fails, it discards the process. When it has the confirmation of the completion of the server update, then it calculates TC_{ns} by replacing W with R_{n6} at (5.10) and TC_s using (5.26). It sets $Stat = Done$ and sends $\{ID_i, SID_i, TC_{ns}, TC_s, Stat\}$ to the user C_i through a secure channel. If the process is discarded, it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i. If the S_i receives this failure message, then it simply reverts the database change. The R sends the secret change request to the S_i again.

VII. Smart Card Update and Acknowledgement by the User

The user C_i receives the message $\{ID_i, SID_i, TC_{ns}, TC_s, Stat\}$ from the R. Then, he checks ID_i and SID_i. If they are correct and $Stat = Done$, then he calculates TC_{cs} using (5.15) and compares whether $TC_s = TC_{cs}$ or not. If the verification passes, then he sets $Stat = Complete$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel. Then, he calculates QX_{ni} by replacing TC_s with TC_{ns} at (5.11) and replaces QX_i with QX_{ni} in the smart card. If any of the verification fails, then he rejects the reply. If the reply is rejected, he discards the process, sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R and retries password recovery again. If the R receives the failure message, then it sends the failure message to the S_i to revert the database change and the S_i does accordingly.

VIII. Database Update and Finalization by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If verification passes and $Stat = Complete$, then it confirms that the card has been updated and it calculates UX_{ni} and EX_{ni} by replacing TC_s with TC_{ns} and TX_s with TX_{ns} at (5.14) and (5.13) respectively. It also replaces UX_i with UX_{ni} and EX_i with EX_{ni} in its own database. Otherwise, if the verification fails, it discards the process. The R sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i and the C_i to revert their changes and they acts accordingly. The password recovery phase is illustrated in Fig. 5.5.

5.2.6 Smart Card Recovery Phase

If the user C_i loses his smart card then he, the registration center R and the server S_i have to perform the following steps:

I. Smart Card Recovery Request by the User

The user C_i needs to provide his user identification ID_i and collect server identification SID_i which is published publicly. Then, he sets $Stat = RecoveryS$ and sends the message $\{ID_i, SID_i, Stat\}$ to the R through a secure channel.

II. User Verification Request by the Registration Center

The R receives the message $\{ID_i, SID_i, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If verification passes and $Stat = RecoveryS$, then it proceeds. Otherwise, if the verification process fails, it terminates the process. It generates a secret random string R_{n7} and calculates R_{cont} using (5.29). It sets $Stat = VerifyS$ and sends a message $\{ID_i, SID_i, R_{n7}, Stat\}$ to the recovery contact (R_{cont}) of the C_i through a secure channel.

III. Verification Reply by the User

The user C_i receives the message $\{ID_i, SID_i, R_{n7}, Stat\}$ from the R. Then, he checks ID_i and SID_i. If they are correct and $Stat = VerifyS$, then he proceeds. He sets $Stat = VerifyS$, chooses a new password PW_{ni}, calculates BP_{ni} by replacing PW_i with PW_{ni} at (5.4) and sends $\{ID_i, SID_i, R_{n7}, BP_{ni}, Stat\}$ to the R through a secure channel. Otherwise, if the verification fails, then he simply discards the message and sends the recovery request again.

IV. Finalization of Verification and Secret Change Request by the Registration Center

The R receives the message $\{ID_i, SID_i, R_{n7}, BP_{ni}, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If the verification passes and $Stat = VerifyS$, then it verifies R_{n7}. If it holds, then it confirms that the user C_i is valid. Otherwise, if any of the verifications fails, it discards the message. If the user C_i is valid, then it chooses a secret random string R_{n8}, calculates TX_s using (5.27) and TX_{ns} by replacing W and BP_i with R_{n8}

and BP_{ni} respectively at (5.5). Then, it sets $Stat = RecoveryS$ and sends $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ to the server S_i through a secure channel.

V. Secret Change and Reply by the Server

The server S_i receives $\{ID_i, SID_i, TX_s, TX_{ns}, Stat\}$ from the R. It verifies ID_i and SID_i. If the verification passes and $Stat = RecoveryS$, then it proceeds. Otherwise, it discards the request. It calculates K_s and X_s using (5.6) and (5.19) respectively and X_{cs} by putting calculated K_s and TX_s at (5.7) and compares whether $X_s = X_{cs}$ or not. If they are not equal, then it discards the request. If they match, then it calculates X_{ns} and SX_{ni} by replacing TX_s with TX_{ns} and X_s with X_{ns} at (5.7) and (5.8) respectively and replaces SX_i with SX_{ni} in its database. Then, it sets $Stat = DoneS$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel. If the process is discarded, then it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the R. If the R receives the failure message, then it sends the secret change request again.

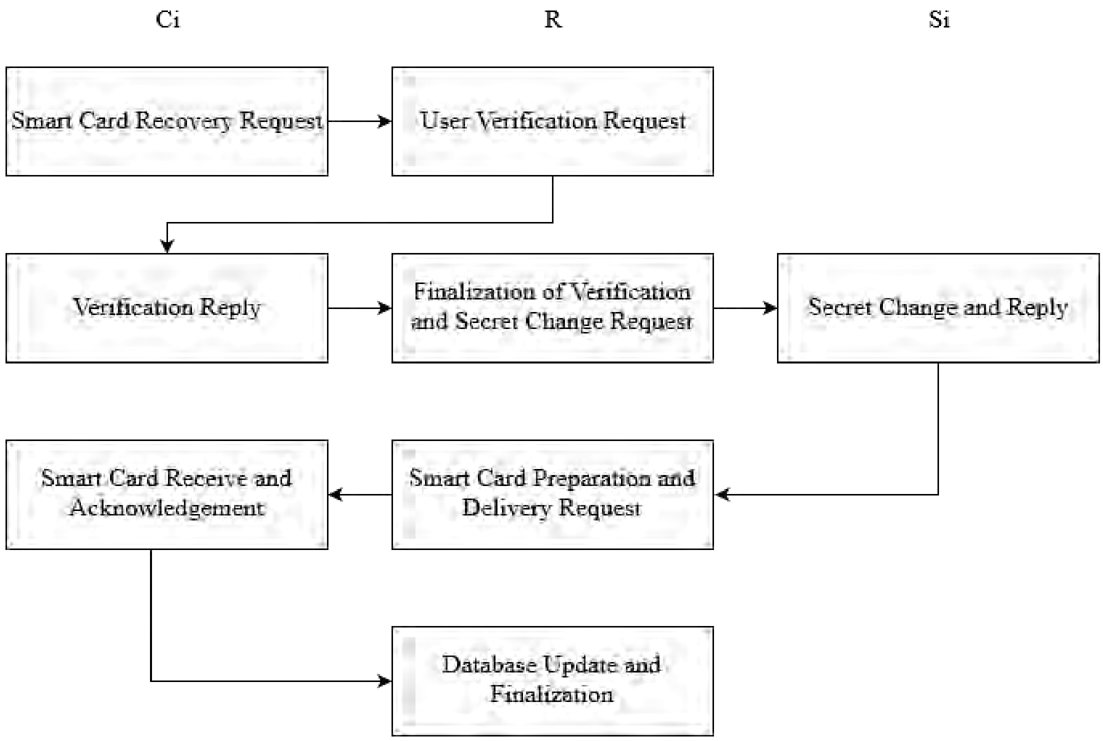

Fig. 5.6. Smart card recovery phase of proposed scheme which involves the user C_i, the registration center R and the server S_i

VI. Smart Card Preparation and Delivery by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the S_i. It verifies ID_i and SID_i. If verification passes and $Stat = DoneS$, then it assumes that the update process in the server has completed. Otherwise, if the verification fails, it discards the process. When it has the confirmation of the completion of the server update, then it calculates TC_{ns} by replacing W with R_{n8} at (5.10). It stores $\{ID_i, SID_i, BP_{ni}, TC_{ns}\}$ into a smart card and distributes it to the user C_i through a secure channel. If the process is discarded, it sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i. If the S_i receives this failure message, then it simply reverts the database change. The R sends the secret change request to the S_i again.

