Design and Analysis of Secure Lightweight Authentication and Signcryption Schemes

PhD Thesis

By:

Shehzad Ashraf Chaudhry
71-FBAS/PHDCS/F11

Supervisor:

Dr. Syed Husnain Abbas Naqvi
Chairman, DCS & SE, FBAS, IIU

Co-Supervisor:

Prof. Dr. Muhammad Sher
Dean, FBAS, IIU

Department of Computer Science & Software Engineering

International Islamic University, Islamabad

(2016)

A dissertation submitted to the
Department of Computer Science,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy in Computer Science.

Department of Computer Science & Software Engineering, International Islamic University, Islamabad

Date

Final Approval

It is certified that we have examined the thesis report submitted by Shehzad Ashraf Chaudhry, Registration No. 71-FBAS/PhD(CS)/F11, and it is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the Doctor of Philosophy in Computer Science.

Committee:

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Shehzad Ashraf Chaudhry

Dedication

This thesis is dedicated to My family, especially to my father **Muhammad Ashraf Chaudhr**y, my mother **Mussarat Nahid Ashraf** and my beloved wife **Madiha Shehzad**.

Shehzad Ashraf Chaudhry

Acknowledgments

This thesis and all my efforts are fruitful only due to Allah Almighty, the Most Merciful and Beneficent, Who gave me strength to complete this task to the best of abilities and knowledge.

I would like to thank my supervisor **Dr. Syed Husnain Abbas Naqvi** and Co-Supervisor **Prof. Dr. Muhammad Sher**, who gave all their knowledge, guidance and support to boost my confidence and learning. I would also like to thank my beloved wife **Madiha Shehzad** who has supported me patiently and firmly during completion of my task.

My great acknowledgements are for my friends, students and colleagues especially for **Mr. Shahzad Siddique Chaudhry (BSSE F12), Mr. Khalid Mahmood, Mr. Tawab Shah, Mr. Zahid Mahmood, Mr. Anwar Ghani, Dr. Ali Daud, Mr. Asim Munir, Mr. Iftikhar Ali Khan, Mr. Mahmood ul Hassan, Mr. Jawwad Shakir** and **Mr. Imran Khan**. All of them encouraged and provided logistic and technical help to me. I must also thank **Prof. Dr. Muhammad Arshad Zia** who always supported me and encouraged me to start the thesis.

It will not be fair, if I forget **Mr. Syed Bilal Shah, Mr. Azib, Mr. Saeed, Mr. Saleem** and other supportive staff at DCS&SE, IIUI, who always encouraged me and helped me to complete official formalities at their earliest. I am thankful to all the supportive staff at DCS&SE, IIUI.

Last but not the least, I would like to thank my respected friends **Dr. Mohammad Sabzinejad Farash, Prof. Dr. Muhammad Khurram Khan, Dr. Saru Kumari, Dr. SK Hafizul Islam** and the most importantly **Prof. Dr. Taeshik Shon**, without their sincere support, I was not able to complete this task.

I owe all my achievements to my truly, sincere and most loving parents, sister, brothers and friends who mean the most to me, and whose prayers have always been a source of determination for me.

Abstract

The precipitated growth in the field of information and communication technologgy, assisted in making daily life more convenient and easy by providing an increasing number of services online: like shopping, healthcare, gaming, videos, and government services etc. All such online services are provided through public networks. Despite all these aids, the main problems of such networks are security and privacy as all public networks are inherently insecure. The adversary can easily intercept, modify and eavesdrop the channel. Therefore, ensuring the security of messages on such channels has become an important issue. Till the time, a number of cryptographic protocols comprising various primitives exist in literature.

The protocols based on symmetric key cryptography like: symmetric encryption/decryption; one way hash/mac functions; exclusive OR etc. are extremely lightweight when compared with all public key primitives. Hence, one has to prefer symmetric primitives in resource constrained environments, but keeping in mind the sensitivity of tasks (e.g. financial, healthcare) carried out by cryptographic protocols which are also having additional threats as compared to traditional threats, asymmetric cryptography looks more promising, which can resists impersonation, password guessing and replay attacks.

In this thesis, we develop some cryptographic protocols majoring in five sub areas: (1) Two party two-factor authentication schemes, (2) Two party three-factor authentication schemes, (3) A mobile handover authentication scheme, (4) Multiserver authentication schemes, and (5) a signcryption scheme.

Four two-factor authentication schemes are proposed to authenticate communicating parties and to share a session key for confidential message exchange. Two three-factor authentication schemes are proposed. Two multi-server authentication schemes are designed. Similarly, a mobile handover authentication scheme is proposed, where a moving mobile node and an access point mutually authenticate each other. Finally, we propose a signcryption scheme. Then, we develop an electronic payment system based on the proposed signcryption scheme to secure electronic transactions.

A rigorous security analysis using provable security model has been carried out for the developed protocols. We have also utilized the formal security model of popular automated tool ProVerif to prove the robustness of proposed schemes.

Contents

List of Figures

List of Tables

List of Publications

- **Published Articles in SCI/SCIE Journals**

1. Shehzad Ashraf Chaudhry, A secure biometric based multi-server authentication scheme for social multimedia networks, Multimedia Tools and Applications, DOI: 10.1007/s11042-015-3194-0 (IF 1.342) (2016)

2. Shehzad Ashraf Chaudhry, Husnain Naqvi , Mohammad Sabzinejad Farash, Taeshik Shon, Muhammad Sher, An improved and robust biometrics based three factor authentication scheme for multiserver environments, Journal of Supercomputing, DOI: 10.1007/s11227-015-1601-y (IF 0.858) (2015)

3. Shehzad Ashraf Chaudhry, Mohammad Sabzinejad,Farash, Husnain Naqvi , SK Hafizul Islam, Taeshik Shon, A Robust and Efficient Privacy Aware Handover Authentication Scheme for Wireless Networks, Wireless Personal Communications, DOI: 10.1007/s11277-015-3139-y (IF 0.653) (2015)

4. Shehzad Ashraf Chaudhry, Khalid Mahmood, Husnain Naqvi , Muhammad Khurram Khan, An Improved and Secure Biometric Authentication Scheme for Telecare Medicine Information Systems based on Elliptic Curve Cryptography, Journal of Medical Systems, DOI: 10.1007/s10916-015-0335-y (IF 2.213) (2015)

5. Shehzad Ashraf Chaudhry, Husnain Naqvi , Muhammad Sher , Mohammad Sabzinejad Farash, Mahmood ul Hassan, An improved and provably secure privacy preserving authentication protocol for SIP, Peer to peer networking and applications, 2015, DOI: 10.1007/s12083-015-0400-9 (IF 0.632) (2015)

6. Shehzad Ashraf Chaudhry, Mohammad Sabzinejad Farash, Husnain Naqvi , Muhammad Sher, A Secure and Efficient Authenticated Encryption for Electronic Payment Systems using Elliptic Curve Cryptography, Electronic Commerce Research, 16(1), 113-139 2016, DOI: 10.1007/s10660-015-9192-5 (IF 1.773)

7. Shehzad Ashraf Chaudhry , Mohammad Sabzinejad Farash, Husnain Naqvi, Saru Kumari, Muhammad Khurram Khan, An enhanced privacy preserving remote user authentication scheme with provable security, Security and Communication Networks, 2015, DOI: 10.1002/sec.1299 (IF 0.72) (2015)

8. Shehzad Ashraf Chaudhry, Husnain Naqvi , Taeshik Shon, Muhammad Sher, Mohammad Sabzinejad farash, Cryptanalysis and improvement of an improved

two factor authentication protocol for Telecare Medical Information Systems, Journal of Medical Systems, 2015, DOI: 10.1007/s10916-015-0244-0 (IF 2.213) (2015)

9. Shehzad Ashraf Chaudhry, Comment on 'Robust and efficient password authenticated key agreement with user anonymity for session initiation protocol-based communications', IET Communications, pp 1, DOI: 10.1049/iet-com.2014.1082 (IF 0.742) (2015)

10. Shehzad Ashraf Ch, Nizam uddin, Muhammad Sher, Anwar Ghani, Husnain Naqvi, and Azeem Irshad. An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography. Multimedia Tools and Applications, pages 1-13, DOI: 10.1007/s11042-014-2283-9 (IF 1.342) (2014)

- **Published Papers in Peer-reviewed Conferences**

1. Shehzad Ashraf Chaudhry, Khalid Mahmood, Husnain Naqvi, Muhammad Sher, A secure authentication scheme for session initiation protocol based on elliptic curve cryptography, 13th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC-2015) Liverpool, United Kingdom, October, 26-28, 2015

2. Noor ul Amin, Muhammad Asad, Nizamuddin , Shehzad Ashraf Chaudhry , An Authenticated Key Agreement with Rekeying for Secured Body Sensor Networks Based on Hybrid Cryptosystem, IEEE, ICNSC 2012 , China, 11-14 April, 2012, 10.1109/ICNSC.2012.6204902

3. Shehzad Ashraf Ch , Nizamuddin, Muhammad Sher, Public Verifiable Signcryption Schemes with forward secrecy based on Hyperelliptic Curve Cryptosystem, Communications in Computer and Information Science, 2012, Volume 285, pp. 135-142,, DOI: 10.1007/978-3-642-29166-1˙12

4. Husnain Naqvi, Shehzad Ashraf Chaudhry, Khalid Mahmood, An improved Authentication Protocol for SIP-based VoIP, International conference on recent advances in computer systems (RACS-2015 held in Hail KSA on November, 30 to December, 1, 2015.

Other related Published Articles in SCI/SCIE Journals

1. Mohammad Sabzinejad farash, Omer Nawaz, Khalid Mahmood, Shehzad Ashraf Chaudhry, Muhammad Khurram Khan, A Provably Secure RFID Authentication

Protocol Based on Elliptic Curve for Healthcare Environments, Journal of Medical Systems, 2015, DOI: 10.1007/s10916-016-0521-6 (IF 2.213) (2016)

2. Khalid Mahmood, Shehzad Ashraf Chaudhry, Husnain Naqvi, Taeshik Shon and Hafiz Farooq Ahmed, A Lightweight Message Authentication Scheme for Smart Grid Communications in Power Sector, Computers and Electrical Engineering, DOI: 10.1016/j.compeleceng.2016.02.017 (IF 0.817) (2016)

3. Azeem Irshad, Muhammad Sher, Shahzad Ashraf Chaudhary, Husnain Naqvi and Mohammad Sabzinejad Farash, An Efficient and Anonymous Multi-Server Authenticated key agreement based on Chaotic Map without engaging Registration Centre, Journal of Supercomputing DOI: 10.1007/s11227-016-1688-9 (IF 0.858) (2016)

4. Azeem Irshad, Muhammad Sher, Eid Rehman, Shehzad Ashraf Ch, Mahmood Ul Hassan, and Anwar Ghani. A single round-trip sip authentication scheme for voice over internet protocol using smart card. Multimedia Tools and Applications, pages 1-18, 2013. DOI: 10.1007/s11042-013-1807-z (IF 1.342) (2014)

5. Saru Kumari, Shehzad Ashraf Chaudhry, Fan Wu, Xiong Li, Mohammad Sabzinejad Farash, Muhammad Khurram Khan, An improved smart card based authentication scheme for session initiation protocol, Peer to peer networking and applications, (IF 0.632) (2015)

6. Mohammad Sabzinejad Farash, Shehzad Ashraf Chaudhry, Mohammad Heydar, S. Mohammad Sajad Sadough, Saru Kumari, Muhammad Khurram Khan, A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security, International Journal of Communication Systems, DOI: 10.1002/dac.3019 (IF 1.106) (2015)

7. Azeem Irshad, Muhammad Sher, Muhammad Shahzad Faisal, Anwer Ghani, Mahmood Ul Hassan, and Shehzad Ashraf Ch. A secure authentication scheme for session initiation protocol by using ecc on the basis of the tang and liu scheme. Security and Communication Networks, 2013, DOI: 10.1002/sec.834 (IF 0.72) (2014)

8. Mohammad Heydari, S. Mohammad Sajad Sadough, Mohammad Sabzinejad Farash, Shehzad Ashraf Chaudhry, Khalid Mahmood, An Efficient Password-Based Authenticated Key Exchange Protocol with Provable Security for Mobile Client-Client Networks, Wireless Personal Communications, DOI: 10.1007/s11277-

015-3123-6 (IF 0.653) 2015.

9. Mohammad Heydari, S. Mohammad-Sajad Sadough, Shehzad Ashraf Chaudhry, Mohammad Sabzinejad Farash, Mohammad Reza Aref. "An Improved Authentication Scheme for Electronic Payment Systems in Global Mobility Networks." Information Technology And Control 44.4: 387-403. (IF 0.623) (2015)

10. Mojtaba Alizadeh, Mazdak Zamani, Sabariah Baharun, Azizah Abdul Manaf, Kouichi Sakurai, Hiroki Anada, Hassan Keshavarz, Shehzad Ashraf Chaudhry, Muhammad Khurram Khan. "Cryptanalysis and Improvement of" A Secure Password Authentication Mechanism for Seamless Handover in Proxy Mobile IPv6 Networks"." PloS one 10, no. 11 (IF 3.234) (2015).

Chapter 1

Introduction

The precipitated growth in the field of information and communication technology, assisted in making daily life more convenient and easy by providing an increasing number of services online: like shopping, healthcare, gaming, videos, and government services etc. All such online services are provided through public networks. Despite all these aids the main problems of such networks are security and privacy as all public networks are inherently insecure. The security of wireless networks has become more important because all wireless networks are open architecture as well as having limited computation and battery resources. The adversary can easily intercept, modify and eavesdrop the channel. Therefore, ensuring the security of messages on such channels has become an important issue [1].

A number of security solutions are available to ensure confidentiality, authenticity, integrity and availability. Some of such solutions are based on the symmetric cryptography, where two participants share a same key and such solutions can ensure confidentiality and integrity. While some other solution are based on asymmetric cryptography, where each participant is having a public/private key pair can also ensure the authenticity.

Besides traditional security, the cryptography has been promoted to some other important requirements like: non-repudiation, sender and message authenticity and most importantly the user privacy and anonymity. Due to the nature of required resource efficiency, the cryptographic primitives based on symmetric cryptography looks more desirable, but keeping in mind the sensitivity of tasks (e.g. financial, health-care) carried out by such schemes which are also having additional threats as compared to traditional threats, asymmetric cryptography looks more promising which can resist impersonation, password guessing and replay attacks. Hence, a trade-off between the two is the need of time.

1.1 Objectives and Scope

This thesis is devoted to develop some security solutions majoring in authentication and authenticated encryption with an emphasis on (1) Two-factor authentication (2) Three-factor authentication (3) Mobile handover authentication (4) Multi-server authentication and (5) Signcryption/Authenticated encryption. Initially, a comprehensive analysis of some recent authentication and signcryption schemes is performed. It is shown that numerous existing schemes are having some design flaws resulting in severe limitations like :(1) High computation cost due to usage of modular exponentiation (2) Vulnerabilities to impersonation and related attacks and (3) Lacking user anonymity and privacy. Then, we design some authentication and signcryption schemes to overcome the weaknesses. The proposed schemes are carefully designed to cater the weaknesses of the existing schemes. The proposed schemes exploit the secure and lightweight properties of elliptic curve, symmetric key cryptography and identity based cryptography. It is worth mentioning that a number of schemes based on these cryptosystems exists in literature but we find that many such schemes are having some design flaws. Some salient features of proposed schemes as compared with related existing schemes are as follows:

1. Proposed schemes are secure under the threat model of automated tool ProVerif.
2. Proposed schemes are secure under the random oracle model.
3. Proposed schemes achieve computation and communication efficiency.
4. Proposed schemes ensure traditional security as well as authenticity, non-repudiation, anonymity and privacy.
5. Proposed schemes resist the known sophisticated attacks.

1.2 Contributions

The main emphasis of the thesis is on authentication and signcryption with five sub areas: (1) Two-factor authentication, (2) Three-factor authentication (3) Mobile handover authentication, (4) Multi-server authentication and (5) Signcryption. We have employed ECC and symmetric cryptography in most of the schemes and avoided identity based cryptosystems (IBC) and bilinear pairing. However, a multi-server authentication scheme is designed on the principles of IBC and pairings. The necessity of the usage of IBC and pairing was the reason to eliminate

the intervention of registration server during authentication between a service provider and the user. The contributions of the thesis are underlined as follows:

1.2.1 Password based Two-factor Authentication Schemes

Three anonymous two-factor authentication schemes based on ECC are proposed for (1) Session initiation protocol (SIP), (2) Remote users and (3) Telecare medicine-information systems (TMIS) respectively. Furthermore, an extremely lightweight authentication scheme is designed using only symmetric cryptography primitives.

1.2.2 Biometric based Three-factor Authentication Schemes

Two biometric based three-factor authentication schemes are proposed for Telecare medicine-information systems, one based on ECC and other on symmetric cryptography primitives.

1.2.3 Mobile Handover Authentication Scheme

An authentication scheme is proposed to facilitate the handover process of a mobile node while moving from the range of an access point to another. The mobile handover authentication scheme is also based on ECC.

1.2.4 Multiserver Authentication Schemes

Two multi-server authentication schemes are proposed. One scheme for the environments where the service providers are assumed to be honest. While other scheme is for the environments, where the service providers are not trusted. The former scheme is designed on the principles of ECC, while the latter is developed using IBC and bilinear mappings.

1.2.5 Signcryption Scheme

A signcryption scheme sometimes also referred as authenticated encryption is proposed using ECC. Furthermore, an e-payment system based on proposed signcryption scheme is designed to facilitate the customer for secure online transactions.

1.2.6 Thesis outline

The rest of the thesis is organized as follows:

- Chapter 2, explains the mathematical background pertaining to the thesis along with some mathematical hard problems, the common adversarial model and a brief introduction to the formal automated tool ProVerif.
- Chapter 3, cryptanalyzes the recent two-factor authentication protocol for session initiation protocol (SIP) based on ECC by Tu et al. [2] and Farash et al [3]. Furthermore, a privacy preserving two-factor authentication protocol for SIP using ECC is proposed. The proposed scheme is provably secure in the random oracle model and under the formal threat model of ProVerif.
- Chapter 4, cryptanalyzes a recent remote user authentication scheme by Huang et al. [4] and proved their scheme to be vulnerable to impersonation attack. Therefore, a privacy preserving remote user authentication scheme is proposed. The proposed scheme is more secure and lightweight than related existing schemes. The security of the proposed scheme is instantiated using the random oracle model and under ProVerif security model.
- Chapter 5, is devoted to explain an extremely lightweight authentication scheme using only symmetric key cryptography. Initially, vulnerabilities of some of the existing anonymous authentication schemes based on symmetric key cryptography are described. Then a cryptanalysis of the most recent scheme presented by Kumri et al. [5] is performed to show its weaknesses. Then an anonymous authentication scheme is proposed. The security analysis of proposed anonymous authentication scheme is instantiated using random oracle model. Furthermore, the security of the proposed scheme is also validated under the formal threat model of automated tool ProVerif supplemented by a rigorous security discussion.
- Chapter 6, first discusses the telecare medicine information system (TMIS) architecture and the need of authentication for TMIS access. Then an analysis of recent authentication schemes for TMIS is performed supplemented by a cryptanalysis of a recent authentication scheme proposed by Islam and Khan [6] to show its weaknesses against user and server impersonation attacks. Then, we proposed an improved ECC based authentication scheme to overcome the weaknesses of existing schemes followed by a rigorous security and performance discussion.
- Chapter 7, discusses the need of three-factor authentication, followed by a brief introduc-

tion to biohashing. Then, a brief analysis of some recent biometric based three-factor authentication schemes is performed followed by the cryptanalysis of the most recent biometric based three-factor authentication scheme by Lu et al.'s [7] to show its weaknesses to user impersonation, server impersonation and anonymity violation attacks. Furthermore, an ECC based three-factor authentication scheme is proposed to overcome the weaknesses. The security of the proposed scheme is instantiated using the automated tool ProVerif.

- Chapter 8, is devoted to develop an extremely lightweight three-factor authentication scheme based on only symmetric key primitives. Initially, an analysis of some recent three-factor authentication schemes based on symmetric key primitives is performed followed by the cryptanalysis of a recent three-factor scheme proposed by Mir and Nikooghadam [8] to explain its weaknesses against smart card stolen and anonymity violation attacks. Then an improved three-factor authentication scheme based on only lightweight symmetric key primitives is proposed. The security of proposed scheme is instantiated under the random oracle model and under the formal threat model of ProVerif.

- Chapter 9, gives a brief introduction of the mobile handover architecture and its security requirements. Then an analysis of some of the recent handover authentication schemes is performed followed by a cryptanalysis of most recent Li et al.'s scheme [9]. Which shows scheme's incapability to resist access point impersonation attack. Then, an improved handover authentication scheme based on ECC is proposed. The proposed scheme is provably secure under random oracle model and under the threat model of automated tool ProVerif.

- Chapter 10, first describes the need of multi-server authentication followed by an analysis of some recent multi-server authentication schemes and cryptanalysis of two most recent Lu et al.'s schemes [10, 11] to show the weaknesses. Furthermore, an ECC based three-factor authentication scheme for securing multi-server architecture is proposed. The proposed scheme is provably secure under the random oracle model and under the threat model of automated tool ProVerif.

- Chapter 11, introduces the concept of multi-server authentication where the service providers are not far granted as trusted. Then, the cryptanalysis of a most recent such authentication scheme is performed to show its weaknesses. The scheme is for cloud computing environments by Tsai and Lo [12] . Then, a bilinear mapping based multi-server authentication scheme is proposed. The proposed scheme is provably secure

under random oracle model and under the threat model of automated tool ProVerif.

- Chapter 12, introduces the concept of signcryption followed by an analysis of the recent signcryption schemes and cryptanalysis of a most recent signcryption scheme proposed by Yang et al. [13]. Then an ECC based signcryption scheme is proposed. Furthermore, an e-payment system is proposed based on proposed signcryption schemes. The proposed schemes are secure under the threat model of automated tool ProVerif.
- Finally, a conclusion is made in chapter 13.

Chapter 2

Mathematical Background

In this chapter, we give a brief discussion relating to symmetric key primitives, elliptic curve cryptography (ECC), identity based cryptography, bilinear mapping along with the computational hard problems. The chapter also discusses the common adversarial model and biohashing.

2.1 Symmetric Key Cryptography Basics

Symmetric encryption/decryption are most common methods to ensure message confidentiality. Symmetric encryption is also referred as private/single key encryption. The encryption algorithm transforms a number or some string into a random cipher text based on the shared key. While, the decryption algorithm uses same shared key and the cipher text and regenerates the original plain. Similarly such settings may include an algorithm for shared key generation. We can define these as follows:

2.1.1 Key Generation

Let $KEYS(sed)$ is the set of all strings with non-zero probability to be a shared key and let $\mathcal{K}$ be the algorithm which returns a key. Then $K \leftarrow \mathcal{K}$ is denoted as a distinct execution of $\mathcal{K}$ which returns K.

2.1.2 Symmetric Encryption

The symmetric encryption algorithm SE takes $K \in KEYS(sed)$ along with some arbitrary *plaintext* $P \in \{0,1\}^*$ and outputs a *ciphertext* $C \in \{0,1\}^* \cup \{\perp\}$. We can formally write $C = SE_K(P)$ as a distinct execution of SE which returns C, while the input is P and K.

2.1.3 Symmetric Decryption

The symmetric decryption algorithm SD takes $K \in KEYS(sed)$ along with some *ciphertext* $C \in \{0,1\}^*$ and outputs the corresponding *plaintext* $P \in \{0,1\}^* \cup \{\perp\}$. We can formally write $P = SD_K(C)$ as a distinct execution of SD which returns P, while the input is C and K.

Following are the characteristics, to qualify a secure symmetric encryption scheme:

- Computationally, it is infeasible to compute $P = SE(C)$, if C is known but K is not known. This characteristic is called the *confidentiality*.
- Computationally, it is impractical to figure out K, if both P and C are known. This is termed as *resistance to known plain text and known ciphertext* property.

2.1.4 One-way Hash Functions

One-way hash function $H : \{0,1\}^* \rightarrow Z_q^*$ produces fixed size output code $C = H(S)$ by taking random size input string S. The produced output is often designated as hash value or hash code. Trivial modification in the input string S can bring nontrivial change in the output C. Following characteristics must be met to qualify a secure hash function:

- Computationally, it is effortless to compute $C = H(S)$, if S is specified.
- Computationally, it is impractical or absurd to figure out S, if $C = H(S)$ is specified.
- It is tedious task to know two inputs S and T such that $H(S) = H(T)$. This characteristic is recognized as collision resistance property.

2.2 Public Key Cryptography

Public key cryptography/infrastructure (PKI) is also referred as asymmetric cryptography. *PKI* is a class of protocols relies on some algorithms using a pair of keys out of these, one key is called *private* and the other is called *public*. In *PKI*, the public keys of all the participants are known to each other. The public keys are also accessible to all outsiders. The key pair perform inverse operations for example: encryption by a public key of $\mathcal{X}$ can only be decrypted using $\mathcal{X}$'s private key and vice versa. Typically, the confidentiality is achieved by encrypting a message using public key of the receiver. Similarly, the sender's authenticity is achieved by encrypting a message using his own private key, so any one can decrypt the message using his public key. Two most common classes of *PKI* are described below:

2.2.1 Conventional PKI

Currently a number of conventional *PKI* techniques are available. A loose characterization includes RSA, ElGamal and DSA. Perhaps RSA is the most common and popular technique, which relies on large integer factorization problem. RSA can be used for both confidentiality and digital signatures. DSA stands for digital signature algorithm and can be used only for digital signature. ElGamal is based on discrete logarithm in a finite field. Readers are encouraged to refer to [14] for details regarding conventional *PKI*.

2.2.2 Elliptic Curve Cryptography

Recently, elliptic curve cryptography (ECC) has got much attention because of it's lightweight operations, which is also the main cause of it's dominance over classical public key cryptosystems. The best known algorithms for classical cryptosystems have a sub-exponential complexity while that of curve based cryptosystems has exponential complexity. So far a given level of security (2^n), the key size for classical cryptosystems grows like n^3, while for curve based cryptography it grows as n^2. As an implication curve based cryptosystems are now suggested to be used for the new products, where backward compatibility is not required. Elliptic curve has proved itself as a base for lightweight cryptography. It's widespread is because of the low cost operations, less memory and low communication cost as compared to classical public key cryptosystems.

ECC E/F_p is illustrated as set of points over F_p (a prime field) and is based on some chosen

real non-singular elliptic curve defined as follow:

$$E_p(a,b) : y^2 = x^3 + ax + b \mod p \tag{2.1}$$

In former equation 2.1 $a, b \in F_p$ and $4a^3 + 27b^2 \mod p \neq 0$ for a large prime p. The integers a, b both define the curve. A point (x, y) over $E_p(a, b)$ must verifies the former elliptic curve equation. Following are two main operations pertaining to ECC:

1. **Point Addition:**
   For all points P and Q, the point addition can be defined as:
   (a) $P + O = P$, where O is taken as additive identity and
   (b) $O = -O$
   (c) The additive inverse of a point P is having same x coordinate, while having additive inverse of y coordinate that is if $P = (x, y)$ then $-P = (x, -y)$
   (d) $P + -P = O$ point at infinity.
   (e) If $P = (x_p, y_p)$ and $Q = (x_q, y_q)$ and $P \neq Q$ then the addition of P, Q can be defined as $R = P + Q$, which can be calculated as:

$$x_r = (\lambda^2 - x_p - x_q) \mod P \tag{2.2}$$
$$y_r = (\lambda(x_p - x_r) - y_p) \mod P \tag{2.3}$$

   Where,

$$\lambda = \begin{cases} \frac{3x_p^2 + a}{2y_p} \mod P & \text{if } P = Q, \\ \frac{y_q - y_p}{x_q - x_p} \mod P & \text{if } P \neq Q \end{cases} \tag{2.4}$$

2. **Scalar Point Multiplication**
   Scalar Multiplication is defined as repeated addition so $vR = R + R + R..... + R$ (v *times*).

ECC provides same level of security as of traditional public key cryptography like RSA, DSA and DH with lesser parameters size [14].

2.2.3 Identity based Cryptography

In 1984, Adi Shamir [15] introduced the concept of identity based cryptography (IBC). IBC allows users to authenticate each other based on their own credentials like: telephone number, email address, name etc. The use of IBC for authentication ultimately ease the generation and storage of public and private key pairs in PKI. Utilization of IBC is dependent on a trusted third party termed as private key generator (PKG), which is responsible for generation of identity based key certificates of the participants. Once a participant receives his certificate can generate signatures, perform encryption and can participate in mutual authentication with other participants.

2.2.4 Bilinear Pairing

Bilinear pairing was introduced by Menezes et al. [16] after their proposed MOV attack on discrete logarithm problem. The main idea was to transport the discrete logarithm on a designated class of elliptic curve. Till then a number of cryptographic protocols [17–24] are proposed based on bilinear mapping. Bilinear mapping can be defined as follows:
Let G_1, G_2, G_3 are the three cyclic groups of order p, where p is sufficiently large prime. G_1, G_2 are additive and G_3 be the multiplicative group. The bilinear mapping e can be written as:

$$e : G_1 \times G_2 \rightarrow G_3$$

Where, e must satisfy the following conditions:

1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}, P \in G_1, Q \in G_2$ *and* $a, b \in Z_p*$.
2. Non Degeneracy: $e(P, P) \neq 1$.
3. Computability: There exists a polynomial time efficient algorithm to compute $e(P, Q)$.

2.2.5 Computational Hard Problems

This subsection elaborates some computationally hard problems useful in the thesis.
Definition 1. *[Collision resistant property aimed at secure hash functions] Prearranged collision resistant secure hash function $H(.)$. The likelihood that an adversary $\mathcal{A}$ can discover a couple $(Str_1 \neq Str_2)$ like $H(Str_1) = H(Str_2)$ is demarcated as $Adv_{\mathcal{A}}^{HASH}(t) = Prb[(Str_1, Str_2) \Leftarrow_r \mathcal{A} : (Str_1 \neq Str_2)$ and $H(Str_1) = H(Str_2)]$, where $\mathcal{A}$ is permitted*

to choose a couple (Str_1, Str_2) *arbitrarily.* $\mathcal{A}$*'s benefit is computed over the arbitrary selections taken up within polynomial time* (t)*. The collision resistant property infers that* $Adv_{\mathcal{A}}^{HASH}(t) \leq \epsilon$ *for any sufficiently small* $\epsilon > 0$*.*

Definition 2. *[Elliptic Curve Discrete Logarithm Problem (ECDLP)]* Against two specified random points $U, V \in E_p(a, b)$, compute a scalar x such that $U = xV$. The likelihood that a polynomial time (t) bound adversary $\mathcal{A}$ can calculate x is as given: $Adv_{\mathcal{A}}^{ECDLP}(t) = Prb[(\mathcal{A}(U, V) = x : x \in Z_p]$. The ECDLP supposition infers that $Adv_{\mathcal{A}}^{ECDLP}(t) \leq \epsilon$ for any sufficiently small $\epsilon > 0$.

Definition 3. *[Elliptic curve Computational Diffie Hellman Problem (ECCDH)]* Against three specified random points $Q,\ aQ,\ bQ \in E_p(a, b)$, compute another point abQ. The likelihood that a polynomial time (t) bound adversary $\mathcal{A}$ can calculate abQ is as given: $Adv_{\mathcal{A}}^{ECCDH}(t) = Prb[(\mathcal{A}(Q, aQ, bQ) = abQ \in E_p(a, b)]$. The ECCDH supposition infers that $Adv_{\mathcal{A}}^{ECCDH}(t) \leq \epsilon$ for any sufficiently small $\epsilon > 0$.

Definition 4. *[Bilinear Diffie Hellman Problem (BDHP)]* In symmetric pairing $(G = G_1 = G_2)$, against four specified random points $Q, aQ,\ bQ,\ cQ \in G$, compute $e(Q, Q)^{abc}$. The likelihood that a polynomial time (t) bound adversary $\mathcal{A}$ can calculate $e(Q, Q)^{abc}$ is as given: $Adv_{\mathcal{A}}^{BDHP}(t) = Prb[(\mathcal{A}(Q, aQ, bQ, cQ) = e(Q, Q)^{abc} \in G_3]$. The BDHP supposition infers that $Adv_{\mathcal{A}}^{BDHP}(t) \leq \epsilon$ for any sufficiently small $\epsilon > 0$.

Definition 5. *[Decisional Bilinear Diffie Hellman Problem (DBDHP)]* In symmetric pairing $(G = G_1 = G_2)$, against four specified random points $Q, aQ,\ bQ,\ cQ \in G$, compute $e(Q, Q)^{abc}$. The likelihood that a polynomial time (t) bound adversary $\mathcal{A}$ can verify if $e(Q, Q)^{abc} \stackrel{?}{=} e(Q, Q)^{d}$ is as given: $Adv_{\mathcal{A}}^{DBDHP}(t) = Prb[(\mathcal{A}(Q, aQ, bQ, cQ) = e(Q, Q)^{abc} \stackrel{?}{=} e(Q, Q)^{d}]$. The DBDHP supposition infers that $Adv_{\mathcal{A}}^{DBDHP}(t) \leq \epsilon$ for any sufficiently small $\epsilon > 0$.

2.2.6 Common Adversarial Model

Common adversarial model is considered in this thesis, as revealed in [25–27]. Where, subsequent considerations are taken up as per the competences of the adversary A:

1. A have control over entire public communication link. A is capable to interrupt, rerun, amend, eliminate or can transmit a new forged message.
2. A can excerpt information engraved in the smart card by conducting power analysis or leaked data [28, 29].
3. A can be stranger or can be a deceitful user or server of the system.
4. Registered users and server's identities are not private and known to insiders.

5. The server is considered to be protected and A cannot compromise server's private key.

2.2.7 BioHashing

The biometric is the unique and quantifiable characteristic commonly utilized to identify and designate or recognize a particular human. Biometric is practically utilized for authentication purpose and demands the physical presence of a particular person in order to be authenticated. At each imprint, biometric features (such as fingerprint, retina, face recognition and iris recognition etc.) may faintly differ from the actual one, leading towards frequent false rejections of legitimate users. Frequent false rejections of legitimate users in return degrade the performance of the latent system. Jin et al. [30] proposed a scheme to tenacity the problem of false rejection. Jin et al.'s scheme implements two factor authentication based on iterated inner product amid biometric characteristics and tokenized pseudo-random number. Moreover, in order to implement Jin et al.'s scheme multiple and explicit user codes are engendered and these explicit user codes are designated as BioHash codes. Recently, numerous biohashing schemes are being introduced [31, 32]. Bio Hashing is verified to be the most suitable and compatible technique that can be utilized in tiny smart devices such as smart card and smart phone etc.

2.3 Automated tool ProVerif

Formal security analysis for cryptographic protocols was initiated during mid 80's with varying techniques including algebraic, state space and logic methods. Applied pi calculus is one of the prevailing logic methods for formal analysis of cryptographic protocols. ProVerif makes use of applied π calculus to validate correctness and robustness of security protocols [33–35]. The analysis capability of ProVerif ranges from proving the trace properties like authentication, reachability and secrecy to ascertain whether or not a presented protocol extends to a bad state [36, 37] to the observational properties like anonymity and privacy [38, 39]. ProVerif protocol model consists of three parts. In declaration part, names and cryptographic primitives are stated, in process parts, the processes and subprocesses are defined, while core protocol steps are defined in main part. To analyze the security of our proposed schemes, we have adopted the formal validation model of ProVerif.

2.4 Chapter Summary

In this chapter, a brief discussion relating to mathematical background of the thesis along with computational hard problems, the common adversarial model, biohashing and introduction to ProVerif is solicited.