VII. Smart Card Receive and Acknowledgement by the User

The C_i receives smart card from the R. Then, the C_i puts the smart card into a card reader. Then, he checks ID_i and SID_i. If they are correct, then C_i provides PW_{ni} and imprints his biometrics to the specific device to generate biometric key B_i. Then, he calculates BP_{ci} by replacing PW_i with PW_{ni} at (5.4) and compares whether $BP_{ni} = BP_{ci}$ or not. If the verification passes, then he accepts the card, sets $Stat = AcceptS$ and sends $\{ID_i, SID_i, Stat\}$ to the R through a secure channel. Then, he calculates QX_{ni} by replacing TC_s with TC_{ns} at (5.11) and replaces TC_{ns} with QX_{ni} in the smart card. He also removes BP_{ni} from the card. If the verification fails, then he discards the process, sets $Stat = Fail$, sends a failure message $\{ID_i, SID_i, Stat\}$ to the R trough a secure channel and sends recovery request again. If the R receives the failure message, then it sends the failure message to the S_i to revert the database change and the S_i does accordingly.

VIII. Database Update and Finalization by the Registration Center

The R receives $\{ID_i, SID_i, Stat\}$ from the user C_i. It verifies ID_i and SID_i. If verification passes and $Stat = AcceptS$, then it confirms that the card has reached to its designated user and it calculates UX_{ni} and EX_{ni} by replacing TC_s with TC_{ns} and TX_s with TX_{ns} at (5.14) and (5.13) respectively. It also replaces UX_i with UX_{ni} and EX_i with EX_{ni} in its own database. Otherwise, if the verification fails, it discards the process. The R sets $Stat = Fail$ and sends a failure message $\{ID_i, SID_i, Stat\}$ to the S_i and the C_i to revert their changes and they act accordingly. The smart card recovery phase is shown in Fig. 5.6.

5.3 Summary

In this chapter, we presented our proposed scheme. It has six phases: server registration phase, user registration phase, login and authentication phase, password change phase, password recovery phase and smart card recovery phase. We use an efficient cancelable biometric key generation scheme [21] for generating 256 bit biometric key. Additionally, we introduce few features that make it different from other existing schemes. One of those features is to stop password being compromised by not storing it. The strength of AES is used to prevent data being compromised from database. In our scheme, security of master key is very important; it cannot be compromised by any means. Otherwise, the scheme will fail to provide security against potential attacks. We do not send master key directly through communication line to avoid any security risk. Moreover, our scheme has password recovery phase and smart card recovery phase which are missing in other existing scheme. We also involve registration center in our password change phase to enhance the security and reliability of our scheme.

CHAPTER 6
Security Analysis

In this chapter, we present security analysis of our proposed scheme. The smart card used in our scheme can hold information like $\{ID_i, SID_i, QX_i\}$. If the attacker A_i somehow manages to steal the smart card, then he can acquire all these information. Also, the A_i can manage to get the messages like $\{ID_i, SID_i, M_1, M_2, Stat\}$, $\{ID_i, SID_i, M_4, M_5, Stat\}$, $\{ID_i, SID_i, M_7, Stat\}$ and $\{ID_i, SID_i, Stat\}$ from both login and authentication phase by eavesdropping the insecure channel. M_1, M_2, M_5 and M_7 are calculated using (5.17), (5.18), (5.21) and (5.23) respectively. The M_4 is calculated by replacing R_{n2} with R_{n3} at (5.17). After acquiring all these information, the attacker A_i can conduct potential security attacks like password guessing attack, secret key stealing, user impersonation attack, server masquerading attack, replay attack, denial of service attack, forgery attack, etc. We will show how our scheme can resist these security attacks and prevent the attacker A_i to cause any potential harm.

6.1 Password Guessing Attack

If the A_i can manage to steal the smart card, then he can manage to extract the information from the card as discussed in section 2.2. When the attacker manages to achieve the information of the smart card, he can try to conduct the password guessing attack as follow:

I. Smart Card Information Collected by the Attacker

The A_i can collect the information $\{ID_i, SID_i, QX_i\}$ from smart card.

II. The Point of Failure

The QX_i is an encrypted information and contains TC_s within it. Moreover, we can clearly see from (5.10) that TC_s holds no information about password. So, there is no way the A_i can conduct password guessing attack by trial and error basis.

6.2 Secret Key Stealing

According to our protocol, master secret key X_s is not stored in the smart card and the registration center. It is also protected in the server. Also, the attacker may try to use login messages like M_1, M_2, M_4, M_5 and M_7 to predict X_s. The discussion regarding this argument is given below:

I. Smart Card

We already know the A_i can collect the information $\{ID_i, SID_i, QX_i\}$ from smart card. To generate X_s, the attacker needs to decrypt QX_i to get TC_s, and also needs biometric key B_i and password PW_i. As discussed in section 5.1, it is impossible for an attacker to gather all these information at the same time.

II. Registration Center

At the trusted registration center, we store EX_i, UX_i and HK_s. From (5.13), (5.14) and (5.3), we can see all these data are protected by means of AES encryption. Unless the attacker can manage RK_{aes}, it is impossible for him to generate master secret key X_s.

III. Server

From (5.8), we can see that X_s is protected by means of AES encryption in SX_i. The attacker can collect X_s if and only if he can manage to collect SK_{aes}. But, in our scheme, SK_{aes} is securely stored by the server. So, it is nearly impossible for the attacker to get X_s from the server.

IV. Login Messages

The attacker A_i can gather login messages like M_1, M_2, M_4, M_5 and M_7 by forging into the insecure channel used during Login and Authentication phase. Here, R_{n2} and R_{n3} are two secret random strings generated during Login and Authentication phase and they change every time during message generation. Therefore, there is no way to collect these strings. This is why, it is nearly impossible for an attacker to guess X_s from these messages.