Chapter 3

A Two-factor ECC based Privacy Preserving Authentication Protocol for SIP

The session initiation protocol (SIP) has got much attractiveness during recent times, as it can achieve sessions including IP calls, multimedia distribution and conferences. SIP works on the standards of the hyper text transport protocol (HTTP), which is based on the request-response messages between client and server. Authentication is considered as a true vital facet for SIP, because the tangled participants must be validated even before the start of the session. In SIP, the client initiates the request message, while server asks for the legality of client by sending a challenge message, which also contains built-in server authentication information. The client after authenticating the server, sends a response message. The server validates the client by examining the response message. The SIP authentication makes use of password based authentication along with symmetric or public key cryptography methods. The former, however, is more cost efficient than later, but the later provides more security. So we need a trade off between the two. The first password based authentication scheme was proposed by Chang et al. [40]. Successively, a number of password based authentication schemes were proposed [34, 41–62]. In earlier password based schemes, the server needs to store a verifier table having an entry for each client. Such schemes were proved to be vulnerable to the stolen verifier attack, scalability issues and having high computational costs, because the server has to secure the verifier table from unauthorized access by internal as well as external attackers. Further, server has to create a distinct entry for each client, which limits the number of clients and need extra computation for storing and comparing verifier

table entries.

Recently, Zhang et al. [63] proposed an efficient authentication scheme, the scheme made an efficient use of elliptic curve cryptography. They introduced the notion of authentication without storing any verifier table on server. Further, they claimed their scheme to provide resistance to known attacks. But Irshad et al. [58], Zhang et al. [64] and Tu et al. [2] independently mentioned a number of weaknesses in Zhang et al.'s scheme [63]. Irshad et al. [58] claimed the scheme [63] to be vulnerable to replay and denial of services attack. Further, Irshad et al. [58] proposed an improved single round scheme, but their scheme was vulnerable to impersonation attack as mentioned by Arshad and Nikooghadam [42], they also proposed an improved scheme. Unfortunately Arshad and Nikooghadam's scheme [42] once again introduced the verification tables on the server side as well as having no provision for user's anonymity. Zhang et al. [64] also proposed an improved scheme of [63], but their improved scheme was proved to be vulnerable to the server impersonation attack by Farash [65]. Farash [65] then proposed an improved scheme, the scheme of Farash [65] once again does not provide user anonymity and is vulnerable to replay and denial of services attacks.

In 2014, Tu et al. [2] also proposed an improved scheme to improve the security of Zhang et al.'s scheme [63] and claimed it to be secure. However, recently Farash [3] mentioned that Tu et al.'s scheme is vulnerable to server impersonation attack. Then Farash [3] proposed an improvement of Tu et al.'s scheme. Here, we show that Tu et al.'s scheme [2] is vulnerable to server impersonation, replay and denial of services attacks as well as lacking user anonymity. Furthermore, we analyze that Farash's improvement [3] on Tu et al.'s scheme [2] is lacking user anonymity and is vulnerable to replay attack. Then an anonymous authenticated key agreement is proposed which is more secure and suitable for all lightweight environments. The rest of the chapter is organized as follows: In section 3.1, a brief discussion relating to SIP architecture and SIP authentication procedure is performed. Section 3.2 reviews Tu et al.'s scheme [2] followed by Farash's improvement [3] , while cryptanalysis of Tu et al.'s and Farash's schemes are presented in section 3.3. Section 5.3 describes our improved authentication scheme for SIP. In section 3.5, we have proved the security of the proposed scheme in the random oracle model. We have also performed automatic security validation using automated tool ProVerif in same section. Section 3.6 presents the performance analysis of improved authentication scheme. Finally, chapter's summary is solicited in section 3.7.

3.1 SIP Architecture

SIP is based on the request-response messages between client and server like HTTP. During SIP based authentication, a uniform resource identifier (URI) is used to identify users. The SIP design is compromising a number of contributors, including a client agent, redirect, proxy, registration and location servers. The client agent works as a terminal, the proxy server acts as an arbitrator amid the client and server, the caller location is notified by redirect server, while register server posts his new location to location server.

3.1.1 SIP authentication procedure

To get SIP services, a client initiates registration process with a proxy server, the registration process includes a message from a client containing his secret information like his identity/user name and password using some secure channel. After registration, the client is allowed to login with a proxy server using pre-shared secrets on some public channel. Then the SIP session procedure is performed to locate another SIP client to establish a session. The login/authentication procedure involves exchange of following messages among client and proxy server:

1: Client → Server: REQUEST
A connection request is sent to server by client.

2: Server → Client: CHALLENGE (nonce, realm, info)
For the received request, the server sends a challenge message to the client. The challenge message must contain some random nonce and realm, further it must also have some built in information to verify the legality of the server.

3: Client → Server: RESPONSE (nonce, realm, username, info)
The client after receiving a challenge message, first verifies sender's legality then it spawns a response message.

4: For the received response message, the server using some pre-shared information verifies the client's legality. If client is not proven to be legal, the session is terminated by the server. Otherwise, a unique session key is established between the both.

Table 3.1: Notation guide

Notations	Description	Notations	Description
n, p	Two large prime numbers	F_p	The finite prime field
$E_p(a, b)$	Elliptic Curve over F_p	G	Additive group of points over $E_p(a, b)$
P	Generator of G	PW_i	i^{th} client password
d_S	Server Private Key	$K_S = d_S P$	Server Public Key
$\|\|$	Concatenation operator	$\oplus$	XOR operation
$h(.), h_1(.)h_2(.)$	Three One way hash Functions	$\mathscr{U}$	The legal Client
$\mathscr{S}$	The legal Server	$\mathscr{A}$	The Adversary

3.2 Tu et al.'s Scheme and Farash's Improvement

This section reviews Tu et al.'s [2] SIP authentication scheme using ECC and it's improvement proposed by Farash [3]. Tu et al.'s scheme consists of four phases: system initialization phase, registration phase, mutual authentication with key exchange phase and password changing phase. The notation guide for chapter is described in table 3.1.

3.2.1 System Initialization Phase

At start Server $\mathscr{S}$ selects an elliptic curve $E_p(a, b)$, then a point P as base point over selected curve. $\mathscr{S}$ chooses three one way hash functions. Then $\mathscr{S}$ selects a random private key $d_S \in Z_n^*$ and calculates public key $K_S = d_S P$. Finally $\mathscr{S}$ publishes $\{E_p(a, b), P, K_S, h(.), h_1(.), h_2(.)\}$ and keeps d_S secret.

3.2.2 Registration Phase

Registration phase consists of two steps. Firstly, the client $\mathscr{U}$ chooses a password PW_i, selects a random integer $a \in Z_n^*$. Then $\mathscr{U}$ computes $h(PW_i||a)$ and sends $h(PW_i||a)$, $username$ to $\mathscr{S}$ via some secure channel. When server $\mathscr{S}$ receives $h(PW_i||a)$ and $username$, $\mathscr{S}$ computes $R = (h(PW_i||a) + h(username||d_S))P$. Then $\mathscr{S}$ stores R in smart card, and delivers the smart card to $\mathscr{U}$ through any secure channel. After receiving R, $\mathscr{U}$ stores a in the smart card. Now smart card contains (R, a).

3.2.3 Mutual Authentication and Key Exchange Phase

Step 1: The client $\mathscr{U}$ initiates authentication process by inserting his smart card in reader and entering the password PW_i. The smart card generates a random number $b \in Z_n^*$,

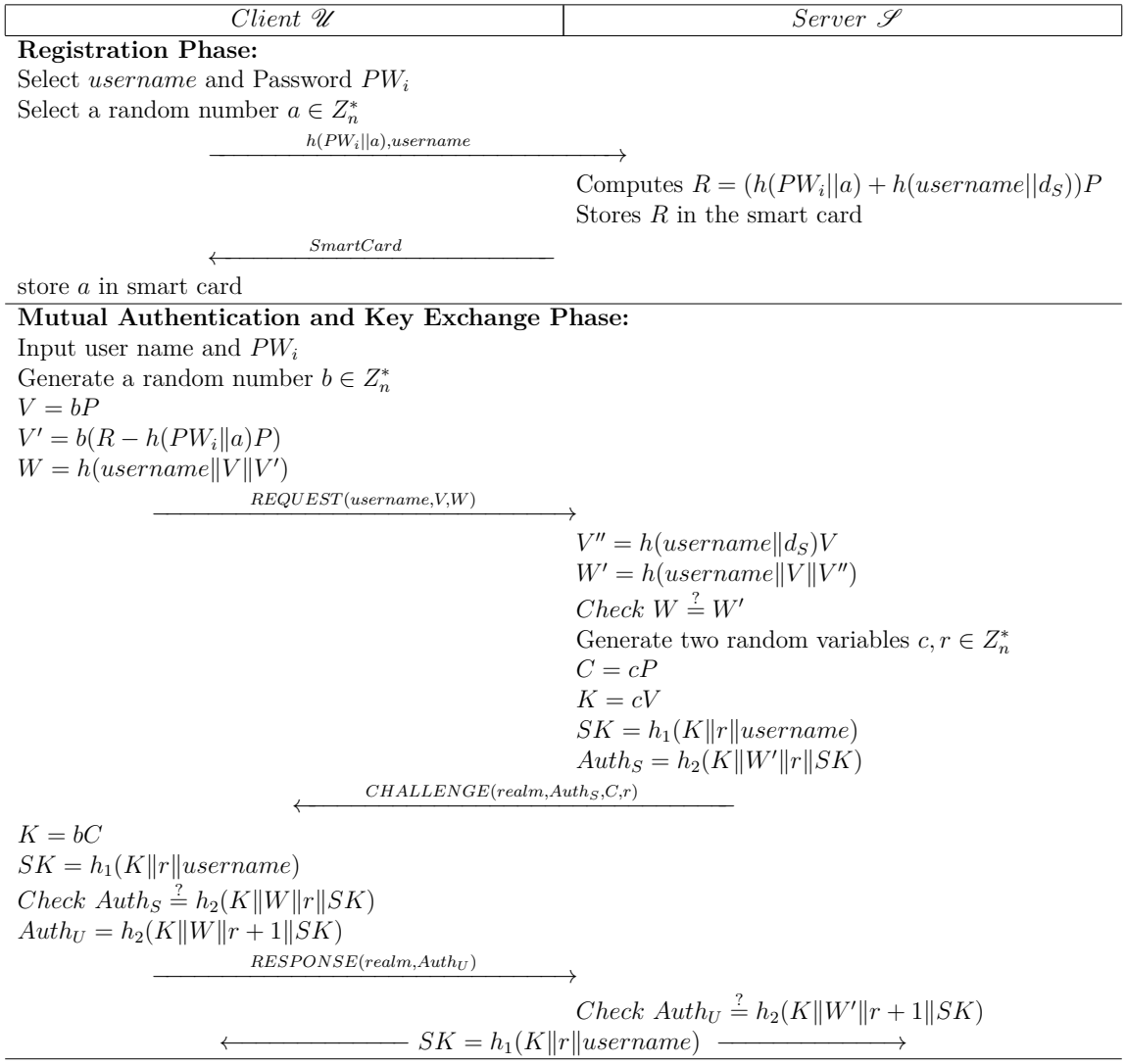


Figure 3.1: Tu et al.'s scheme

then computes $V = bP$, $V' = b(R - h(PW_i||a)P)$ and $W = h(username||V||V')$. Further $\mathscr{U}$ requests authentication by sending $username$, $V\&$ W in a request message to $\mathscr{S}$.

Step 2: After receiving the request $\mathscr{S}$ calculates $V'' = h\,(username||d_S)\,V$ and $W' = h(username||V||V'')$. $\mathscr{S}$ verifies $W \stackrel{?}{=} W'$, if not true $\mathscr{S}$ aborts the session. Otherwise, $\mathscr{S}$ chooses two random numbers $c, r \in Z_n^*$, and calculates $C = cP$,$K = cV$. Then, $\mathscr{S}$ computes the shared key $SK = h_1(K||r||username)$, and $Auth_S = h_2(K||W||r||SK)$. Finally, it sends challenge message with $(realm, Auth_S, C, r)$ to client via public channel.

Step 3: $\mathscr{U}$ computes $K = bC$ and $SK = h_1(K||r||username)$ upon receiving the challenge message from $\mathscr{S}$. $\mathscr{U}$ further verifies $Auth_S \stackrel{?}{=} h_2(K||W||r||SK)$, if the relationship

proves to be false, the session is aborted by $\mathscr{U}$. Otherwise, $\mathscr{U}$ computes $Auth_U = h_2(K||W||r+1||SK)$, it further sends the response message $(realm, Auth_U)$ to $\mathscr{S}$. $\mathscr{U}$ keeps SK as shared key with $\mathscr{S}$.

Step 4: When $\mathscr{S}$ receives the response message, first checks $h_2(K||W||r+1||SK) \stackrel{?}{=} Auth_U$, if relationship does not exist, the session is aborted by $\mathscr{S}$. Otherwise, $\mathscr{S}$ stores session key SK.

3.2.4 Password Change Phase

A password change request is initiated after generation of a session key. Following steps are performed between $\mathscr{U}$ and $\mathscr{S}$ for successful password update.

Step 1: $\mathscr{U}$ selects a new password PW_n and two random numbers $a_n, N_n \in Z_n^*$, then $\mathscr{U}$ computes, $C_u = E_{SK}(username\|N_n\|h(PW_n\|a_n)\|h(username\|N_n\|h(PW_n\|a_n)))$. Finally, $\mathscr{U}$ sends password change request $\{C_u, N_n\}$ to $\mathscr{S}$.

Step 2: For the received password change request $\{C_u, N_n\}$. $\mathscr{S}$ first decrypts C_u, then checks the validity of message tag $h(username\|N_n\|h(PW_n\|a_n))$. If it is valid $\mathscr{S}$ computes $R_n = (h(PW_n||a_n)+h(username||d_S))P$ and $C_S = E_{SK}(R_n\|h(username\|N_n+1\|R_n))$. Finally, $\mathscr{S}$ sends C_S to $\mathscr{U}$.

Step 3: Upon receiving C_S, $\mathscr{U}$ decrypts it and verifies the tag $h(username\|N_n+1\|R_n)$, if it is valid. $\mathscr{U}$ stores R_n and a_n in smart card.

3.2.5 Farash's Improvement

This subsection reviews Farsh's improvement on Tu et al.'s scheme. Farash slightly modified the authentication phase of Tu et al.'s scheme. Farash's modification is an alternation in the computation of $Auth_S$ shown as follows:

$$Auth_S = h_2(K||V''||r||SK)$$

While there is no change in system initialization, registration and password change phases.

3.3 Cryptanalysis of Tu et al.'s Scheme and Farash's Improvement

This section shows that an adversary can easily launch impersonation attack on Tu et al.'s scheme. We show that the adversary can easily masquerade as a legitimate server to share a session key. Further, we show that Tu et al.'s scheme and Farash's improvement both are lacking the user's anonymity and are vulnerable to replay and denial of services attacks.

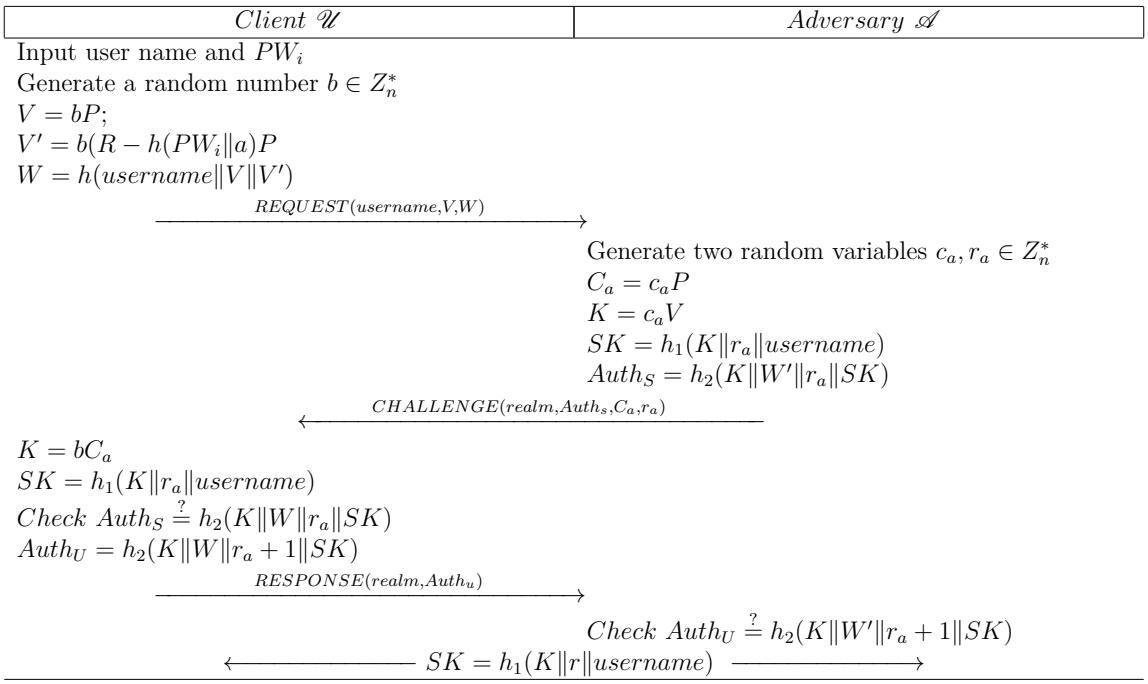


Figure 3.2: Server impersonation attack on Tu et al.'s scheme

3.3.1 Weaknesses of Tu et al.'s scheme

Following subsections shows that Tu et al.'s scheme is vulnerable to the server impersonation attack as well as lacking the user's anonymity. It is also shown that Tu et al.'s scheme can not resist replay and denial of services attack.

3.3.1.1 Server Impersonation Attack

By impersonation attack, an active adversary $\mathscr{A}$ can easily badge itself as a legal server without knowing the private key of the server. The adversary $\mathscr{A}$ do the following steps in order to masquerade the legal server $\mathscr{S}$ to share the session key with the client $\mathscr{U}$.

Step 1: Initially, when a legal client $\mathscr{U}$ sends $REQUEST\,(username, V, W)$ to the server $\mathscr{S}$, the attacker $\mathscr{A}$ intercept the message and selects two random numbers $c_a, r_a \in Z_n^*$. $\mathscr{A}$ Further, calculates $C_a = c_a P$, $K = c_a V$, $SK = h_1(K||r||username)$ and $Auth_S = h_2(K||W'||r||SK)$.

Step 2: $\mathscr{A}$ sends $CHALLENGE(realm, Auth_S, C_a, r_a)$ to $\mathscr{U}$.