6.3 User Impersonation Attack

To conduct user impersonation attack, the attacker A_i needs to send login request that contains message $\{ID_i, SID_i, M_1, M_2, Stat\}$. More precisely, he needs to generate M_1 and M_2. From previous discussion at section 6.2.4, we already know that the attacker cannot manage X_s and R_{n2}. So, it is not possible for him to generate M_1 and M_2. Let us consider that the attacker A_i guesses X_{sa} and R_{n2a} and try to login as follow:

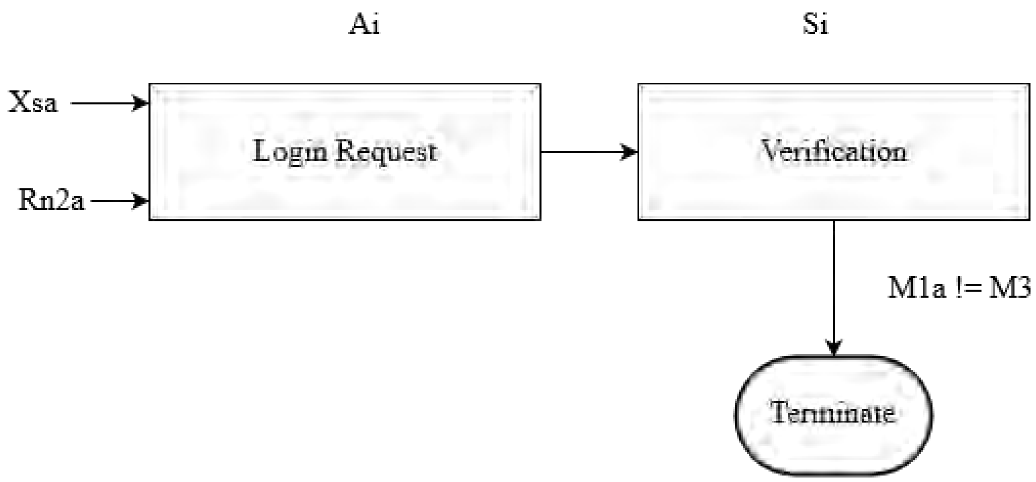

Fig. 6.1. User impersonation attack on proposed scheme which involves the attacker A_i and the server S_i

I. Login Request by the Attacker

The A_i calculates M_{1a} and M_{2a} using (6.1) and (6.2) respectively.

$$M_{1a} = h(X_{sa} \parallel R_{n2a}) \quad (6.1)$$

$$M_{2a} = h(ID_i \parallel X_{sa}) \oplus R_{n2a} \quad (6.2)$$

It sends a login request $\{ID_i, SID_i, M_{1a}, M_{2a}, Stat\}$ to the server S_i.

II. Verification by the Server and the Point of Failure

The server S_i receives $\{ID_i, SID_i, M_{1a}, M_{2a}, Stat\}$ from the A_i. It calculates X_s, R_{n2a} and M_{3a} using (5.19), (6.3) and (6.4) respectively.

$R_{n2a} = M_{2a} \oplus h(ID_i \parallel X_s)$ ……………………………………………………… (6.3)

$M_{3c} = h(X_s \parallel R_{n2a})$ ……………………………………………………… (6.4)

Then, it compares whether $M_{1a} = M_{3c}$ or not. Because X_{sa} and X_s are not equal, therefore M_{1a} and M_{3c} cannot be equal. So, the verification fails and the session is terminated. The user impersonation attack on proposed scheme is shown in Fig. 6.1.

6.4 Server Masquerading Attack

To conduct server masquerading attack, the attacker A_i needs to send mutual authentication request that contains message $\{ID_i, SID_i, M_4, M_5, Stat\}$. More precisely, he needs to generate M_4 and M_5. From previous discussion at section 6.2.4, we already know that the attacker cannot manage X_s, R_{n2} and R_{n3}. So it is not possible for him to generate M_4 and M_5. Let us consider that the attacker A_i guesses X_{sa}, R_{n2a} and R_{n3a} and try as follow:

I. Login Request by the User

The user C_i inserts his smart card into the card reader. He also provides his ID_i, PW_i and imprints his biometrics to a specific device to generate biometric key B_i. Then, the smart card verifies ID_i. If the verification fails, then he terminates the session. Then, he calculates BP_i, TC_s and X_s using (5.3), (5.15) and (5.16) respectively. It generates a secret random string R_{n2}, and calculates M_1 and M_2 using (5.17) and (5.18) respectively. Then, he sets $Stat = Login$ and sends $\{ID_i, SID_i, M_1, M_2, Stat\}$ to the server (here the A_i).

II. Verification and Mutual Authentication Request by the Attacker

The server A_i receives the login message $\{ID_i, SID_i, M_1, M_2, Stat\}$ from the user C_i. It verifies ID_i of the message with stored ID_i and SID_i with its server id. If the verification fails, it terminates the session. If the verification passes and $Stat = Login$, then it proceeds. It calculates M_{4a} and M_{5a} as follow:

$M_{4a} = h(X_{sa} \parallel R_{n3a})$ ……………………………………………………… (6.5)

$$M_{5a} = h(ID_i \parallel X_{sa} \parallel R_{n2a}) \oplus R_{n3a} \quad \text{(6.6)}$$

Then, it sets $Stat = Auth$ and sends $\{ID_i, SID_i, M_{4a}, M_{5a}, Stat\}$ to the user C_i.

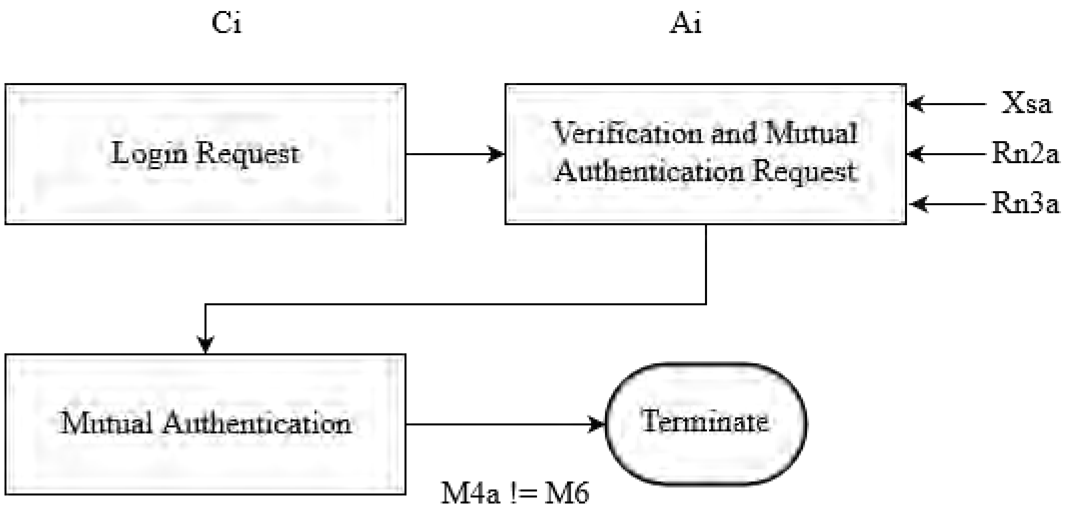

Fig. 6.2. Server masquerading attack on proposed scheme which involves the user C_i and the attacker A_i

III. Mutual Authentication by the User and the Point of Failure

The user C_i receives the message $\{ID_i, SID_i, M_{4a}, M_{5a}, Stat\}$ from the A_i. He verifies ID_i and SID_i of the message. If the verification fails, then he terminates the session. If the verification passes and $Stat = Auth$, then he proceeds.

$$R_{n3a} = M_{5a} \oplus h(ID_i \parallel X_s \parallel R_{n2}) \quad \text{(6.7)}$$

$$M_{6c} = h(X_s \parallel R_{n3a}) \quad \text{(6.8)}$$

He calculates R_{n3a} and M_{6c} using (6.7) and (6.8) respectively and compares whether $M_{4a} = M_{6c}$ or not. Because X_{sa} and X_s are not equal and R_{n2a} and R_{n2} are not equal, therefore M_{4a} and M_{6c} cannot be equal. So, the verification is failed and the session is terminated. The server masquerading attack on proposed scheme is illustrated in Fig. 6.2.