Step 3: Upon receiving the message $\mathscr{U}$ calculates $K = bC$ and $SK = h_1(K||r||username)$, then $\mathscr{U}$ checks $Auth_S \stackrel{?}{=} h_2(K||W||r||SK)$, it is obvious that $Auth_S$ hold. $\mathscr{U}$ further computes $Auth_U = h_2(K||W||r+1||SK)$

Step 4: $\mathscr{U}$ sends $RESPONSE(realm, Auth_U)$ to $\mathscr{S}$.

Step 5: $\mathscr{A}$ intercepts the response message, the shared key between $\mathscr{U}$ and $\mathscr{A}$ is $SK = h_1\,(K||r||username)$.

Therefore, $\mathscr{A}$ successfully launched server impersonation attack and exchanged the session key $SK = h_1(K||r||username)$ with legal user $\mathscr{U}$.

3.3.1.2 No Provision for User Anonymity

Along with traditional security, user anonymity and privacy has emerged as an extremely important factor to be considered. Without privacy and anonymity, user's sensitive personal information can be accessed by an adversary by just analyzing the session's information. Specially in mobile communication, the attacker may become able to identify $\mathscr{U}$'s login history, his movement patterns, current location and so on. Furthermore, such sensitive information may be misused by the adversary. Tu et al.'s scheme did not consider these loopholes, hence lacking user anonymity.

3.3.1.3 Replay and Denial of Services Attacks

In Tu et al.'s scheme, an active attacker $\mathscr{A}$ after intercepting a login request $REQUEST(username, V, W)$ can replay it later on, because the request does not contain any time stamp. Off course $\mathscr{A}$ will not be able to stake the session key because such replay will be fixed in response message $RESPONSE(realm, Auth_U)$ by the attacker, but such attack can hoax $\mathscr{S}$ and $\mathscr{U}$ to perform step 2 and 3 of authentication phase, resulting into a counterfeit utilization of computation power as well as communication and storage resources. A simultaneous execution of a large number of such attacks can even lead to denial of services, causing access prevention to the legal client.

3.3.2 Weaknesses of Farash's scheme

Following subsections shows Farash's scheme is lacking the user's anonymity and is vulnerable to replay and denial of services attack.

3.3.2.1 No Provision for User Anonymity

Farash presented an improvement of Tu et al.'s scheme. Unfortunately in his improvement, Farash did not consider the importance of user's anonymity and just change the computation of $Auth_S$, while *username* is sent in plaintext to the server. Therefore, Farash's improvement is also lacking user anonymity, which can cause serious threats as discussed earlier in subsection 3.3.1.2.

3.3.2.2 Replay and Denial of Services Attack

Similar to Tu et al.'s scheme, in Farash's scheme an active attacker $\mathscr{A}$ after intercepting a login request $REQUEST(username, V, W)$ can replay it later on, forcing $\mathscr{S}$ to process the request and send the challenge message to $\mathscr{U}$, because the request does not contain any time stamp. Which may not only burdens the system, but can also cause denial of services to legitimate client.

3.4 Proposed Scheme

The security breaches of Tu et al.'s and Farash's schemes are due to the fact that security of their schemes rely on public parameters V, W and *username* transmitted on an insecure channel. In Tu et al.'s scheme V and W are also involved in the computation of SK and $Auth_S$. So an adversary can easily generate SK and $Auth_S$ in order to masquerade itself as the legal server. Similarly, the absence of the time stamp in both Tu et al.'s and Farash's schemes resulted in burdening the system and replay as well as denial of service attacks. Therefore, in improved scheme the transmission of W and *username* is replaced by $\overline{W}$ and $\overline{username}$ to provide the user's anonymity and resistance to impersonation and replay attacks. We have amended only registration and mutual authenticated key exchange phases, the proposed scheme works as follows:

3.4.1 Registration Phase

Registration phase consists of two steps firstly, client $\mathscr{U}$ chooses a password PW_i, selects a random integer $a \in Z_n^*$. Then $\mathscr{U}$ computes $h(PW_i\|a)$, and sends $h(PW_i\|a)$, $username$ to $\mathscr{S}$ via some secure channel. Upon reception of registration request message $h(PW_i\|a), username$, the server $\mathscr{S}$ selects random $r \in Z_n^*$ and computes $\overline{username} = Enc_{d_S}(username\|r)$, $R = (h(PW_i\|a) + h(username\|d_S))P$. Further, $\mathscr{S}$ stores R and $\overline{username}$ in the smart card, and delivers the smart card to $\mathscr{U}$ through any secure channel. After receiving the smart card, $\mathscr{U}$ stores a in it. Finally, the smart card contains $(R, \overline{username}, a)$.

3.4.2 Mutual Authentication and Key Exchange Phase

Step 1: $\mathscr{U} \rightarrow \mathscr{S}$: $\{\overline{username}, V, \overline{W}, t_i\}$

The client $\mathscr{U}$ initiates the authentication process by inserting his smart card (SC) in the reader and entering the password PW_i. SC then generates a random number $b \in Z_n^*$, and computes:

$$V = bP \tag{3.1}$$

$$V' = b(R - h(PW_i||a)P) \tag{3.2}$$

$$W = h(username||V||V') \tag{3.3}$$

$$\overline{W} = h_1(W \oplus V \oplus t_i) \tag{3.4}$$

Where t_i is freshly generated time stamp. Further, $\mathscr{U}$ requests authentication by sending $\overline{username}$, V and $\overline{W}, t_i$ in request message to $\mathscr{S}$.

Step 2: $\mathscr{S} \rightarrow \mathscr{U}$:$\{realm, Auth_S, C, r, Z\}$

After receiving the request, $\mathscr{S}$, generates a new time stamp t_s and compares it with received t_i. If the difference between both is within a threshold time period Δ. $\mathscr{S}$ considers the time stamp as fresh and proceeds with the login request. Otherwise, $\mathscr{S}$ aborts the session. For valid time stamp, $\mathscr{S}$ proceeds with login request as follows:

$$username\|r = Dec_{d_S}(\overline{username}) \tag{3.5}$$

$$V'' = h(username||d_S)V \tag{3.6}$$

$$W' = h(username||V||V'') \tag{3.7}$$

Further, $\mathscr{S}$ verifies $W \stackrel{?}{=} h_1(W' \oplus V \oplus t_i)$, if is not true, $\mathscr{S}$ aborts the session. Otherwise,

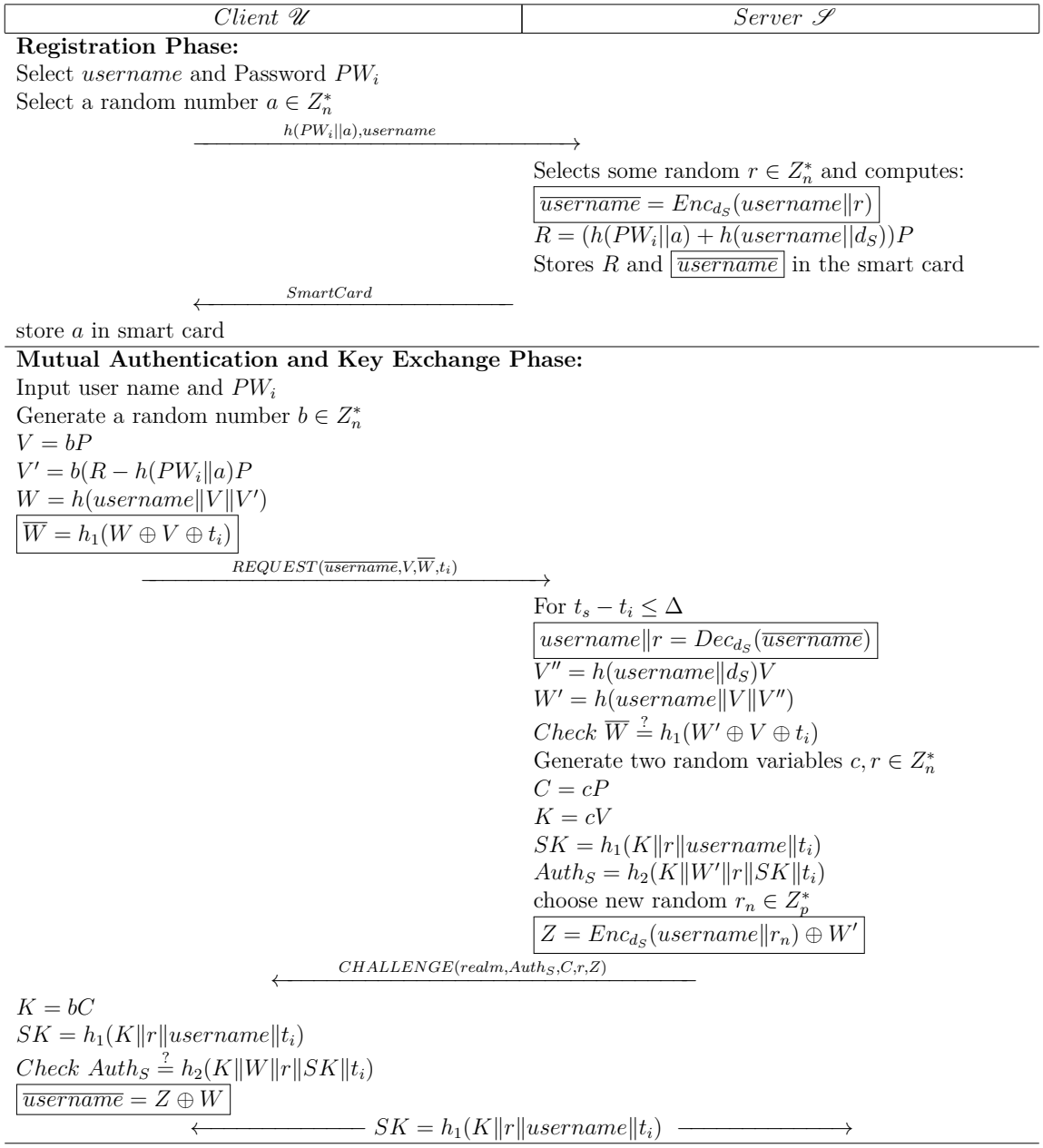


Figure 3.3: Proposed Scheme

$\mathscr{S}$ chooses three random numbers $c, r, r_n \in Z_n^*$ and computes:

$$C = cP \tag{3.8}$$

$$K = cV \tag{3.9}$$

$$SK = h_1(K||r||username\|t_i) \tag{3.10}$$

$$Auth_S = h_2(K\|W'\|r\|SK\|t_i) \tag{3.11}$$

$$Z = Enc_{d_S}(username\|r_n) \oplus W' \tag{3.12}$$

Finally, $\mathscr{S}$ sends $\{realm, Auth_S, C, r, Z\}$ to client via public channel.

Step 3: $\mathscr{U}$ computes $K = bC$ and session key $SK = h_1(K\|r\|username\|t_i)$ upon receiving the challenge message from server and it verifies $Auth_S \stackrel{?}{=} h_2(K\|W||r||SK\|t_i)$, if the relationship proves to be false, the session is aborted by $\mathscr{U}$. Otherwise, $\mathscr{U}$ replaces $\overline{username} = Z \oplus W$. Finally, SK is set as shared key with $\mathscr{S}$.

3.5 Security Analysis

This section analyzes the security of proposed scheme, the scheme provides mutual authentication, resist user and server impersonation attacks and is secure against stolen verifier, man-in-middle and offline password guessing attack. The scheme also provides perfect forward secrecy. We have proved the security of proposed scheme in the random oracle model as well by using automated tool ProVerif. Further, we have also performed informal security comparisons with existing schemes.

3.5.1 Provable Security Model

To analyze the security of the proposed scheme, we have adopted the formal security model introduced in [66, 67].

3.5.1.1 Security model

There are two participants in the proposed authentication protocol $\mathcal{P}$: a client $\mathscr{U}$ and a server $\mathscr{S}$. During execution of $\mathcal{P}$, there may be several instances of each participant, where each instance is linked with a number z and is termed as an oracle, jumbled in a divergent execution of $\mathcal{P}$. We outline U^x as the x^{th} instance of $\mathscr{U}$, similarly S^y is outlined as y^{th} instance of $\mathscr{S}$, we also term I^z for both the instances U^x and S^y with eradication of differences. There can be three possible outcomes of an oracle, accept, reject or $\perp$. An oracle ranges to an accept form, if it receives a righteous message. The wrong message leads to reject form, while $\perp$ state appears if no decision is made or no result returned.

Even before execution of $\mathcal{P}$, $\mathscr{U}$ owns a $username$, PW_i, while the smart card SC contains $R, \overline{username}, a$. $\mathscr{S}$ is having a private and public key pair d_S and $K_S = d_S P$. There are finite number of passwords, while the password dictionary $\mathcal{D}$ is of size $|\mathcal{D}|$. $\mathscr{S}$ is assumed to be secure.

According to adversary capabilities, the attacker $\mathscr{A}$ is having full control over public communication channel. $\mathscr{A}$ can initiate and arbitrate the session between $\mathscr{U}$ and $\mathscr{S}$. $\mathscr{A}$ aims to violate communication privacy and session key secrecy. $\mathscr{A}$ can make a number of queries in the oracles and may get replies. The list of such queries is itemized below:

- $h(s/s1/s2, rec)$: It is a hash oracle and it results into some arbitrary value r. Employment of this query builds a record (rec, r), depending upon the first parameter, it generates three different hash lists h_{slist}, h_{s1list} and h_{s2list}. Dealing of these records is in proof process.
- $Send(U^x/S^y, msg/SCLD)$: This query replicates the active attack on communication, it yields the message that U^x or S^y generates upon reception of message msg, if second argument of $Send$ query is $SCLD$, the output is the message $\{\overline{username}, V, \overline{W}, t_i\}$ in step 1 of authentication phase. The query normally finishes as the steps in mutual authentication phase of $\mathcal{P}$.
- $Execute(U^x, S^y)$: This query enables the attacker to perform a passive attack on the communication channel. By simulating $Execute$, $\mathscr{A}$ can access the messages exchanged over insecure communication channel between U^x or S^y.
- $Reveal(I^x)$: This query designates the known session key attack. By this query, $\mathscr{A}$ can acquire the computed session key between U^x and S^y.
- $Corrupt(SC)$: This query enables $\mathscr{A}$ to obtain all the parameters stored in the smart card (SC).
- $Test(I^z)$: This query stands for obtaining the session key. The simulation of $Test$ query results into $\perp$, if I^z does not generate a session key. Otherwise, it outputs into flipping of a coin Ω. If $\Omega = 1$, $Test$ query outputs the existent session key, if $\Omega = 0$ uniform random string is returned, whose length is same as the actual session key. $\mathscr{A}$ is allowed to ask $Test$ query only once to the $fresh$ oracle.

Following are some definitions used to prove the security of proposed scheme.

- $Partnering$: Each participating instance U^x, S^y is having a partner identity pid_U^x or pid_S^y along with a session key sk_U^x or sk_S^y, an identifier sid_U^x or sid_S^y, which is accepted and agrees a session key. U^x and S^y are termed as partners if and only if $sid_U^x = sid_S^y$, $pid_S^y = U^x$, $pid_U^x = S^y$ and $sk_U^x = sk_S^y$.
- $fresh$: Any instance I^z is believed as $fresh$, if no $Reveal$ query happened on I^z.
- $PAP - security$: The advantage for $\mathscr{A}$ to break the security of $\mathcal{P}$ is defined as the

probability that can acceptably guess the result of flipping of coin Ω by $Test(I^z)$, where I^z is $fresh$ as well as accepted. Let $\mathscr{A}$ outputs $\Omega^{'}$, the advantage is as follows:

$$Adv_{\mathcal{P}}^{PAP}(\mathscr{A}) = |2Pr[\Omega = \Omega^{'}] - 1| \tag{3.13}$$

The proposed authentication protocol is designated as $PAP - secure$ if $Adv_{\mathcal{P}}^{PAP}(\mathscr{A})$ is negligible.

- We define the Elliptic curve computational Diffie-Hellman (ECCDH) assumption as follows: Given three point $\alpha P, \beta P$ and P over an elliptic curve $E_p(a, b)$, where $\alpha, \beta \in Z_n^*$, the probability $\mathscr{A}$ can compute $\alpha\beta P$ in polynomial time t can be defined as $Adv_{\mathscr{A}}^{ECCDH}(t)$. The ECCDH assumption implies that $Adv_{\mathscr{A}}^{ECCDH}(t) \leq \epsilon$.

3.5.1.2 Security proof

Theorem 1. *The password engaged by $\mathscr{U}$ is from a password dictionary $\mathcal{D}$ having size $|\mathcal{D}|$. Let l_{hs} be the length of hash value, $\mathcal{P}$ is the proposed authentication protocol. An adversary $\mathscr{A}$ during polynomial time t can make maximum q_{snd} Send queries, q_{exe} Execute queries and q_{hs}, q_{hs1}, q_{hs2} hash queries. $\mathscr{A}$'s advantage is as follows:*

$$\begin{aligned} Adv_{\mathscr{P}}^{PAP}(\mathscr{A}) \leq & \frac{q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{2^{l_{hs}}} + \frac{(q_{snd} + q_{exe})^2}{2(p-1)} \\ & + 2q_{exe} \cdot Adv_{\mathscr{A}}^{ECCDH}(\overline{W}) + 2\max\{\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\} \end{aligned} \tag{3.14}$$

Proof. For proof, we mark a sequence of games ranging from G_0 to G_4, the event $Succ_i$ means that $\mathscr{A}$ correctly gausses Ω during G_i effectively in $Test$. As per the requirements for our model, there is no need for $\mathscr{A}$ to compute identity of the client because there is only one user. The games for our proof are listed below:

- Game G_0: It is the real protocol in random oracle model. Here, we selected random coin flipped value $\Omega^{'}$. We realize that $\mathscr{A}$'s advantage to guess Ω correctly is as follows:

$$Adv_{\mathcal{P}}^{PAP}(\mathscr{A}) = 2Pr[Succ_0] - 1 \tag{3.15}$$

- Game G_1: We have simulated all oracles for the queries. Also, three lists are used to

store the record (rec, r) formed after query mentioned in the security model. h_{slist}, h_{s1list} and h_{s2list} are used to store answers to h oracle. On hash query, if there exists a record (rec, r) in corresponding hash list, r is returned, otherwise a random value r' is returned to $\mathscr{A}$ and a record is added to corresponding hash list against r'. When h oracle is queried by $\mathscr{A}$ then the record in h_{Alist}. From $\mathscr{A}$'s view point G_0 and G_1 are not distinguishable through the simulation, so:

$$Pr[Succ_1] = Pr[Succ_0] \tag{3.16}$$

- Game G_2: Some of the collisions are avoided during G_2, which is aborted when some collisions ensued on transcripts (V, C) and on hash values. As $b, c \in [1, p-1]$ and the length of each hash value is l_{hs}. Referring the birthday paradox, the the maximum collision probability in result of hash oracles are $q_{hs}^2/2^{l_{hs}+1}$, $q_{hs1}^2/2^{l_{hs}+1}$ and $q_{hs2}^2/2^{l_{hs}}$. Similarly, the maximum collision probability in the transcripts is $(q_{snd} + q_{exe})^2/2(p-1)$. So we have:

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{2^{l_{hs}+1}} + \frac{(q_{snd} + q_{exe})^2}{2(p-1)}. \tag{3.17}$$

- Game G_3: This game is aborted, if $\mathscr{A}$ computes correct messages without hash oracles, the game is divided into two cases according to two messages:

  1. To forge $Send(S^y, (\overline{username}, V, \overline{W}, t_i))$ query, $\mathscr{A}$ must make $(W \oplus V \oplus t_i)$ and V' queries, Or we can say that $(W \oplus V \oplus t_i) \in h_{Alist}$ should be true. If we have not found it as a role of server, the probability is up to $\frac{q_{snd}}{2^{l_{hs}}}$. Note that $\mathscr{S}$ does not know pw_u, so the record $(username_u||pw_u||a_u, *)$ can not be checked. The probability is $\frac{q_{hs}}{2^{l_{hs}}}$.
  2. To forge $Send(U^i, (realm, Auth_S, C, r, Z))$, A must make $(K\|W'\|r\|SK\|t_i)$. The probabilities are upper bounded by $\frac{q_{hs1}}{2^{l_{hs}}}$ and $\frac{q_{snd}}{2^{l_{hs}}}$ respectively for the matter that the two records do not exist in h_{Alist} .

  Hence, games G_3 and G_2 are indistinguishable unless the messages are forged without

hash queries. So we have

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{2q_{snd} + 2q_{hs1}}{2^{l_{hs}}} \tag{3.18}$$

- Game G_4: For this game, ECCDH is brought in, $\mathcal{A}$ is allowed to make oracles normally. $\mathscr{A}$ can acquire session key SK, if he wins this game. To win this game, $\mathscr{A}$ has to solve ECCDH. To compute SK, $\mathcal{A}$ must ask $(kP\|r\|username_u)$ query. If this record exists in the list h_{Alist}, $\mathscr{A}$ breaks ECCDH problem. The difference between the game G_4 and the game G_3 is as follows:

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq q_{exe} \cdot Adv_{\mathscr{A}}^{ECCDH}(\overline{W}). \tag{3.19}$$

There are two possible cases where the adversary distinguishes the real session key SK and the random key as follows:

Case 1. The adversary queries $(K, r, username)$ to h_{s1}. The probability that this event occurs is $\frac{q_{hs1}}{2^{l_{hs}}}$.

Case 2. The adversary asks $Send(U^x)$ query and successfully impersonates U to S. The adversary is not allowed to reveal static key PW_i of $\mathscr{U}$. Thus, in order to impersonate $\mathscr{U}$, the adversary has to obtain some information of the password PW_i of $\mathscr{U}$. The probability is $1/|D|$. Since, there are at most q_{snd} sessions of this kind, the probability that this event occurs is lower than $q_{snd}/|D|$

As a conclusion:

$$\Pr[Succ_4] = \frac{1}{2} + \max\{\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\}. \tag{3.20}$$

Combining the equations Eqs. (3.15), (3.16), (3.17), (3.18), (3.19) and (3.20), the announced result is as follows:

$$
\begin{aligned}
Adv_{\mathscr{P}}^{PAP}(\mathscr{A}) &= \Pr[Succ_0] - 1| \\
&= 2|\Pr[Succ_0] - \Pr[Succ_4] + \max\{\frac{q_{h1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\}| \\
&\leq 2(|\Pr[Succ_0] - \Pr[Succ_4]| + \max\{\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\}) \\
&\leq 2(|\Pr[Succ_1] - \Pr[Succ_2]| + |\Pr[Succ_3] - \\
&\Pr[Succ_4]| + \max\{\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\}) \\
&\leq \frac{q_{hs}^2 + q_{hs1}^2 + q_{hs2}^2}{2^{l_{hs}}} + \frac{(q_{snd} + q_{exe})^2}{2(p-1)} + \\
&2q_{exe} \cdot Adv_{\mathscr{A}}^{ECCDH}(\overline{W}) + 2\max\{\frac{q_{hs1}}{2^{l_{hs}}}, \frac{q_{snd}}{|D|}\}.
\end{aligned}
$$

□

3.5.2 Automated Security Verification

In this subsection, we have performed the automated security analysis of the proposed scheme using the widespread automated tool ProVerif [68]. In-order to prove the security of the proposed scheme, we have imprinted the steps as mentioned in section 5.3 and shown in Fig. 3.3. Then we check the secrecy of the session key and the reachability property as shown in Fig. 3.4. Finally, we got the results as follows:

1. inj-event(end_Server(id)) ==> inj-event(begin_Server(id)) is true.
2. inj-event(end_User(id_1780)) ==> inj-event(begin_User(id_1780)) is true.
3. not attacker(SK[]) is true.

The results (1) and (2) verify that both server and user processes started and terminated successfully, while (3) verifies that SK (session key) is not revealed to adversary and secrecy is maintained.

3.5.3 Further Security Discussion

This subsection analyzes the security of proposed scheme. The analysis verifies that proposed scheme resists all known attacks, while ensuring the user's anonymity and untraceability.

```
(*************** Channels **************)
free CH1_Sec:channel [private].
free CH2_Pub:channel.
(********** Names & Variables **********)
free username:bitstring.
free PWi:bitstring [private].
const Ks:bitstring.
const ds:bitstring [private].
const p:bitstring.
const q:bitstring.
const P:bitstring.
(**Constructors*destructors*Equations **)
fun H(bitstring):bitstring.
fun H1(bitstring):bitstring.
fun H2(bitstring):bitstring.
fun concat(bitstring,bitstring):bitstring.
fun add(bitstring,bitstring):bitstring.
fun ExcOR(bitstring,bitstring):bitstring.
fun multi(bitstring,bitstring):bitstring.
fun ECMP(bitstring,bitstring):bitstring.
fun subtract(bitstring,bitstring):bitstring.
fun syme(bitstring,bitstring):bitstring.
reduc forall m:bitstring,key:bitstring;
symd(syme(m,key),key)=m.
equation forall a:bitstring,b:bitstring;
ExcOR(ExcOR(a,b),b)=a.
event begin_User(bitstring).
event end_User(bitstring).
event begin_Server(bitstring).
event end_Server(bitstring).
```

(a) Declarations

```
(********Process Replication**********)
process ((pClientU)|!(pServerS))
(************** *queries* **************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj event(end_User(id)) ==>
    inj event(begin_User(id)) .
query id:bitstring; inj event(end_Server(id))
    ==>inj event(begin_Server(id)) .
```

(c) Main

```
(************User Process *************)
let pClientU=
(*Registration*)
new a:bitstring;
let HUPa = H(concat(PWi,a)) in
out(CH1_Sec,(HUPa,username));
in(CH1_Sec,(xR:bitstring,x_username:bitstring))
    ;
(*Login and Mutual Authentication*)
event begin_User(username);
new b:bitstring;
let V=ECMP(b,P) in
let V'=multi(b,subtract(xR,ECMP(H(concat(PWi,a)
    ),P))) in
let W=H(concat(username,(V,V'))) in
new ti: bitstring;
let x_W=H1(ExcOR(W,(V,ti))) in
out(CH2_Pub,(x_username,V,x_W,ti));
in(CH2_Pub,(xAuths:bitstring,xC:bitstring,xr:
    bitstring, xZ:bitstring));
let K=multi(b,xC) in
let SK=H1(concat(K,(xr,username,ti))) in
if(xAuths= H2(concat(K,(W,xr,SK,ti))) ) then
let X_username=ExcOR(xZ,W) in
event end_User(username)
else 0.
(********** Server Process *************)
let pServerS=
(*Registration*)
in(CH1_Sec,(xHUPa:bitstring,xusername:bitstring
    ));
new r:bitstring;
let X_username = syme(concat(username,r),ds) in
let R = ECMP(add(xHUPa,H(concat(xusername,ds)))
    ,P) in
out(CH1_Sec,(R,X_username));
(*      Login & Mutual Authentication    *)
in(CH2_Pub,(x_username:bitstring,xV:bitstring,
    x_W:bitstring,xti:bitstring));
event begin_Server(ds);
let (xxusername:bitstring,xr:bitstring) =symd(
    x_username,ds) in
let V'' = multi(H(concat(xxusername,ds)),xV) in
let W'=H(concat(username,(xV,V'')))in
if(x_W=H1(ExcOR(W',(xV,xti)))) then
new c:bitstring;
new r1:bitstring;
let C = ECMP (c,P) in
let K = multi (c,xV) in
let SK = H1(concat(K,(r,xxusername,xti))) in
let Auths = H2(concat(K,(W',r,SK,xti)))in
new rn:bitstring;
let Z=syme(ExcOR(concat(xxusername,rn),W'),ds)
    in
out(CH2_Pub,(Auths,C,r,Z));
event end_Server(ds)
else 0.
```

(b) Processes

Figure 3.4: ProVerif Validation

Table 3.2: Security Comparisons

Schemes → Security Properties ↓	Our	[3]	[2]	[63]	[42]	[58]	[64]	[65]
Resists insider attack	✓	✓	✓	✓	✓	✓	✓	✓
Resists offline guessing attack	✓	✓	✓	✓	✓	✓	✓	✓
Resists user impersonation attack	✓	✓	✓	✗	✓	✗	✓	✓
Resists server impersonation attack	✓	✓	✗	✗	✓	✓	✗	✓
Resists known key attack	✓	✓	✓	✗	✓	✓	✓	✓
Resists smart card lost attack	✓	✓	✓	✗	✓	✗	✓	✓
Resists man in middle attack	✓	✓	✗	✗	✓	✗	✗	✓
Provides user anonymity	✓	✗	✗	✗	✗	✗	✗	✗
Provides forward secrecy	✓	✓	✓	✓	✓	✓	✓	✓
Provable security	✓	✓	✗	✗	✗	✗	✗	✗
No verifier stored at server	✓	✓	✓	✓	✗	✓	✓	✓
Resists strong replay and denial of services attacks	✓	✗	✗	✗	✗	✓	✗	✗

Table 3.2 illustrates the security comparisons of proposed scheme with related existing schemes. It is evident from Table 3.2 that only proposed scheme provides the user's anonymity and untraceability, while all other schemes are lacking user anonymity and untraceability. Similarly, only the proposed scheme and Irshad et al.'s scheme [58] provides resistance against replay and denial of service attacks. The provable security analysis is provided by proposed and Farash's scheme [3] only, likewise only Farash [3, 65], Arshad et al. and the proposed schemes are resistant to impersonation attacks. In short except proposed scheme, all other schemes are lacking at least two security requirements.

3.5.3.1 Mutual Authentication

In proposed scheme, initially the user sends $\{\overline{username}, V, \overline{W}\}$, where $\overline{W}$ involve user's password PW_i, the adversary without knowing the user password cannot generate valid V and $\overline{W}$ pair. Similarly, without the knowledge of the server's secret key d_S the adversary cannot generate valid W. Further, $Auth_S$ can be generated after having valid W. So the user is authenticated by checking $\overline{W} = h_1(W \oplus V \oplus t_i)$, while the server by verifying $Auth_S = h_2(K\|W\|r\|SK\|t_i)$. Hence, proposed scheme provides mutual authentication.

3.5.3.2 Impersonation Attack

The adversary may impersonate as a legal user, if it successfully generates valid V, W pair. The valid V, W pair requires user PW_i and information stored in smart card so the scheme resist user impersonation attack. Similarly, the adversary can impersonate as a legal server, if he becomes able to generate valid $Auth_S$, but $Auth_S$ involves the computation of $V'' = h(username\|d_S)V$ and $W' = h(username\|V\|V'')$, both of these require the secret key d_S of the server.

3.5.3.3 Privileged Insider Attack

Instead of password we just send $h(PW_i||a)$ during registration phase, so privileged insiders cannot have access to user password PW_i.

3.5.3.4 Stolen Verifier Attack

In proposed scheme, no verifier table is maintained for user's password. $\mathscr{S}$ makes use of his secret key d_S for authentication. Therefore, the proposed scheme is secure against stolen

verifier attack.

3.5.3.5 Man-in-Middle Attack

In proposed scheme, valid V' can only be generated by using user password, while V'' can only be computed by server master key d_S. Therefore, the scheme withstands the man-in-middle attack.

3.5.3.6 Replay Attack

The adversary can easily intercept the request message $\{\overline{username}, V, \overline{W}, t_i\}$. Also the adversary can easily replicate the request message. When such replicated request reaches, the server simply verifies the freshness of t_i, as t_i is old dated, server will know its a replay message. Furthermore, the adversary can generate new time stamp t_a and can replay request after changing t_i by t_a, as time stamp is fresh, server after computing V'' and W', checks $\overline{W} \stackrel{?}{=} h_1(W' \oplus V \oplus t_a)$. The adversary will not pass this test, because $\overline{W}$ contains inbuilt t_i. Similarly, adversary will not be able to compute session key $SK = h_1(K||r||username\|t_i)$ without knowing user password PW_i and either the value of b or c obtaining b from $V = bP$ and c from $C = cP$, the adversary has to solve untraceable elliptic curve discrete logarithm problem. Similarly, if the adversary intercepts $\{realm, Auth_S, C, r\}$ and sends it to the user. The replayed message cannot pass the $Auth_S \stackrel{?}{=} h_2(K\|W\|r\|SK\|t_i)$ test. Therefore, the scheme is secure against replay attack.

3.5.3.7 Offline Password Guessing Attack

Assuming the adversary gets smart card and obtains the secret information (R, a), further the adversary intercepts the message $\{\overline{username}, V, \overline{W}, t_i\}$. In order to guess user password PW_i, the adversary still needs server secret key d_S to check password validity from $V'' = h(username||d_S)V$. Therefore, the proposed scheme resist off-line password guessing attack.

3.5.3.8 Perfect Forward Secrecy

The perfect forward secrecy means that if long term secret keys of one or more legal users are compromised, the secrecy of old session keys will not be affected. For estimating an old session key, the attacker needs to guess more than one session parameters, the random

Table 3.3: Computational Cost Analysis

	Client	Server	Total	Running time
Farash [65]	$3T_{ecpm} + 5T_h$	$4T_{ecpm} + 1_{ecpa} + 5T_h$	$7T_{ecpm} + 1_{ecpa} + 10T_h$	≈ 15.8408
Zhang et al. [64]	$3T_{ecpm} + 4T_h$	$4T_{ecpm} + 1_{ecpa} + 4T_h$	$7T_{ecpm} + 1_{ecpa} + 8T_h$	≈ 15.6292
Irshad et al. [58]	$3T_{ecpm} + 6T_h$	$4T_{ecpm} + 5T_h$	$7T_{ecpm} + 11T_h$	≈ 15.6073
Arshad et al. [42]	$2T_{ecpm} + 4T_h$	$2T_{ecpm} + 4T_h$	$4T_{ecpm} + 8T_h$	≈ 8.9224
Zhang et al. [63]	$4T_{ecpm} + 1T_{ecpa} + 6T_h$	$4T_{ecpm} + 1T_{ecpa} + 5T_h$	$8T_{ecpm} + 2T_{ecpa} + 11T_h$	≈ 17.8909
Tu et al. [2]	$3T_{ecpm} + 1T_{ecpa} + 5T_h$	$3T_{ecpm} + 5T_h$	$6T_{ecpm} + 1T_{ecpa} + 10T_h$	≈ 13.4078
Farash [3]	$3T_{ecpm} + 1T_{ecpa} + 5T_h$	$3T_{ecpm} + 5T_h$	$6T_{ecpm} + 1T_{ecpa} + 10T_h$	≈ 13.4078
Proposed	$3T_{ecpm} + 1T_{ecpa} + 5T_h$	$3T_{ecpm} + 5T_h + 2T_{sed}$	$6T_{ecpm} + 1T_{ecpa} + 10T_h + 2T_{sed}$	≈ 13.417

number b is separately generated by the client $\mathscr{U}$ for each session, while server generates random number c exclusively for each session. In order to find b from $V = bP$ or c from $C = cP$ the adversary has to solve a hard problem $ECDLP$. Hence, the attacker could not estimate the previous session keys out of compromised current session key and/or the password.

3.6 Comparative Performance Analysis

This section describes the comparative computation and communication cost analysis as follows:

3.6.1 Computation Cost Analysis

Following notations are used for computation cost analysis:

- T_{ecpm} : Time for Elliptic curve point multiplication
- T_{ecpa} : Time for Elliptic curve point addition
- T_h : Time for one way hash function
- T_{sed} : Time for a symmetric encryption/decryption operation

According to Kilinc and Yanik [69], T_{ecpm} : takes 2.226 ms, T_{ecpa} takes 0.0288 ms, T_{sed} : takes 0.0046 ms, while T_h : takes 0.0023 ms to complete their processing on a personal computer with Dual CPU E2200 2.20 GHz processor, 2048 MB of RAM and the Ubuntu Operating system by using PBC Library.

Computation cost of proposed scheme as compared with schemes proposed in [2,3,42,58,63–65] is summarized in Table 3.3, the proposed scheme achieves low computation cost as compared with schemes in [58, 63–65]. Arshad et al.'s [42] scheme takes least computation resources

Table 3.4: Storage and Communication Cost Analysis

Schemes →	Our	[3]	[2]	[63]	[42]	[58]	[64]	[65]
Memory needed in smart card	480	320	320	320	160	480	320	320
Communication overhead (Bits)	1184	1056	1056	1056	832	1508	1056	1056
Exchanged messages	2	3	3	3	3	2	3	3

because in their scheme the verifier is stored at server. The proposed scheme incurs only $2T_{sed}$ more on server side as compared with Tu et al.'s and Farsh's schemes [2, 3].