6.5 Replay Attack

The attacker A_i may use captured M_1 and M_2 for sending a login request or captured M_7 for sending acknowledgement. The A_i can try to conduct the replay attack by using captured M_1, M_2 and M_7 as follow:

I. Login Request by the Attacker

The A_i sends a login request $\{ID_i, SID_i, M_1, M_2, Stat\}$ using captured M_1 and M_2 to the server S_i. Here, $Stat = Login$.

II. Verification and Mutual Authentication Request by the Server

The server S_i receives the login message $\{ID_i, SID_i, M_1, M_2, Stat\}$ from the user A_i. It verifies ID_i of the message with stored ID_i and SID_i with its server id. If the verification fails, then it terminates the session. If the verification passes and $Stat = Login$, then it proceeds. It calculates X_s, R_{n2} and M_3 using (5.19), (5.20) and (5.21) respectively. Then, it compares whether $M_1 = M_3$ or not. If they are not equal, it terminates the session. Otherwise, the user is authenticated. It generates a secret random string R_{n3}. It calculates M_4 and M_5 using (5.22) and (5.23) respectively. Then, it sets $Stat = Auth$ and sends $\{ID_i, SID_i, M_4, M_5, Stat\}$ to the user (A_i).

III. Mutual Authentication and Acknowledgement by the Attacker

The user (A_i) receives the message $\{ID_i, SID_i, M_4, M_5, Stat\}$ from the S_i. He verifies ID_i and SID_i of the message. If the verification fails, then he terminates the session. If the verification passes and $Stat = Auth$, then he proceeds. He cannot calculate R_{n3} and M_6 because X_s and R_{n2} are unknown to him. Let us consider that he uses captured M_7, sets $Stat = Auth$ and sends $\{ID_i, SID_i, M_7, Stat\}$ to the server S_i.

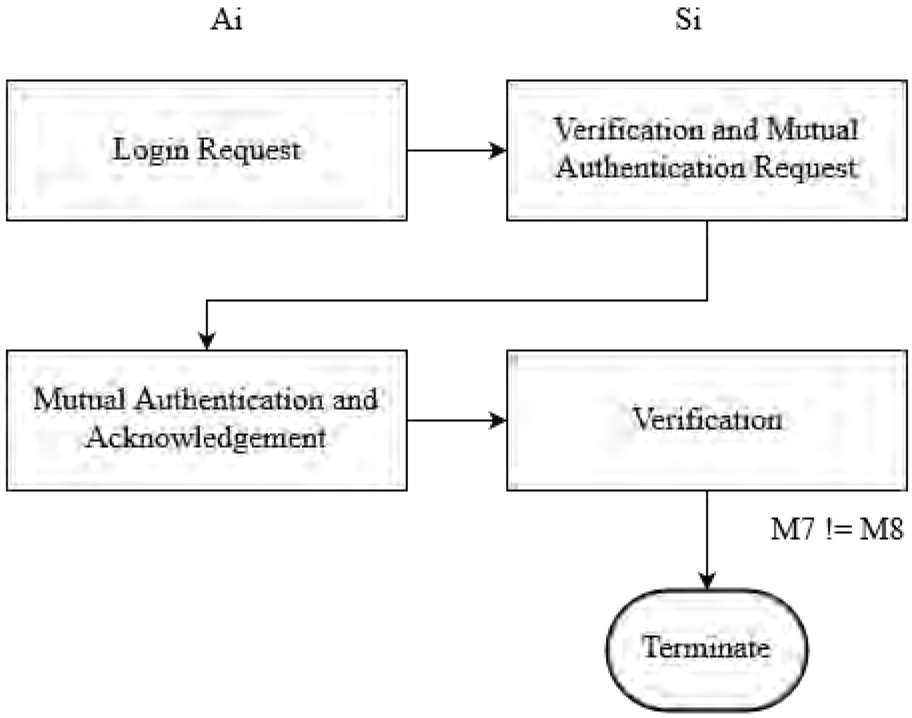

Fig. 6.3. Replay attack on proposed scheme which involves the attacker A_i and the server S_i

IV. Verification by the Server and the Point of Failure

The server S_i receives $\{ID_i, SID_i, M_7, Stat\}$ from the A_i. Then, it checks ID_i and SID_i. If ID_i is desired user id, SID_i is desired server id and $Stat = Auth$, then it calculates M_8 using (5.27) and compares whether $M_7 = M_8$ or not. But, they are not equal because recently generated R_{n3} doesn't match with the previously generated R_{n3} of M_7. Therefore, it discards the message and terminates the session. Fig. 6.3. shows the replay attack on proposed scheme.

6.6 Mutual Authentication

According to [15], generally if a scheme is insecure against impersonation attack and server masquerading attack, then it cannot provide mutual authentication. However we have shown that our scheme can provide security against impersonation attack at section 6.3 and server masquerading attack at section 6.4. Therefore, we can claim that our scheme provides mutual authentication.

6.7 Password and Smart Card Recovery

Our scheme also comes up with a very good and secure password and smart card recovery options. A user needs to provide a correct and trusted recovery contact during registration and follow the password/smart card recovery steps when necessary.

6.8 Proper Biometric Verification

If any scheme uses biometric templates directly, then there exists a possibility that sometimes it may fail to match the provided templates with stored templates. It is due to existence of noise, different orientation of imprinting the biometrics, etc. However, our scheme does not use templates directly. We use algorithms which are relied on biometric cryptosystem or cancellable biometrics technology and can release unique biometric key from the templates which are relatively close enough. This is how our scheme provides proper biometric verification.

6.9 Forgery Attack

The attacker may forge into the insecure channel and manage to get the messages used during login and authentication phase. But, the attacker cannot use these messages to gather any information to generate future login and authentication messages. We have already shown that how the scheme prevents the impersonation attack and the server masquerading attack. The attacker also may try to use the old messages to gain access. We have also shown that how our scheme can resist replay attack. So, considering all these analysis, we can say our scheme can resist forgery attack.

6.10 Session Key Support

The session key is required to conduct further secret communication between the user and the server after login. Our scheme provides a mechanism to generate session key during authentication phase. It reduces the overhead of computation and communication and also provides the opportunity to conduct further secret communication smoothly.

6.11 Comparison with Other Schemes

In this section, we will show a comparison between our scheme and the other schemes in terms of security features and functionality. The comparison is shown in Table 6.1.

Table. 6.1. Security features and functionality comparison

Properties	S_1	S_2	S_3	S_4	S_5
Prevents Password Guessing Attack	Yes	Yes	No	Yes	No
Prevents Security Key Stealing	Yes	No	No	No	No
Prevents User Impersonation Attack	Yes	Yes	No	Yes	No
Prevents Server Masquerading Attack	Yes	No	No	No	No
Prevents Replay Attack	Yes	Yes	Yes	Yes	Yes
Password Recovery	Yes	No	No	No	No
Smart Card Recovery	Yes	No	No	No	No
Provides Mutual Authentication	Yes	No	No	No	No
Provides Proper Biometric Verification	Yes	No	No	No	Yes
Prevents Forgery Attack	Yes	No	No	No	Yes
Supports Session Key	Yes	No	No	No	Yes

S_1 = Proposed, S_2 = Hwang et al. [11], S_3 = Das [14], S_4 = An [15], S_5 = Li et al. [16] scheme respectively

From the security features and functionality comparison presented in Table 6.1, we can see that the scheme presented by Hwang et al. [11] can provide security against password guessing attack, user impersonation attack, and replay attack. However, it cannot prevent security key stealing, server masquerading attack and forgery attack. Moreover, it cannot provide mutual authentication, proper biometric verification, password and smart card recovery phase, and support session key. The scheme presented by Das [14] can provide security only against replay attack. We can also see that the scheme presented by An [15] has similar security features and functionality like Hwang et al.'s scheme [11]. The scheme presented by Li et al. [16] can prevent replay attack and forgery attack. Moreover, it can provide proper biometric verification and support session key. However, it cannot prevent password guessing attack, user impersonation attack, security key stealing, and server masquerading attack. Also, it is unable to provide mutual authentication, password recovery phase, and smart card recovery phase. From the Table 6.1, it is clear that our proposed scheme has all the above mentioned security features and functionality.

6.12 Summary

In this chapter, we discussed the security analysis of our proposed scheme. Through security analysis, we have shown that it can prevent security attacks like password guessing attack, secret key stealing, user impersonation attack, server masquerading attack, replay attack, denial of service attack, forgery attack, etc. Moreover, it provides password, and smart card recovery options. Our scheme also supports session key agreement to ensure the further secret communication for reducing the overhead of computation and communication. We have depicted a comparison table with few of the existing schemes and our proposed scheme which clearly shows the security advantages of our scheme over those schemes.

CHAPTER 7
Implementation and Simulation

7.1 Implementation Scenario

We have implemented our proposed scheme and simulated it in local environment. However, it can be implemented in any network environment. The users can choose their usernames, biometrics and passwords freely. The registration center is a trusted third party who is trusted by both the user and the server. Here, registration center is involved in all the phases accept login and authentication phase. It can monitor the whole process and save necessary data if needed. It can halt any process at anytime if needed. The registration center keeps its private AES key secret and protects the key in any situation. The servers must register with registration center before starting its operation. The server keeps its private AES key secret and protects the key in any situation. The sensitive data are encrypted before saving into the database. The sensitive information of the messages is hashed before transmission. Only login and authentication phase uses insecure channel; all other phases use secure channel to exchange messages.

7.2 Implementation Tools

We simulated our work using HTML [32], CSS [33], JAVASCRIPT [34], PHP [35] and MYSQL [36]. We also used bootstrap [37], font-awesome CSS library [38] and jquery JAVASCRIPT library [39]. We also used Apache server [40] to run our simulation. We simulated the following phases: Server Registration Phase, User Registration Phase, Login and Authentication Phase, Password Change Phase, Password Recovery Phase, and Smart Card Recovery Phase. We also simulated following attacks: User Impersonation Attack, Server Masquerading attack, and Replay Attack. The prevention of other remaining attacks and weaknesses are theoretically discussed in chapter 6.

7.3 Simulation of Different Phases

The simulation of different phases of our proposed scheme is discussed in the following.

7.3.1 Server Registration Phase

The simulation of server registration phase is shown in Fig. 7.1. Here, we try to register a server (https://testserver.com). All the steps of server registration are shown in the two console panels labeled as server and registration center. After completion of registration process, the list of registered servers is illustrated in Fig. 7.2.

Fig. 7.1. Simulation of server registration phase which involves the server and the registration center

S.N.	Server Domain	Server ID	HKs
1	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	��0=\9���CWtBnYv��� E&K0S�n�x�/6�"� ���=�^����-P��
2	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	�<]K�f�$!��o����YB��!m��Yg����Y� $KM����q�q�8P�d��

Fig. 7.2. Registered server list

7.3.2 User Registration Phase

The simulation of user registration phase is shown in Fig. 7.3. Here, a user (Alice) is trying to register with a server (https://testserver.com). All the steps of user registration phase are shown in the three console panels labeled user, registration center and server. After completion of registration process, the list of registered users is illustrated in Fig. 7.4.

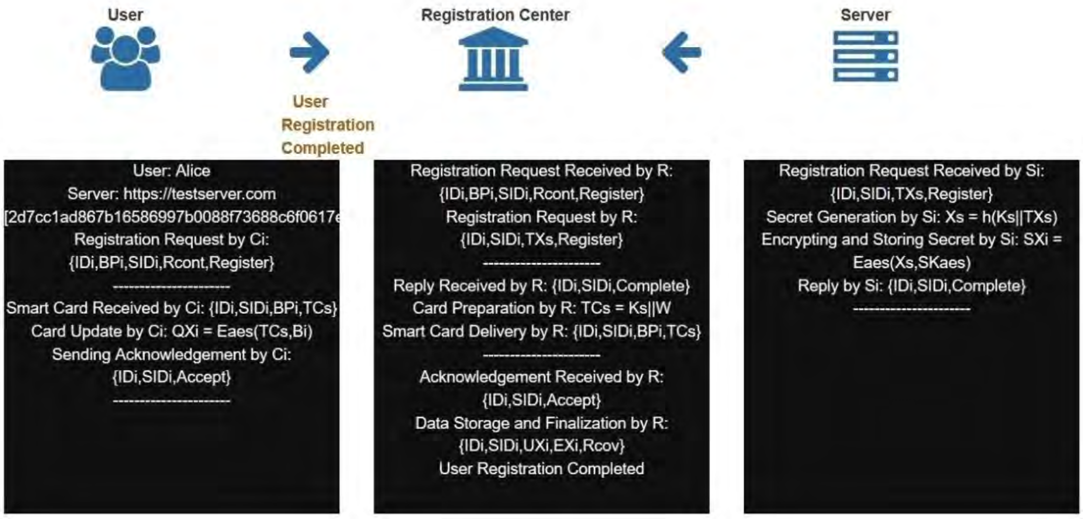

Fig. 7.3. Simulation of user registration phase which involves the user, the registration center and the server

S.N.	User ID	Server Domain	Server ID	UXi	EXi
1	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	��0=\9���CWtBnYv��� E&K0S�n�x�/6�"� ���=�^�����-P����^V� \��S�39!�~1t�˜#� �5A���� C��T����:G��	�^V�\��S�39!�~1t�˜#� �5A��9ㄅu�F�5�_?�� c�p�!�P�=
2	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	�<]K�f�$!��o�� ��YB��!m��Yg����Y� $KM������q�q�8P�d�岪 ��'C�����u� �T?Fj3Cu���2ب�[�S��z��} ?TH��	���'C�����u� �T?Fj3Cu���2ب�[;k�A? ��0A�`�TZ�V�. ����R��� ���bh��l#����` c�%^l�����R�8g�� � �����r����`�

Fig. 7.4. Registered user list

7.3.3 Login and Authentication Phase

The simulation of login and authentication phase is shown in Fig. 7.5. Here, a user (Alice) tries to get access to a server (https://testserver.com). All the steps of the login and authentication phase are shown in the two console panels labeled as user and server. Another console labeled as attacker shows how an attacker collects messages from an insecure channel. Also, a login log is maintained by server where information of every login is maintained. The login log is shown in Fig. 7.6.

Fig. 7.5. Simulation of login and authentication phase which involves the user and the server

S.N.	User ID	Server Domain	Server ID	Time
1	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:17:46
2	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	17-07-2017 01:17:19
3	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:16:58
4	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:09:29
5	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	12-06-2017 23:21:17

Fig. 7.6. Login log of the server

Moreover, the attacker maintains a database where messages captured during login and authentication phase are stored. The simulation of such a database is shown Fig. 7.7.