3.6.2 Storage & Communication Cost Analysis

We have also compared the storage and computation costs of proposed scheme with recent related schemes [2, 3, 42, 58, 63–65]. We selected hash function SHA-1, whose output is 160 bit long, further we employed AES as symmetric key algorithm of block size 128 bits. We selected 64 bits *username* length, while size of realm is 32 bits. The NIST recommended size for ECC operations is 160 bits. The storage and communication cost analysis is illustrated in Table 3.4. Proposed scheme incurs some extra storage in smart card and having some more communication overhead as compared with schemes [2, 3, 42, 63–65], while it is having equal storage and less communication cost as compared with [58]. Furthermore, only proposed scheme and Irshad et al.'s scheme [58] achieves authentication in only 2 messages, while rest of the schemes [2, 3, 42, 63–65] achieves same in 3 messages. Hence, proposed scheme is more suitable for practical environments.

3.7 Chapter Summary

This chapter analyzed Tu et al.'s authentication and key agreement scheme for SIP and Farash's improvement on Tu et al.'s scheme. We have shown that Tu et al.'s scheme is vulnerable to server impersonation attack. Further, we have also shown that both Tu et al.'s scheme and Farash's improvement do not provide user anonymity and are vulnerable to replay as well as denial of services attack. To overcome the weaknesses, we have proposed an improved privacy preserving scheme, which ensures mutual authentication and is secure against all known attacks.

Chapter 4

A Remote User Authentication Scheme Using ECC

Swift advancement in wireless and communication technologies has led to their immense growth and utilization in day to day life. A large number of people are getting advantages of these wireless devices such as smartphones, notebooks and many other portable and smart devices. These smart devices enable public to utilize sundry online services at any time and place. These online services are offered in the form of net-browsing, video conferencing, telemedicine information system, VoIP and government services. However, intrinsic Internet infrastructure can be compromised easily because it is openly accessible to everyone. Therefor any adversary can steal, snoop and modify the information shared between authentic users. All these factors demand an authentication scheme in order to secure the message transmission and maintain the privacy of the participants. Early on, password based authentication techniques were introduced in order to mitigate the security concerns. Lamport [70] took an initiative in this regard by developing first password based scheme for authentication. Later on, various password based schemes have been introduced by researchers for diverse applications [10, 71–74].

Soon it was realized that these single factor or password based authentication schemes can be breached easily and therefore fails to offer ample retreat against possible threats. The foundation stone laid by password based schemes provides the base for the emergence of new schemes. Therefore, researchers introduced such authentication scheme that utilized two-factor approach [6, 51–53, 57–59, 75–79] in order to offer more safety. Smart card is used as a second factor alongside good old factor password in two-factor schemes.

Nevertheless, two-factor authentication schemes offer more security and reliability, but most

of the systems around us in communication technologies are resource constrained in nature. Therefore, these systems appreciate such authentication schemes that involve lightweight computation operations such as random numbers and simple one-way hash functions. An efficient and computationally effective scheme is presented by Tsai et al. [80] that utilized the random numbers and simple one-way hash functions to achieve reasonable security. Although, several lightweight schemes have been presented [81–83] and are becoming common due to abridged computation cost but reduction in computation is achieved at the expense of security. In other words lightweight schemes don't offer reliable and comprehensive security and can be compromised easily [34, 84, 85].

Juang et al. [86] utilized elliptic curve cryptosystem for their key agreement and authentication scheme in order to reduce computation and transmission cost. Xu et al. [87] introduced an enhanced two-factor scheme when they noticed that two schemes of Lee et al [88, 89] are vulnerable to offline password guessing and forgery attacks. Juang et al. also proved the security of their scheme through the random oracle model along the assumption of computational Diffie-Hellman scheme.

Later on, Sood et al. [90] and Song and Rongong [91] found that Xu et al.'s scheme can be compromised by impersonation and internal attacks, therefore they presented an enhanced scheme in order to mitigate the chance of said attacks. Then Chen et al. [92] analyzed both enhanced schemes and declared that Sood et al.'s scheme fails to provide mutual authentication, whereas Song and Ronggong's scheme is susceptible to offline password guessing and stolen smart card attack. Chen et al. introduced enhanced scheme and stated that their scheme is protected against all well-known attacks. Jiang et al. soon realized that Chen et al.'s is vulnerable to offline dictionary attack and moreover doesn't attain user anonymity.

Qu et al. [93] presented two-factor key agreement scheme for authentication and claimed that their scheme is invincible against impersonation, and stolen smart card attacks and offers user anonymity. Later on, Huang et al. [4] proved the claim of Qu et al. null and void and declared that their scheme is still vulnerable to impersonation, and stolen smart card attacks. Therefore, Huang et al. introduced an enhanced key agreement scheme for authentication. However, this chapter proves that Huang et al.'s scheme has correctness issues and can be compromised by impersonation attack. This chapter introduced more enriched key agreement scheme to prevent forgery attack and resolve correctness issues present in the Huang et al.'s scheme. The section wise organization for the rest of this chapter is as follows: The scheme of Huang et al. is reviewed in section 4.1 and then cryptanalysis of Huang et al.'s scheme is discussed in section 4.2. After cryptanalysis of Huang et al.'s scheme, proposed scheme is introduced in section 4.3. Then security analysis and its verification through ProVerif is

Table 4.1: Notation Guide

Notations	Description	Notations	Description
RC, $\mathcal{S}_j$	Registration center, Server	$\mathcal{U}_i$, Adv	User, Attacker
SID_j, ID_{ui}	identities of $\mathcal{S}_j$, $\mathcal{U}_i$	PW_{ui}, BIO_{ui}	$\mathcal{U}_i$'s password and Biometrics
x_{ui}	$\mathcal{U}_i$'s private key	Pub_{sj}, Pri_{sj}	Public and private key pair of $\mathcal{S}_j$
PSK_{rs}	Secret key between $\mathcal{S}_j$ and RC	SC_{ui}	$\mathcal{U}_i$'s smart card
$h(.)$, $H(.)$	Hash and Bio hash functions	$\|$, $\oplus$	Concatenation, XOR operators

presented in section 4.4 and 4.5 respectively. Performance and security comparisons are given in section 4.6. Finally, chapter's summary is solicited in section 4.7.

4.1 Review of Huang et al.'s Scheme

This section presents the review of Huang et al.'s scheme [4]. The scheme of Huang et al. is composed of four phases, which are illustrated in Fig. 4.1. The details of these phases are described as follows:

4.1.1 Registration Phase

The registration phase involves three steps. The user $\mathcal{U}_a$ picks up his identity ID_{ua}, password PW_{ua} along with a random number r_{ua}. Then one-way hash function is applied over concatenated ID_{ua}, PW_{ua} and r_{ua}. Then user $\mathcal{U}_a$ communicates registration entreaty $\{ID_{ua}, H_1(ID_{ua}\|PW_{ua}\|r_{ua})\}$ towards server $\mathcal{S}$ through protected strait. The server $\mathcal{S}$ determines $AID_{ua} = (H_1(msk) + 1).H_1(ID_{ua}\|PW_{ua}\|r_{ua}).P$, $BID_{ua} = H_2(H_1(ID_{ua}).H_1(ID_{ua} - \|PW_{ua}\|r_{ua})$ against registration entreaty. The server then hoards AID_{ua} and BID_{ua} into smart card and this smart card is delivered to user $\mathcal{U}_a$ through protected strait. The user $\mathcal{U}_a$ inserts r_{ua} into smart card after acquiring it from server $\mathcal{S}$. Therefore, at the end of the registration phase smart card holds $\{AID_{ua}, BID_{ua}, r_{ua}\}$.

4.1.2 Login Phase

The login phase finishes in following two phases:

Step LP1: User $\mathcal{U}_a$ enters his/her smart card into specific smart card reader and types in his/her unique ID_{ua} and password PW_{ua}. The smart card computes $BID'_{ua} = H_2(H_1(ID_{ua}).H_1(ID_{ua}\|PW_{ua}\|r_{ua})$ and it verifies, whether the computed BID'_{ua} is

equal to BID_{ua} which is already engraved in the smart card. If this equality holds ID_{ua} and password PW_{ua} are considered as valid, otherwise the session is terminated.

Step LP2: Smart card yields q_{ua} and $Q_{ua} = q_{ua}.P$ and calculates $M_{ua} = q_{ua}.mpk$, $TID_{ua} = AID_{ua} - H_1(ID_{ua}\|PW_{ua}\|r_{ua}).P$, $CID_{ua} = H_4(ID_{ua}\|M_{ua}) \oplus H_2(M_{ua}\|TID_{ua})$, $DID_{ua} = M_{ua} \oplus H_1(ID_{ua}\|PW_{ua}\|r_{ua}).P$, $EID_{ua} = H_3(H_4(ID_{ua}\|M_{ua})\|Q_{ua}\|M_{ua})$. Finally, $\mathcal{U}_a$ transmits login entreaty towards server in the form of $\{CID_{ua}, DID_{ua}, EID_{ua}, Q_{ua}\}$.

4.1.3 Authentication Phase

In authentication phase, the server $\mathcal{S}$ performs following steps in response to login entreaty from user $\mathcal{U}_a$:

Step AP1: The server determines $M'_{ua} = msk.Q_{ua}$, $H_1(ID_{ua}\|PW_{ua}\|r_{ua}).P = DID_{ua} \oplus M'_{ua}$, $TID'_{ua} = H_1(msk).(DID_{ua} \oplus M'_{ua})$, $H_4(ID_{ua}\|M_{ua}) = CID_{ua} \oplus H_2(M'_{ua}\|TID'_{ua})$, $EID'_{ua} = H_3(H_4(ID_{ua}\|M_{ua})\|Q_{ua}\|M'_{ua})$. Then server checks either $EID'_{ua} \stackrel{?}{=} EID_{ua}$ holds or not, if it doesn't then the session in turn is terminated, else $\mathcal{U}_a$ is assumed as authorized user. The server then yields random number q_{sb} and calculates $Q_{sb} = q_{sb}.Q_{ua}$, $T_{sb} = Q_{sb} \oplus M_{ua}$ and $H_{sb} = H_3(EID'_{ua}\|Q_{sb}\|TID'_{ua})$. The server $\mathcal{S}$ then transmits $\{T_{sb}, H_{sb}\}$ in response to login entreaty from user $\mathcal{U}_a$.

Step AP2: The user $\mathcal{U}_a$ computes $Q'_{sb} = T_{sb} \oplus M_{ua}$ and $H'_{sb} = H_3(EID_{ua}\|Q'_{sb}\|TID_{ua})$ after that verifies the condition $H'_{sb} \stackrel{?}{=} H_{sb}$. The session will be terminated on successful verification, else $H_{ua} = H_2(Q_{ua}\|Q'_{sb})$ is computed along with session key $SK = H_5(Q_{ua}\|Q_{sb}\|M_{ua}\|TID_{ua})$. Then at the end user $\mathcal{U}_a$ sends $\{H_{ua}\}$ towards server $\mathcal{S}$.

Step AP3: The server also calculates session key after getting $\{H_{ua}\}$ from user $\mathcal{U}_a$. Then it computes $H'_{ua} = H_2(Q'_{ua}\|Q_{sb})$ and verifies $H'_{ua} \stackrel{?}{=} H_{ua}$, if it doesn't hold, the session is immediately terminated, else session key SK will be declared legal.

The computed shared key between $\mathcal{U}_a$ and $\mathcal{S}$ is:

$$SK = H_5(Q_{ua}\|Q_{sb}\|M_{ua}\|TID_{ua}) \tag{4.1}$$

Figure 4.1: Huang et al.'s Scheme

4.2 Cryptanalysis of Huang et al.'s Scheme

In this section, we have performed cryptanalysis of Huang et al.'s authentication scheme under the mentioned adversarial model in subsection 2.2.6. It is shown that Huang et al.'s scheme is susceptible to user forgery/impersonation attack and having incorrect notion of perfect anonymity.

4.2.1 User Impersonation Attack

In this subsection, we prove that an adversary after registering to the system can forge himself as any other user of the system. Let Adv be a dishonest registered user of the system. Adv will perform following steps to deceive the server:

Step UFA1: Adv extracts the information stored in his smart card $\{AID_{adv}, BID_{adv}, r_{adv}\}$. Adv using his password PW_{adv} and identity ID_{adv} computes following:

$$TID_{adv} = AID_{adv} - H_1(ID_{adv} \| PW_{adv} \| r_{adv}) \tag{4.2}$$

$$XID_{adv} = TID_{adv}.(H_1(ID_{adv} \| PW_{adv} \| r_{adv}))^{-1} = H_1(msk).P \tag{4.3}$$

Step UFA2: Adv computes:

$$M_{Adv} = q_{adv}.mpk \tag{4.4}$$

$$CID_{Adv} = H_4(ID_{ua} \| M_{Adv}) \oplus H_2(M_{Adv} \| TID_{Adv}) \tag{4.5}$$

$$DID_{Adv} = M_{Adv} \oplus Z.P \tag{4.6}$$

$$EID_{ua} = H_3(H_4(ID_{ua} \| M_{Adv}) \| Q_{adv} \| M_{Adv}) \tag{4.7}$$

Step UFA3: After that Adv sends $\{CID_{Adv}, DID_{Adv}, EID_{Adv}, Q_{adv}\}$ towards servers.

Step UFA4: Receiving $\{CID_{Adv}, DID_{Adv}, EID_{Adv}, Q_{adv}\}$ from Adv. The server computes

following:

$$M^{'}_{Adv} = msk.Q_{adv} \tag{4.8}$$
$$Z.P = DID_{Adv} \oplus M^{'}_{Adv} \tag{4.9}$$
$$TID^{'}_{Adv} = H_1(msk).(DID_{Adv} \oplus M^{'}_{Adv}) \tag{4.10}$$
$$H_4(ID_{ua} \| M_{Adv}) = CID_{Adv} \oplus H_2(M^{'}_{Adv} \| TID^{'}_{Adv}) \tag{4.11}$$
$$EID^{'}_{Adv} = H_3(H_4(ID_{ua} \| M_{Adv}) \| Q_{adv} \| M^{'}_{Adv}) \tag{4.12}$$

Step UFA5: The server $\mathcal{S}$ verifies $EID^{'}_{Adv} \stackrel{?}{=} EID_{Adv}$, the session is immediately terminated in case the condition gets false, else it produces q_{sb} (a random number) and computes the following:

$$Q_{sb} = q_{sb}.Q_{adv} \tag{4.13}$$
$$T_{sb} = Q_{sb} \oplus M_{Adv} \tag{4.14}$$
$$H_{sb} = H_3(EID^{'}_{Adv} \| Q_{sb} \| TID^{'}_{Adv}) \tag{4.15}$$

Step UFA6: Then $\{T_{sb}, H_{sb}\}$ is transmitted towards Adv by the server.

Step UFA7: On receiving $\{T_{sb}, H_{sb}\}$, Adv computes the following:

$$Q^{'}_{sb} = T_{sb} \oplus M_{Adv} \tag{4.16}$$
$$H^{'}_{sb} = H_3(EID_{Adv} \| S^{'}_{sb} \| TID_{Adv}) \tag{4.17}$$

Step UFA8: Adv verifies $H^{'}_{sb} \stackrel{?}{=} H_{sb}$, if it doesn't hold session is immediately terminated, else Adv calculates $SK = H_5(Q_{adv} \| Q_{sb} \| M_{Adv} \| TID_{Adv})$ and computes:

$$H_{adv} = H_2(Q_{adv} \| Q^{'}_{sb}) \tag{4.18}$$

Step UFA9: $\{H_{adv}\}$ is transmitted to server $\mathcal{S}$.

Step UFA10: Server $\mathcal{S}$ receives $\{H_{adv}\}$ and computes $SK = H_5(Q_{adv} \| Q_{sb} \| M_{Adv} \| TID_{Adv})$ and

$$H^{'}_{adv} = H_2(Q_{adv} \| Q^{'}_{sb}) \tag{4.19}$$

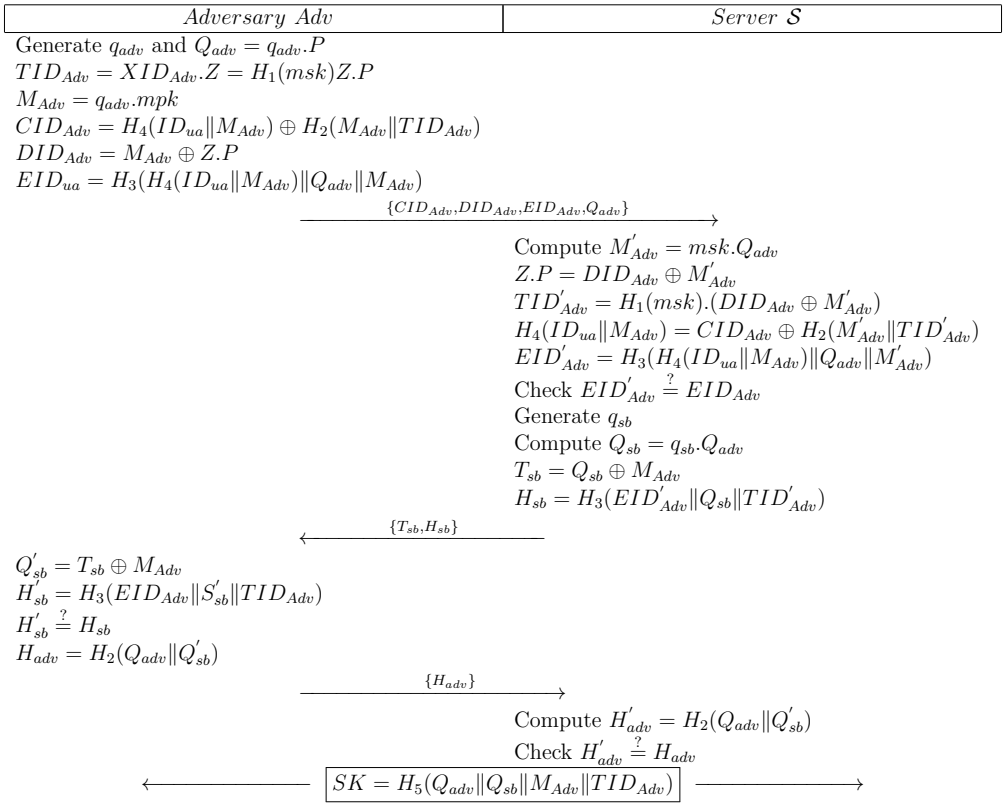


Figure 4.2: User Impersonation Attack on Huang et al.'s Scheme

Step UFA11: Lastly, server $\mathcal{S}$ checks $H'_{adv} \stackrel{?}{=} H_{adv}$, if the condition holds true then it verifies that the server $\mathcal{S}$ has the shared session key.