S.N.	User ID	Server Domain	Server ID	M1	M2	M4	M5	M7	Time
1	Alice	https://testserver.com	2d7cc1ad86...	101511f6fc...	3534653532...	2ca429a48f...	3636316237...	04e9ad1878...	17-07-2017 01:17:46
2	max	https://abc.com	70fc2ceba8...	16ad4e0f5d...	3035303338...	578126d261...	6266356438...	1db02cc89d...	17-07-2017 01:17:19
3	Alice	https://testserver.com	2d7cc1ad86...	3f4c0153ca...	3534653532...	73aaa875aa...	3062343637...	23be41b4fa...	17-07-2017 01:16:57
4	Alice	https://testserver.com	2d7cc1ad86...	72d8fdbe3f...	3534653532...	5e803b1fb7...	6166333730...	376d4568ab...	17-07-2017 01:09:29
5	max	https://abc.com	70fc2ceba8...	6fd3da3d91...	3035303338...	ecb3f92124...	3763643632...	e9b24e7f06...	12-06-2017 23:21:17

Fig. 7.7. Data center of the attacker

7.3.4 Password Change Phase

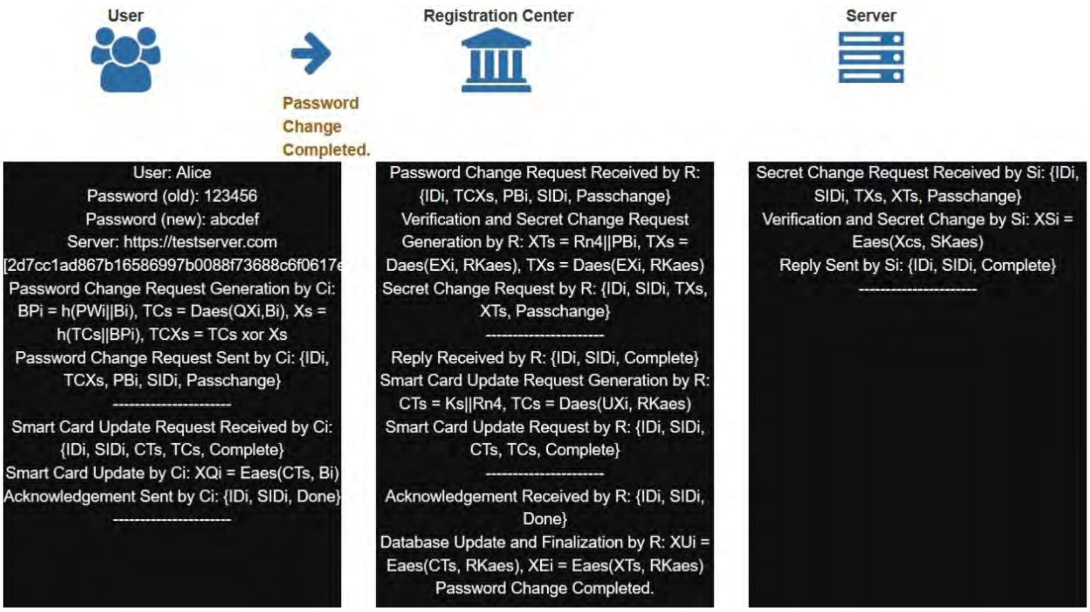

Fig. 7.8. Simulation of password change phase which involves the user, the registration center and the server

The simulation of password change phase is shown in Fig. 7.8. Here, a user (Alice) is trying to change her password. All the steps of password change phase are shown in the three console panels labeled user, registration center and server. A successful login after completion of this phase is shown in Fig. 7.9.

S.N.	User ID	Server Domain	Server ID	Time
1	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:30:44
2	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:17:46
3	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	17-07-2017 01:17:19
4	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:16:58
5	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:09:29

Fig. 7.9. Login log after password change

7.3.5 Password Recovery Phase

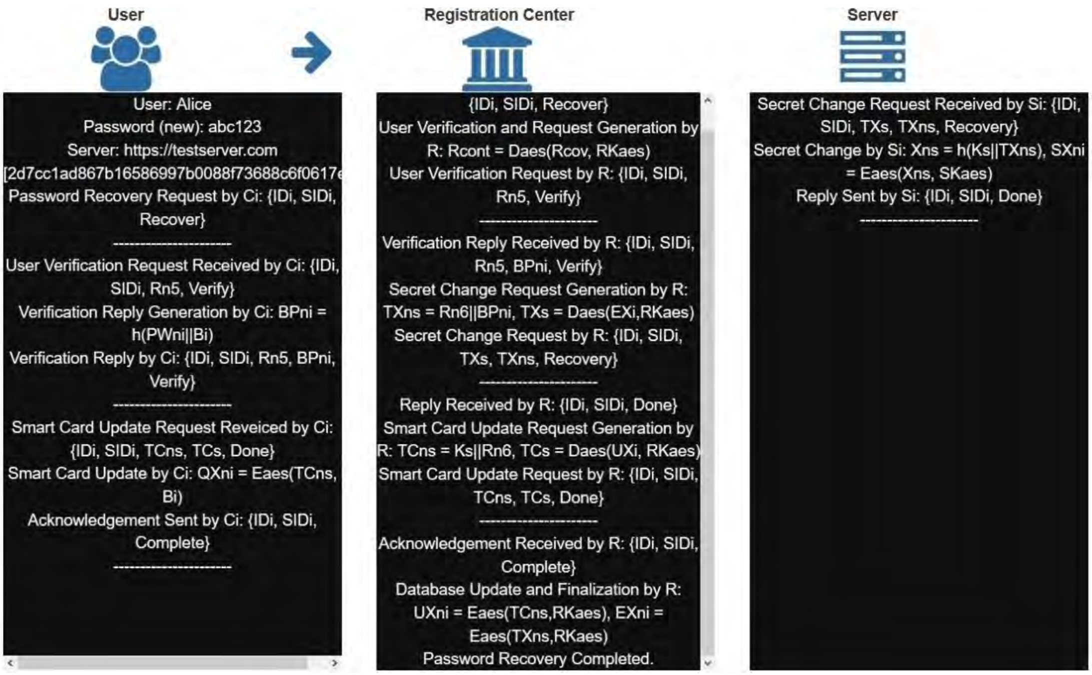

Fig. 7.10. Simulation of password recovery phase which involves the user, the registration center and the server

The simulation of password recovery phase is shown in Fig. 7.10. Here, a user (Alice) is trying to recover her password. All the steps of password recovery phase are shown in the three console panels labeled user, registration center and server. A successful login after completion of this phase is shown in Fig. 7.11.

S.N.	User ID	Server Domain	Server ID	Time
1	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 02:30:54
2	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:30:44
3	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:17:46
4	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	17-07-2017 01:17:19
5	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:16:58

Fig. 7.11. Login log after password recovery

7.3.6 Smart Card Recovery Phase

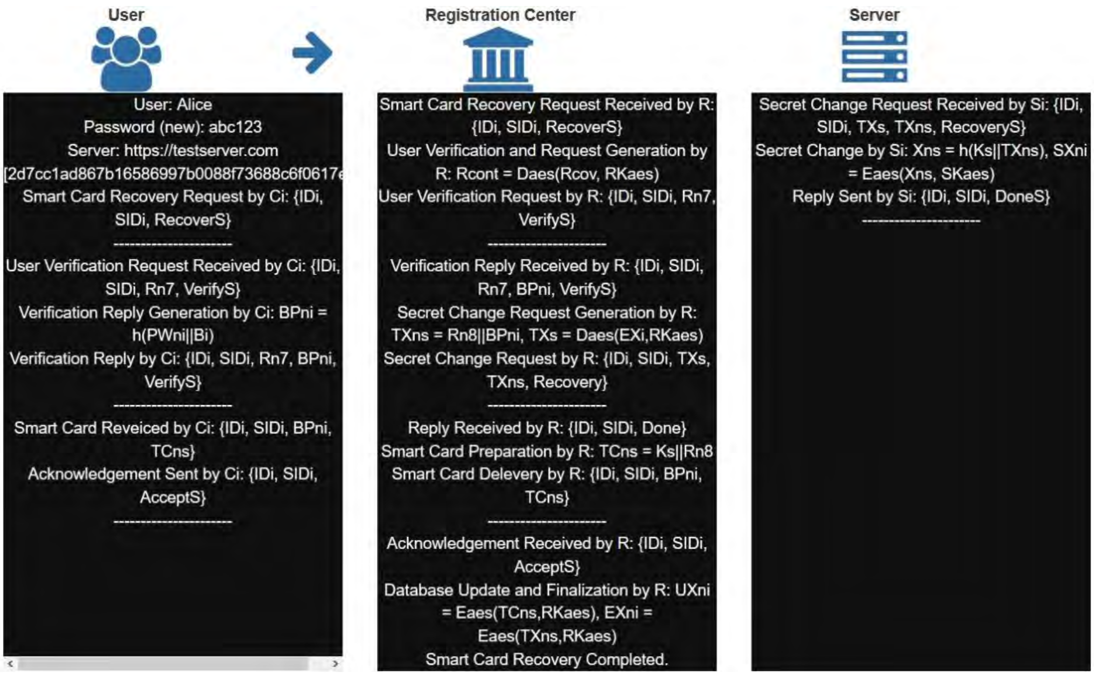

Fig. 7.12. Simulation of smart card recovery phase which involves the user, the registration center and the server

The simulation of smart card recovery phase is shown in Fig. 7.12. Here, a user (Alice) is trying to recover her smart card. All the steps of smart card recovery phase are shown in the three console panels labeled user, registration center and server. A successful login after completion of this phase is shown in Fig. 7.13.

S.N.	User ID	Server Domain	Server ID	Time
1	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 02:34:28
2	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 02:30:54
3	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:30:44
4	Alice	https://testserver.com	2d7cc1ad867b16586997b0088f73688c6f0617e6	17-07-2017 01:17:46
5	max	https://abc.com	70fc2ceba871112f7cf123d9196058c3f27c1dcc	17-07-2017 01:17:19

Fig. 7.13. Login log after smart card recovery phase

7.4 Simulation of Attacks

The simulation of several attacks is discussed in the following.

7.4.1 User Impersonation Attack

Fig. 7.14. Simulation of user impersonation attack which involves the attacker and the server

The simulation of user impersonation attack is shown in Fig. 7.14. Here, an attacker is trying to impersonate as a user (Alice) and trying to get access of a server (https://testserver.com). All the steps of this attack are shown in the two console panels labeled as attacker and server. As shown in the server console of the Fig. 7.14, the verification will fail and thus, attack won't be successful.

7.4.2 Server Masquerading Attack

The simulation of server masquerading attack is shown in Fig. 7.15. Here, an attacker is trying to masquerade as a server (https://testserver.com) and trying to hear from a user (Alice). All the steps of this attack are shown in the two console panels labeled as user and attacker. As shown in the user console of the Fig. 7.15, the verification will fail and thus, attack won't be successful.

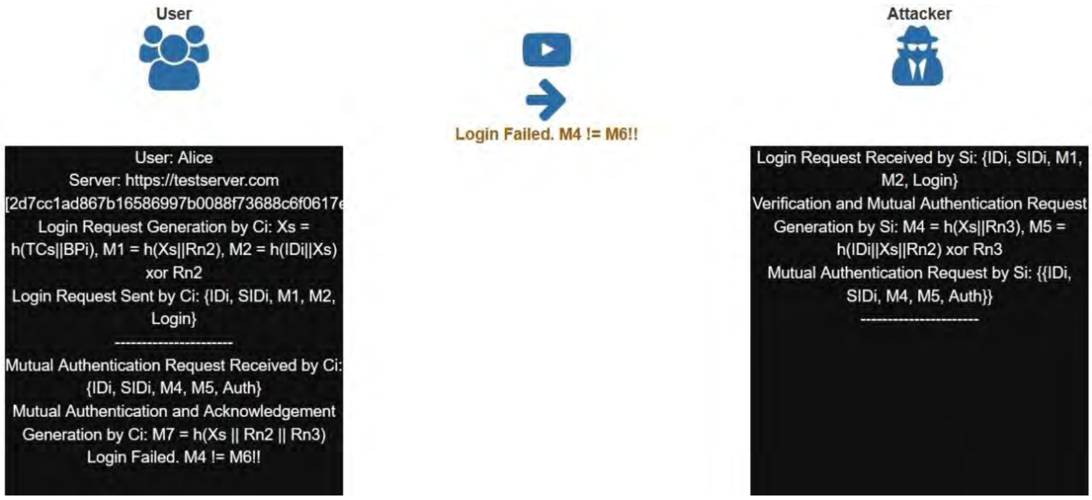

Fig. 7.15. Simulation of server masquerading attack which involves the user and the attacker

7.4.3 Replay Attack

The simulation of replay attack is shown in Fig. 7.16. Here, an attacker is trying to send old login messages of a user (Alice) to get access to a server (https://testserver.com). All the steps of this attack are shown in the two console panels labeled as attacker and server. As shown in the server console of the Fig. 7.16, the verification will fail and thus, attack won't be successful.

Fig. 7.16. Simulation of replay attack which involves the attacker and the server

7.5 Cost and Usability

Even though we implemented the proposed scheme for simulation, it can be implemented for real environment integrated with biometric device and smart card reader. The cost of practical implementation depends on what kind of biometric is used and also the quality of the device used during implementation. If we use a device which has fingerprint scanner and smart card reader integrated within it, then the cost of implementation will be around 7600Tk [41]. But, if iris scanner and smart card reader are used, then the cost of implementation will be around 16500Tk [42-43].

From the above discussion, we can say that device with fingerprint scanner and smart card reader can be used at personal environment. However, both devices can be used at industrial environment. There is a trade of between security and cost. We belief that the implementation cost is reasonable considering the level of security our proposed scheme can provide.

7.6 Summary

In this chapter, we discussed about the simulation of our proposed scheme. We used HTML [32], CSS [33], JAVASCRIPT [34], PHP [35] and MYSQL [36] to simulate our work. Additionally, we used bootstrap [37], font-awesome CSS library [38] and jquery JAVASCRIPT library [39] to design our interface, and we used Apache server [40] to run our applications. Here, using several figures, we show the simulation results of various phases our scheme and potential attacks that can be prevented by it.

CHAPTER 8
Conclusion and Future Work

In this thesis, we have presented a secure three factor user authentication scheme using biometric and smart card. Through security analysis, we have shown that our scheme outperforms existing schemes in terms of security and features. Our proposed scheme uses the strength of AES to prevent the attackers from stealing data as well as resists several attacks to ensure the security of the login and authentication mechanism. Moreover, it provides password, and smart card recovery options. Our scheme also supports session key agreement to ensure that further secret communication incurs reduced overhead of computation and communication. It also uses secure key generation process to generate biometric keys from biometrics. We have depicted a comparison in the Table 6.1 with few of the existing schemes and our proposed scheme which clearly shows the security advantages of our scheme over those schemes.