Hence, it can be concluded that Adv has impersonated successfully on behalf of $\mathcal{U}_a$ by betraying server $\mathcal{S}$.

4.2.2 Incorrect Notion of Perfect Anonymity

Huang et al. introduced a new notion of perfect anonymity, where a server remains unable to recognize the identity of a user requesting to login. In our opinion, such notion of perfect anonymity is erroneous and is not desirable in any environment, because if the server is not able to know the identity of a user, he will not be able to provide him user's specific services. Furthermore, in case of generic services, the user will remain enjoying the services provided by the server even if he unregistered the system or his lease expires.

4.3 Proposed Scheme

This section discusses the proposed enhancements made in the Huang et al.'s scheme. Since, it is proved that Huang et al.'s scheme is susceptible to forgery attack, therefore Huang et al.'s scheme has been modified accordingly, which is illustrated as proposed scheme in Fig. 4.3

4.3.1 Registration Phase

The registration phase involves three steps. The user $\mathcal{U}_a$ picks up his distinctive ID_{ua}, password PW_{ua} along with random number r_{ua}. Then one-way hash function is applied over concatenated ID_{ua}, PW_{ua} and r_{ua}. Then user $\mathcal{U}_a$ communicates registration entreaty $\{ID_{ua}, H_1(ID_{ua}\|PW_{ua}\|r_{ua})\}$ towards server $\mathcal{S}$ through protected strait. The server $\mathcal{S}$ determines $AID_{ua} = (H_1(msk \oplus ID_{ua}) + H_1(ID_{ua}\|PW_{ua}\|r_{ua})).P$, $BID_{ua} = H_2(H_1(ID_{ua}).H_1(ID_{ua} -$

$\|PW_{ua}\|r_{ua})$ against registration entreaty. The server then hoards AID_{ua} and BID_{ua} into smart card and this smart card is delivered to user $\mathcal{U}_a$ through protected strait. The user $\mathcal{U}_a$ inserts r_{ua} into smart card after acquiring it from server $\mathcal{S}$. Therefore, at the end of the registration phase smart card holds $\{AID_{ua}, BID_{ua}, r_{ua}\}$.

4.3.2 Login Phase

The login phase finishes in following two phases:

Step LP1: User $\mathcal{U}_a$ enters his/her smart card into specific smart card reader and type in his/her unique ID_{ua} and password PW_{ua}. The smart card computes $BID'_{ua} = H_2(H_1(ID_{ua}).H_1(ID_{ua}\|PW_{ua}\|r_{ua})$ and after that it verifies, does the computed BID'_{ua} is equal to BID_{ua} that is already engraved in the smart card. If this equality holds ID_{ua} and password PW_{ua} are considered as valid, otherwise session is terminated.

Step LP2: Smart card yields q_{ua} and $Q_{ua} = q_{ua}.P$ and calculates $M_{ua} = q_{ua}.mpk$, $TID_{ua} = AID_{ua} - H_1(ID_{ua}\|PW_{ua}\|r_{ua}).P$, $DID_{ua} = M_{ua} \oplus ID_{ua}$ and $EID_{ua} = H_3(H_4(TID_{ua} - \|M_{ua})\|Q_{ua}\|M_{ua})$. Finally, $\mathcal{U}_a$ transmits login entreaty to server in the form of $\{DID_{ua}, EID_{ua}, Q_{ua}\}$.

4.3.3 Authentication Phase

In authentication phase, the server $\mathcal{S}$ follows the following steps in response to login entreaty from user $\mathcal{U}_a$:

Step AP1: The server determines $M_{ua}^{'} = msk.Q_{ua}$, $ID_{ua}^{'} = M_{ua}^{'} \oplus DID_{ua}$, $TID_{ua}^{'} = H_1(msk \oplus ID_{ua}P$, $EID_{ua}^{'} = H_3(H_4(TID_{ua}^{'} \| M_{ua}^{'}) \| Q_{ua} \| M_{ua}^{'})$. Then server checks either $EID_{ua}^{'} \stackrel{?}{=} EID_{ua}$ holds or not, if it doesn't then the session in turn is terminated, else $\mathcal{U}_a$ is assumed as authorized user. The server then yields random number q_{sb} and calculates $Q_{sb} = q_{sb}.Q_{ua}$, $T_{sb} = Q_{sb} \oplus M_{ua}$ and $H_{sb} = H_3(EID_{ua}^{'} \| Q_{sb} \| TID_{ua}^{'})$. The server $\mathcal{S}$ then transmits $\{T_{sb}, H_{sb}\}$ in response to login entreaty from user $\mathcal{U}_a$.

Step AP2: The user $\mathcal{U}_a$ computes $Q_{sb}^{'} = T_{sb} \oplus M_{ua}$ and $H_{sb}^{'} = H_3(EID_{ua} \| Q_{sb}^{'} \| TID_{ua})$ after that verifies the condition $H_{sb}^{'} \stackrel{?}{=} H_{sb}$. The session will be terminated on successful verification, else $H_{ua} = H_2(Q_{ua} \| Q_{sb}^{'})$ is computed along with session key $SK = H_5(Q_{ua} \| Q_{sb} \| M_{ua} \| TID_{ua})$. Then at the end user $\mathcal{U}_a$ sends $\{H_{ua}\}$ towards server $\mathcal{S}$.

Step AP3: The server also calculates session key after getting $\{H_{ua}\}$ from user $\mathcal{U}_a$. Then it computes $H_{ua}^{'} = H_2(Q_{ua}^{'} \| Q_{sb})$ and verifies $H_{ua}^{'} \stackrel{?}{=} H_{ua}$, if it doesn't hold then session is immediately terminated, else session key SK will be declared legal.

4.4 Security Analysis

Security analysis related to proposed scheme is presented in this section. This analysis verifies the robustness and shows the invincibility of proposed scheme against number of well-known attacks beneath the joint adversarial model, given in section 2.2.6. The detailed evidences are given in the subsequent subsections:

4.4.1 Anonymity and Privacy

Identity ID_{ua} of $\mathcal{U}_a$ is not transferred in cleartext, instead DID_{ua} is calculated using $DID_{ua} = M_{ua} \oplus ID_{ua}$. Therefore, the server $\mathcal{S}$ can only find the identity of $\mathcal{U}_a$. Moreover, $Q_{ua} = q_{ua}.P$ contains q_{ua} which is session specific and resist the adversary to foretell whether same user has initiated the two distinct sessions or not.

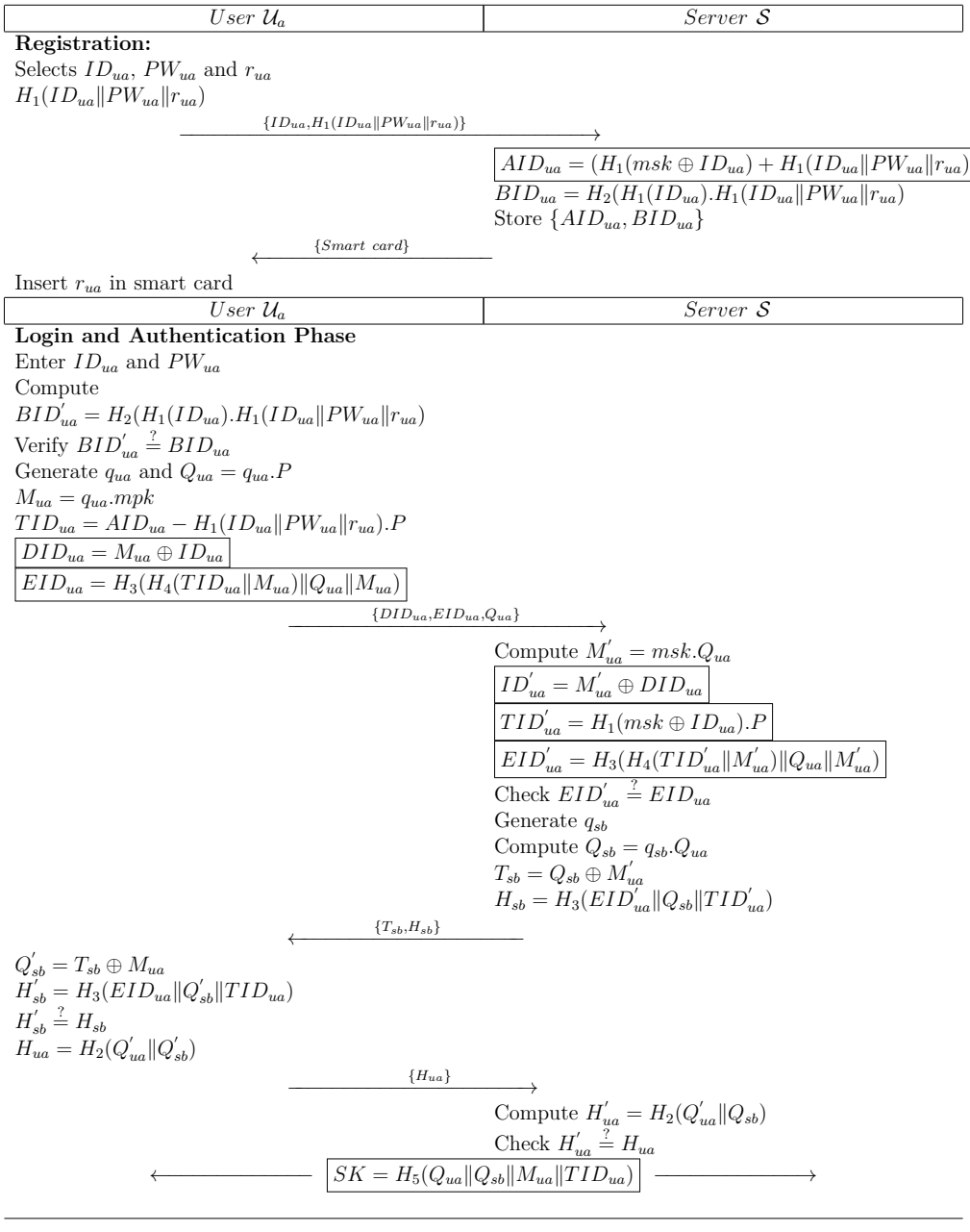


Figure 4.3: Proposed Scheme

4.4.2 Mutual Authentication

$\mathcal{S}$ authenticates $\mathcal{U}_a$ by confirming $EID^{'}_{ua} \stackrel{?}{=} EID_{ua}$. Adversary requires to find $H_4(TID^{'}_{ua} \| M^{'}_{ua})$ in order to correctly calculate $EID^{'}_{ua}$. Moreover, computation of $H_4(TID^{'}_{ua} \| M^{'}_{ua})$ involves both smart card and password of $\mathcal{U}_a$. Correspondingly, $\mathcal{U}_a$ authenticates $\mathcal{S}$ by confirming $H^{'}_{sb} \stackrel{?}{=} H_{sb}$. Therefore, it can be concluded that proposed scheme offer mutual authentication because only legitimate user can clear the authentication trial imposed by the server and vice versa.

4.4.3 User and Server Impersonation Attacks

Authentication request $\{DID_{ua}, EID_{ua}, Q_{ua}\}$ and response $\{H_{ua}\}$ against challenge message $\{T_{sb}, H_{sb}\}$ from server can only be made by legitimate user. Likewise, only legitimate server can answer to authentication request with challenge message $\{T_{sb}, H_{sb}\}$ as substantiated in subsection 4.4.2.

4.4.4 Smart Card Theft/Stolen Attack

Consider an adversary is able to get $\mathcal{U}_a$'s smart card. Then adversary can easily retrieve engraved values $AID_{ua} = (H_1(msk \oplus ID_{ua}) + H_1(ID_{ua} \| PW_{ua} \| r_{ua})).P$, $BID_{ua} = H_2(H_1(ID_{ua}).H_1(ID_{ua} \| PW_{ua} \| r_{ua})$ and r_{ua}. But in order to guess the secret factor or parameter the adversary still requires PW_{ua}. Therefore, adversary cannot take any advantage of getting or stealing smart card for imitation.

4.4.5 Replay Attack

Suppose an adversary is able to intercept and replay the message but adversary will fail to respond the challenge message coming from server. Therefore, proper replay attack is not possible on the proposed scheme.

4.4.6 Perfect Forward Secrecy

The session key that is computed between $\mathcal{U}_a$ and $\mathcal{S}$ encloses Q_{ua} and Q_{sb} from both contributors respectively. Therefore, if adversary is able to get long term private key of any

contributor, he will still be unable to find preceding session keys. Hence, proposed scheme can be declared to enjoy perfect forward secrecy.

4.4.7 Insider and Stolen Verifier Attacks

Proposed scheme doesn't insist any verifier table and also $\mathcal{S}$ doesn't maintain any data or parameter concerning password PW_{ua} of $\mathcal{U}_a$ help to avoid stolen verifier attack. Moreover, $\mathcal{U}_a$ doesn't expose his/her PW_{ua} by sending it in plaintext. So, any insider will be unable to know and misuse $\mathcal{U}_a$'s password.

4.4.8 Password Guessing Attack

Password PW_{ua} of $\mathcal{U}_a$ is secured with his/her unique ID_{ua} and a random number r_{ua}. Further, one-way hash function is applied over concatenation of PW_{ua} with ID_{ua} and r_{ua}. Moreover, smart card doesn't maintain any parameter to provide any kind of clue regarding password validity. Hence, it can be concluded that it is infeasible for any adversary to launch offline password guessing attack.

4.4.9 No Clock Synchronization

Both the participants generate their own random numbers and don't utilize time stamps at all. Therefore, the proposed scheme doesn't impose the overhead of clock synchronization and in turn save precious resources.

4.4.10 Formal Security Analysis

To demonstrate that proposed scheme is provably secure, we adopted the same analysis as mentioned in [8, 94]. Following oracles are defined for analysis purpose:

- ***Reveal:*** This oracle unconditionally outputs a string S from the one way hash function $R = h(S)$.
- ***Extract:*** This oracle unconditionally outputs the scalar multiplier k out of a given elliptic curve points $O = kP$ and P.

Theorem 2. *The proposed remote user authentication scheme is provably secure against an attacker $\mathcal{A}$ for resolution of $\mathcal{U}_a$'s identity (ID_{ua}), the private key (msk) of the server $\mathcal{S}$ and the computed session key SK between $\mathcal{U}_a$ and $\mathcal{S}$ under the hardness assumption of $ECDLP$ and ruminating the secure hash function as random oracle.*

Proof. Consider an adversary $\mathcal{A}$ with capabilities to derive $\mathcal{U}_a$'s ID_{ua}, $\mathcal{S}$'s secret key msk and computed session key SK. $\mathcal{A}$ executes the algorithmic experiment $EXPE1^{ECDLP,HASH}_{\mathcal{A},PRUAS}$ against the proposed remote user authentication scheme $PRUAS$ by simulating both the oracles *Extract* and *Reveal*. We define the success probability of the above cited experiment as $Succe_1 = |Prb[EXPR1^{ECDLP,HASH}_{\mathcal{A},PRUAS} = 1] - 1|$. The advantage carried by $\mathcal{A}$ is defined as $Advt1^{HASH,ECDLP}_{\mathcal{A},TFBAMS}(t_e, q_{ex}, q_{rv}) = max_{\mathcal{A}}(Succe_1)$, Where $\mathcal{A}$ can make maximum q_{ex} *Extract* and q_{rv} *Reveal* queries. According to the experiment $\mathcal{A}$ can compute ID_{ua}, msk and SK iff he can (i) invert secure hash function and (2) break the ECDLP. However, referring to Definition 1 it is computationally infeasible to invert a secure one way hash function, similarly by Definition 2 it is computationally infeasible to break ECDLP. Hence, we have $Advt1^{ECDLP,HASH}_{\mathcal{A},PRUAS}(t, q_{rv}, q_{ex}) \leq \epsilon$. Therefore, proposed remote user authentication scheme is invincible against an adversary $\mathcal{A}$ to compute $\mathcal{U}_a$'s ID_{ua}, $\mathcal{S}$'s secret key msk and computed session key SK. □