8.1 Future Work

Our password recovery phase and smart recovery phase are relatively complex and require little bit more time than other phases. In future, we will try to reduce the complexity and time of those phases. Though our scheme performs well at multi-server platform, but there is a drawback of using smart card at such platform. If the smart card is somehow unavailable to the user, then he cannot access any of the servers until it is recovered. The smart card can be replaced by some other factor in future to enhance the scheme and make it more suitable for multi-server platform.

References

[1] Lamport, L., "Password Authentication with Insecure Communication," Communications of the ACM, vol. 24(11), pp. 770-772, 1981.

[2] Morris, R., and Thompson, K., "Password security: A case history," Communications of the ACM, vol. 22(11), pp. 594-597, 1979.

[3] Bellovin, S.M., and Merritt, M., "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *IEEE Symposium on Research in Security and Privacy*, pp. 72-84 (1992).

[4] Spafford, E.H., "Opus: Preventing weak password choices," Computers & Security, vol. 11(3), pp. 273-278, 1992.

[5] Florencio, D. and Herley, C., "A large-scale study of web password habits," in *16th international conference on World Wide Web*, May, 2007, pp. 657-666 (2007).

[6] Ives, B., Walsh, K.R. and Schneider, H., "The domino effect of password reuse," Communications of the ACM, vol. 47(4), pp. 75-78, 2004.

[7] Bonneau, J., Herley, C., Van Oorschot, P.C. and Stajano, F., "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *IEEE Symposium on Security and Privacy*, May, 2012, pp. 553-567 (2012).

[8] Hwang, M.S. and Li, L.H., "A new remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 46(1), pp. 28-30, 2000.

[9] Shen, J.J., Lin, C.W. and Hwang, M.S., "A modified remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 49(2), pp. 414-416, 2003.

[10] Sun, H.M., "An efficient remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 46(4), pp. 958-961, 2000.

[11] Li, C.T. and Hwang, M.S., "An efficient biometrics-based remote user authentication scheme using smart cards," Journal of Network and computer applications, vol. 33(1), pp. 1-5, 2010.

[12] Khan, M.K., Zhang, J. and Wang, X., "Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices," Chaos, Solitons & Fractals, vol. 35(3), pp.519-524, 2008.

[13] Lin, C.H. and Lai, Y.Y., "A flexible biometrics remote user authentication scheme," Computer Standards & Interfaces, vol. 27(1), pp.19-23, 2004.

[14] Das, A.K., "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," IET Information Security, vol. 5(3), pp.145-151, 2011.

[15] An, Y., "Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards," BioMed Research International, pp.1-6, 2012.

[16] Li, X., Niu, J., Khan, M.K. and Liao, J., "Robust biometrics based three-factor remote user authentication scheme with key agreement," in *International Symposium on Biometrics and Security Technologies (ISBAST)*, July, 2013, pp. 105-110 (2013).

[17] Kocher, P., Jaffe, J. and Jun, B., “Differential Power Analysis,” in *Annual International Cryptology Conference*, August, 1999, pp. 388-397 (1999).

[18] Messerges, T.S., Dabbish, E.A. and Sloan, R.H., “Examining smart-card security under the threat of power analysis attacks,” IEEE Transactions on Computers, vol. 51(5), pp. 541–552, 2002.

[19] Rathgeb, C. and Uhl, A., “A survey on biometric cryptosystems and cancelable biometrics,” EURASIP Journal on Information Security, vol. 2011(1), pp. 3, 2011.

[20] Dodis, Y., Reyzin, L. and Smith, A., “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in *International Conference on the Theory and Applications of Cryptographic Techniques*, May, 2004, pp. 523-540 (2004).

[21] Gaddam, S.V. and Lal, M., “Efficient Cancelable Biometric Key Generation Scheme for Cryptography,” International Journal of Network Security, vol. 11(2), pp. 61-69, 2010.

[22] Lalithamani, N. and Soman, K.P., “An efficient approach for non-invertible cryptographic key generation from cancelable fingerprint biometrics,” in *International Conference on Advances in Recent Technologies in Communication and Computing*, October, 2009, pp. 47-52 (2009).

[23] Jagadeesan, A., Thillaikkarasi, T. and Duraiswamy, K., "Cryptographic key generation from multiple biometric modalities: Fusing minutiae with iris feature,” International Journal of Computer Applications, vol. 2(6), pp. 16-26, 2010.

[24] Stallings, W., “Cryptographic hash functions,” in Cryptography and Network Security: Principles and Practice, 5th ed., chap. 11, pp. 351–377, Pearson Education, India, 2011.

[25] Rivest, R., "The MD5 message-digest algorithm," Internet Engineering Task Force, RFC-1321, 1992.

[26] Eastlake 3rd, D. and Jones, P., "US secure hash algorithm 1 (SHA1)," Internet Engineering Task Force, RFC-3174, 2001.

[27] Barker, E., Barker, W., Burr, W., Polk, W. and Smid, M., "Secure hash standard (shs)," Federal Information Processing Standards (FIPS) Publications (PUBS), pp. 180-4, 2015.

[28] Dworkin, M.J., "Dworkin, M. J., "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," Federal Information Processing Standards (FIPS) Publications (PUBS) 202, 2015.

[29] Daemen, J. and Rijmen, V., "The design of Rijndael: AES-the advanced encryption standard," in Springer Science & Business Media, 2013.

[30] Pub, N.F., "Advanced Encryption Standard (AES)," Federal Information Processing Standards (FIPS) Publications (PUBS) 197, 2001.

[31] Stallings, W., "Advanced Encryption Standard," in Cryptography and Network Security: Principles and Practice, 5th ed., chap. 5, pp. 171–202, Pearson Education, India, 2011.

[32] HTML, https://www.w3schools.com/html/ [Last access on 27 Jul. 2017].

[33] CSS, https://www.w3schools.com/css/default.asp/ [Last access on 27 Jul. 2017].

[34] Javascript, https://www.w3schools.com/js/default.asp/ [Last access on 27 Jul. 2017].

[35] PHP, http://php.net/ [Last access on 27 Jul. 2017].

[36] Mysql, https://www.mysql.com/ [Last access on 27 Jul. 2017].

[37] Bootstrap, http://getbootstrap.com/ [Last access on 27 Jul. 2017].

[38] Fontawesome, http://fontawesome.io/ [Last access on 27 Jul. 2017].

[39] Jquery, https://jquery.com/ [Last access on 27 Jul. 2017].

[40] Apache, https://httpd.apache.org/ [Last access on 27 Jul. 2017].

[41] Fingerprint and Smart Card Reader, https://www.bayometric.com/hamster-pro-duo-scpiv-fingerprint-card-reader/ [Last access on 27 Jul. 2017].

[42] Iris Scanner, http://biometricsupply.com/iritech-irishield-usb-mk-2120u.html/ [Last access on 27 Jul. 2017].

[43] Smart Card Reader, https://www.aliexpress.com/item/Hot-sell-ID-Card-Reader/32222806111.html?spm=2114.search0301.4.19.7IdbUn/ [Last access on 27 Jul. 2017].