Algorithm 1 $EXPR1^{ECDLP,HASH}_{\mathcal{A},PRUAS}$

1: Eavesdrop the login message $\{DID_{ua}, EID_{ua}, Q_{ua}\}$, Where $DID_{ua} = M_{ua} \oplus ID_{ua}$, $EID_{ua} = H_3(H_4(TID_{ua}\|M_{ua})\|Q_{ua}\|M_{ua})$, $Q_{ua} = q_{ua}.P$
2: Call *Reveal* oracle on EID_{ua} and get $(H_4(TID_{ua}\|M_{ua})'\|Q'_{ua}\|M'_{ua}) \leftarrow Reveal(EID_{ua})$
3: Call *Reveal* oracle on $(H_4(TID_{ua}\|M_{ua})'$ and get $(TID'_{ua}\|M''_{ua}) \leftarrow Reveal(H_4(TID_{ua}\|M_{ua})')$
4: **if** $(M''_{ua} = M'_{ua})$ **then**
5: Compute $EID'_{ua} = H_3(H_4(TID'_{ua}\|M'_{ua})\|Q_{ua}\|M'_{ua})$
6: **if** $(EID_{ua} = EID'_{ua})$ **then**
7: Accept ID'_{ua}
8: Call *Extract* oracle on TID'_{ua} to get $H_1(msk \oplus ID_{ua})' \leftarrow Extract(TID'_{ua})$
9: Call *Reveal* oracle on $H_1(msk \oplus ID_{ua})'$ and get $(msk \oplus ID_{ua})' \leftarrow Reveal(H_1(msk \oplus ID_{ua})'$
10: Compute $msk' = (msk \oplus ID_{ua})' \oplus ID'_{ua}$
11: Eavesdrop the challenge message $\{T_{sb}, H_{sb}\}$, Where $T_{sb} = Q_{sb} \oplus M'_{ua}$, $H_{sb} = H_3(EID'_{ua}\|Q_{sb}\|TID'_{ua})$
12: Compute $Q'_{sb} = T_{sb} \oplus M'_{ua}$
13: Compute $H'_{sb} = H_3(EID_{ua}\|Q_{sb}\|TID_{ua})$
14: **if** $(H'_{sb} = H_{sb})$ **then**
15: Accept msk
16: Eavesdrop the response message $\{H_{ua}\}$, Where $H_{ua} = H_2(Q_{ua}\|Q'_{sb})$
17: Compute $H'_{ua} = H_2(Q_{ua}\|Q'_{sb})$
18: **if** $(H'_{ua} = H_{ua})$ **then**
19: Compute session key $SK = H_5(Q_{ua}\|Q_{sb}\|M'_{ua}\|TID'_{ua})$
20: **else**
21: **return** Fail
22: **end if**
23: **else**
24: **return** Fail
25: **end if**
26: **else**
27: **return** Fail
28: **end if**
29: **else**
30: **return** Fail
31: **end if**

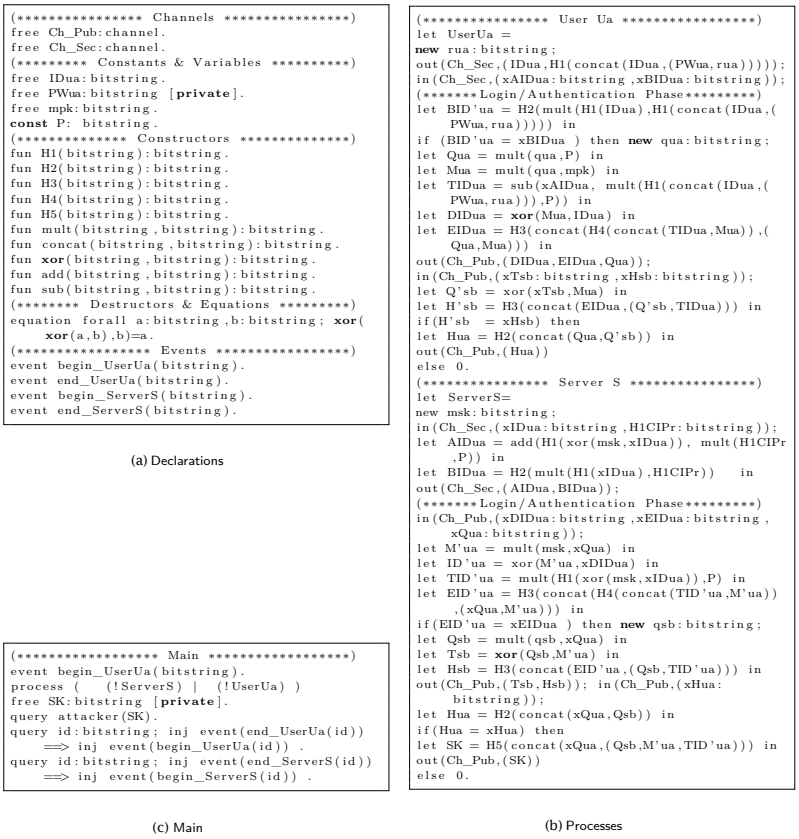


Figure 4.4: ProVerif Validation

4.5 Formal Security Verification using ProVerif

This section presents the security validation proof of proposed scheme through formal automated application ProVerif [34]. As shown in Fig. 4.4(a) two channels public *Ch_Pub* and private *Ch_Sec* are defined along with cryptographic functions, which are demarcated as constructors and equations within declaration part. Whereas, in process part two processes are implemented that are designated as $UserUa$ and $ServerS$ also shown in Fig. 4.4(b). The main part as shown in Fig. 4.4(c) actually models the starting and ending events for each user and server process. The scheme mimic the parallel execution of both user and server processes. At the end, proposed scheme's correctness and session key's secrecy is evaluated using queries and the corresponding results are as under:

1. RESULT inj-event(end˙ServerS(id)) ==> inj-event($begin\_ServerS(id)$) is true.
2. RESULT inj-event(end_UserUa(id_1235)) ==> inj-event($begin\_UserUa(id\_1235)$) is true.

Table 4.2: Performance Comparison

Scheme:	Proposed	Huang et al. [4]	Qu et al. [93]
Computation cost	$6t_{pme} + 1t_{pae} + 12t_{hf}$	$6t_{pme} + 1t_{pae} + 17t_{hf}$	$9t_{pme} + 5t_{pae} + 13t_{hf}$
Communication cost	960	1120	1120

3. RESULT not attacker(SK[]) is true.

First two results ratify the correctness of the proposed scheme due to successful initiation and termination of the user and server processes. They also ensure that the proposed scheme holds the reachability characteristics. Third result proves that session key ($SK[]$) cannot be compromised by the adversary. Hence, proposed scheme can be declared as correct and achieve reachability along with secrecy characteristics.

4.6 Performance and Security Comparisons

This section highlights the comprehensive performance and security comparison of related schemes with the proposed scheme. Subsequent notations are utilized for performance comparison:

- t_{hf} : time to compute Hash code.
- t_{pme} : time to perform point multiplication.
- t_{pae} : time to perform point addition.

Table 4.2 illustrates the performance comparisons, it is obvious that proposed scheme is lightweight as compared to schemes of Huang et al. [4] and Qu et al. [93] in terms of computation cost. Moreover, proposed scheme also outperforms the schemes of Huang et al. and Qu et al. in terms of communication cost. Security comparison of proposed scheme with related schemes is illustrated in Table 4.3 under the said adversarial model presented in section 2.2.6. The security comparison reveals that proposed scheme performs better than the related schemes as it remains invincible against the known attacks. Whereas Huang et al.'s scheme is vulnerable to forgery or impersonation attack. Moreover, Qu et al.'s scheme fails to provide forward secrecy and is susceptible to forgery, smart card stolen and password guessing attacks. Hence, it can be declared that the proposed scheme is not only lightweight but it also offers additional security features in order to maintain its invincibility against well-known attack.

Table 4.3: Comparison of Security Parameters

Scheme:	Proposed	Huang et al. [4]	Qu et al. [93]
Anonymity and privacy	Yes	Yes	Yes
Mutual Authentication	Yes	Yes	Yes
Resists forgery attack	Yes	No	No
Resists smart card theft attack	Yes	Yes	No
Resists replay attack	Yes	Yes	Yes
Forward secrecy	Yes	Yes	No
Resists insider/Stolen verifier attacks	Yes	Yes	Yes
Resists password guessing attack	Yes	Yes	No
No clock synchronization	Yes	Yes	Yes

4.7 Chapter Summary

In this chapter, we have analyzed Huang et al.'s remote user authentication scheme using elliptic curve cryptography. The comprehensive analysis has shown that Huang et al.'s scheme is prone to user impersonation attack. Then we proposed an improved scheme to overcome the weaknesses. We have proved the security of proposed scheme in random oracle model. Furthermore, we have also performed automated security validation using the popular automated tool ProVerif. The analysis has shown that proposed scheme is more robust and more lightweight as compared with Huang et al.'s scheme. Hence, due to better security and performance, the proposed scheme is more suitable for security sensitive and resource constrained environments.

Chapter 5

An Anonymous Remote User Authentication Scheme Based on Symmetric Key Cryptography

The most efficient and widely used method to solve security issues on public networks is the smart card based password authentication scheme which was first proposed by Chang et al. [40]. Subsequently, a large number of smart card based authentication schemes are proposed [5, 36, 42, 50–53, 58, 59, 63, 65, 83, 94–103].

The past research on authentication has ascertained that the design of a correct authentication scheme is exceptionally difficult [84] as smart card is very small device equipped with limited computation, memory and power resources. So authentication schemes [5, 83, 94, 97–103] based on symmetric key primitives (HASH, MAC, XOR, symmetric encryption etc.) look more desirable, instead of the schemes [36, 42, 50–53, 58, 59, 63, 65, 95, 96] based on expensive asymmetric primitives (point multiplication, exponentiation, pairing etc.). However, keeping in mind the sensitivity of tasks (e.g. financial, healthcare) carried out by such schemes which are also having additional threats as compared to traditional threats, asymmetric cryptography looks more promising which can resists impersonation, password guessing and replay attacks. Besides security, privacy and anonymity has emerged as of wide interest. If the privacy of user is compromised the adversary can predict victim's life style, habits and in some cases the location of remote user. There are two main properties for user anonymity: user identity hiding and untraceability. The first one guarantees that adversary cannot reveal real identity of user while the latter means adversary cannot figure out two different sessions are initiated by same user [84].

Symmetric key based authentication schemes are more suitable for resource constrained devices. Till now a number of symmetric key based anonymous authentication schemes are proposed [83, 94, 97–101]. Unfortunately, all such schemes are either vulnerable to different attacks or having correctness problems [85, 104].

In 2009, Wang et al. [99] proposed a dynamic ID based authentication scheme and claimed it to be secure against known attacks. But Wen et al. [105] demonstrated their scheme to be insecure against impersonation attack as well as offline password guessing attack. Furthermore, they proposed an improved scheme [105]. Tang et al. [106] proved that their improved scheme [105], is still vulnerable to password guessing, impersonation and insider attacks. They [106] also showed that the scheme [105] was lacking forward secrecy. Recently Chung et al. [101] described that Wang et al.'s scheme [99] is not a proper dynamic identity scheme as the real identity of user is sent in plaintext during login session. They [101] also identified that the scheme [99] is having incorrect password change phase. Furthermore, Chung et al. proposed an improved dynamic identity based authentication scheme and claimed their scheme to protect the user's anonymity as well as resisting all known attacks. Very recently, Kumari et al. [5] identified the Chung's scheme [101], to be vulnerable to impersonation attack, password guessing attack, anonymity violation attack, invalid password change phase, insider attack and lacking proper mutual authentication. Furthermore, Kumari et al. [5], proposed an improved scheme and claimed it to be secure against all known attacks. Furthermore, Kumari et al. claimed that their scheme preserves the user's anonymity. In this chapter, we analyze Kumari et al.'s scheme and find it to be vulnerable to the user anonymity violation attack and the smart card stolen attack. We show thats, a legal user can break the anonymity of another legal user. Similarly, we show that if a legal user steals smart card of another user then he can establish session and share key with the legal server on behalf of latter. Then we propose an anonymous smart card based authentication scheme using only symmetric key primitives. The proposed scheme is more secure than the related existing schemes.

Rest of the chapter is organized as follows. In Section 5.1, we review Kumari et al.'s scheme, while its cryptanalysis is performed in section 5.2. Proposed supplementary scheme is described in section 5.3. We have analyzed our scheme informally and formally using the random oracle model in section 5.4. Section 5.5 verifies the security using automated tool ProVerif. The performance comparison is performed in section 5.6. Finally, chapter's summary is solicited in Section 5.7.

Table 5.1: Notation Guide

Notations	Description	Notations	Description
$\mathcal{S}$	Server	$\mathcal{U}_i$	The legal client
ID_i	Identity of U_i	$\mathcal{A}$	The Adversary
PW_i	Password of patient U_i	k_i	Unique random number of $\mathcal{U}_i$
k_{s1}, k_{s2}	Secret keys of $\mathcal{S}$	$\|$	String concatenation operator
T_{ui}	Timestamp of $\mathcal{U}_i$	T_{si}	i^{th} timestamps of $\mathcal{S}$
$\oplus$	Bitwise XOR operation	$h(.)$	A one way hash function
PID_i	Pseudo identity of $\mathcal{U}_i$	SC_{ui}	$\mathcal{U}_i$'s smart card
$E_k(.)$	Symmetric Encryption	$D_k()$	Symmetric Decryption

5.1 Review of Kumari et al.'s Scheme

This section reviews Kumari et al.'s remote user authentication scheme [5]. In Kumari et al.'s scheme, the server $\mathcal{S}$ keeps two secret keys named as k_{s1} and k_{s2}. Then server assigns a unique secret random variable k_i to each user $\mathcal{U}_i$. Kumari et al.'s scheme is shown in Fig. 5.1. We have also illustrated the notation guide in table 5.1. We also describe their scheme in following four phases:

5.1.1 Registration Phase

Registration phase consists of three steps: initially $\mathcal{U}_i$ chooses his identity ID_i and password PW_i along with a random number c. $\mathcal{U}_i$ further computes $RP_i = h(c\|PW_i)$ and sends $\{ID_i, RP_i\}$ to $\mathcal{S}$ on a private channel. After receiving $\{ID_i, RP_i\}$ from $\mathcal{U}_i$, $\mathcal{S}$ computes $G_i = h(ID_i\|k_{s1}) \oplus RP_i$, $K_i = k_i \oplus h(ID_i\|k_{s1})$, $H_i = h(ID_i\|k_i\|RP_i)$ and $J_i = k_i \oplus h(k_{s1}\|k_{s2})$. $\mathcal{S}$ stores $\{K_i, H_i, J_i, h()\}$ into smart card SC_{ui} and sends SC_{ui} and G_i to $\mathcal{U}_i$. Upon receiving SC_{ui}, $\mathcal{U}_i$ computes $R_i = (ID_i\|PW_i) \oplus c$, $L_i = G_i \oplus c$ and inserts R_i, L_i into SC_{ui}. Now SC_{ui} contains $\{K_i, H_i, J_i, h(), R_i, L_i\}$.

5.1.2 Login Phase

During login phase $\mathcal{U}_i$ inserts his SC_{ui} into card reader, submits his identity ID_i and password PW_i. SC_{ui} performs following steps:

Step L 1: SC_{ui} computes $c = R_i \oplus (ID_i\|PW_i)$, $RP_i = h(c\|PW_i)$, $h(ID_i\|k_{s1}) = L_i \oplus RP_i \oplus c$, $k_i = K_i \oplus h(ID_i\|k_{s1})$ and $H_i^* = h(ID_i\|k_i\|RP_i)$.

Step L 2: SC_{ui} further checks $H_i \stackrel{?}{=} H_i^*$, if condition does not hold then SC_{ui} aborts the session.

Step L 3: SC_{ui} calculates $h(k_{s2}\|k_{s1}) = k_i \oplus J_i$, $G_i = L_i \oplus c$, $PID_i = ID_i \oplus h(G_i\|k_i\|T_{ui})$, $\overline{G_i} = G_i \oplus h(k_i\|T_{ui})$, $P_i = G_i \oplus RP_i = h(ID_i\|k_{s1})$, $Q_i = h(G_i\|k_i\|P_i\|T_{ui})$ and $S_i = k_i \oplus (h(k_{s2}\|k_{s1})\|T_{ui})$.

Step L 4: SC_{ui} sends $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$ to $\mathcal{S}$.

5.1.3 Authentication Phase

$\mathcal{S}$ first verifies the validity of timestamp T_{ui}, aborts the session if difference between T_{s1} and T_{ui} is greater than ΔT. Otherwise, $\mathcal{S}$ performs following steps:

Step A 1: Initially, $\mathcal{S}$ computes $k_i = S_i \oplus (h(k_{s2}\|k_{s1})\|T_{ui})$, $G_i = \overline{G_i} \oplus h(k_i\|T_{ui})$, $ID_i = PID_i \oplus h(G_i\|k_i\|T_{ui})$, $P_i^* = h(ID_i\|k_{s1})$ and $Q_i^* = h(G_i\|k_i\|P_i^*\|T_{ui})$.

Step A 2: $\mathcal{S}$ checks whether $Q_i \stackrel{?}{=} Q_i^*$, if it does not hold, $\mathcal{S}$ aborts the session. Otherwise, $\mathcal{S}$ computes $a = h(P_i^*\|k_i\|T_{s2})$. $\mathcal{S}$ further sends $\{a, T_{s2}\}$ to $\mathcal{U}_i$.

Step A 3: Upon receiving $\{a, T_{s2}\}$, $\mathcal{U}_i$ checks the validity of T_{s2}, if it is fresh then $\mathcal{U}_i$ computes $a^* = h(P_i^*\|k_i\|T_{s2})$.

Step A 4: $\mathcal{U}_i$ checks whether $a^* \stackrel{?}{=} a$, if it holds, $\mathcal{S}$ is authenticated.

Step A 5: Both $\mathcal{S}$ and $\mathcal{U}_i$ compute the shared session key as:

$$SK = h(P_i\|k_i\|T_{ui}\|T_{s2}||h(k_{s2}\|k_{s1})) \tag{5.1}$$

5.1.4 Password Change Phase

The password change phase is carried out without intervention of $\mathcal{S}$. To change password, $\mathcal{U}_i$ inserts SC_{ui} into card reader and enters his password PW_i and ID_i.

Step PC 1: To verify ID_i and PW_i, SC_{ui} computes $c = R_i \oplus (ID_i\|PW_i)$, $RP_i = h(c\|PW_i)$, $h(ID_i\|k_{s1}) = L_i \oplus RP_i \oplus c$, $k_i = K_i \oplus h(ID_i\|k_{s1})$ and $H_i^* = h(ID_i\|k_i\|RP_i)$.

Step PC 2: SC_{ui} further checks $H_i^* \stackrel{?}{=} H_i$, if true, SC_{ui} asks $\mathcal{U}_i$ to enter new password.

Step PC 3: $\mathcal{U}_i$ submits new password $PW_{i_{new}}$. SC_{ui} computes $RP_{i_{new}} = h(c\|PW_{i_{new}})$, $R_{i_{new}} = (ID_i\|PW_{i_{new}}) \oplus c$, $L_{i_{new}} = L_i \oplus RP_i \oplus RP_{i_{new}}$ and $D_{i_{new}} = h(ID_i\|k_i\|RP_{i_{new}})$.

Step PC 4: SC_{ui} replaces R_i, D_i, L_i, RP_i with new values $R_{i_{new}}$, $D_{i_{new}}$, $L_{i_{new}}$, $RP_{i_{new}}$.

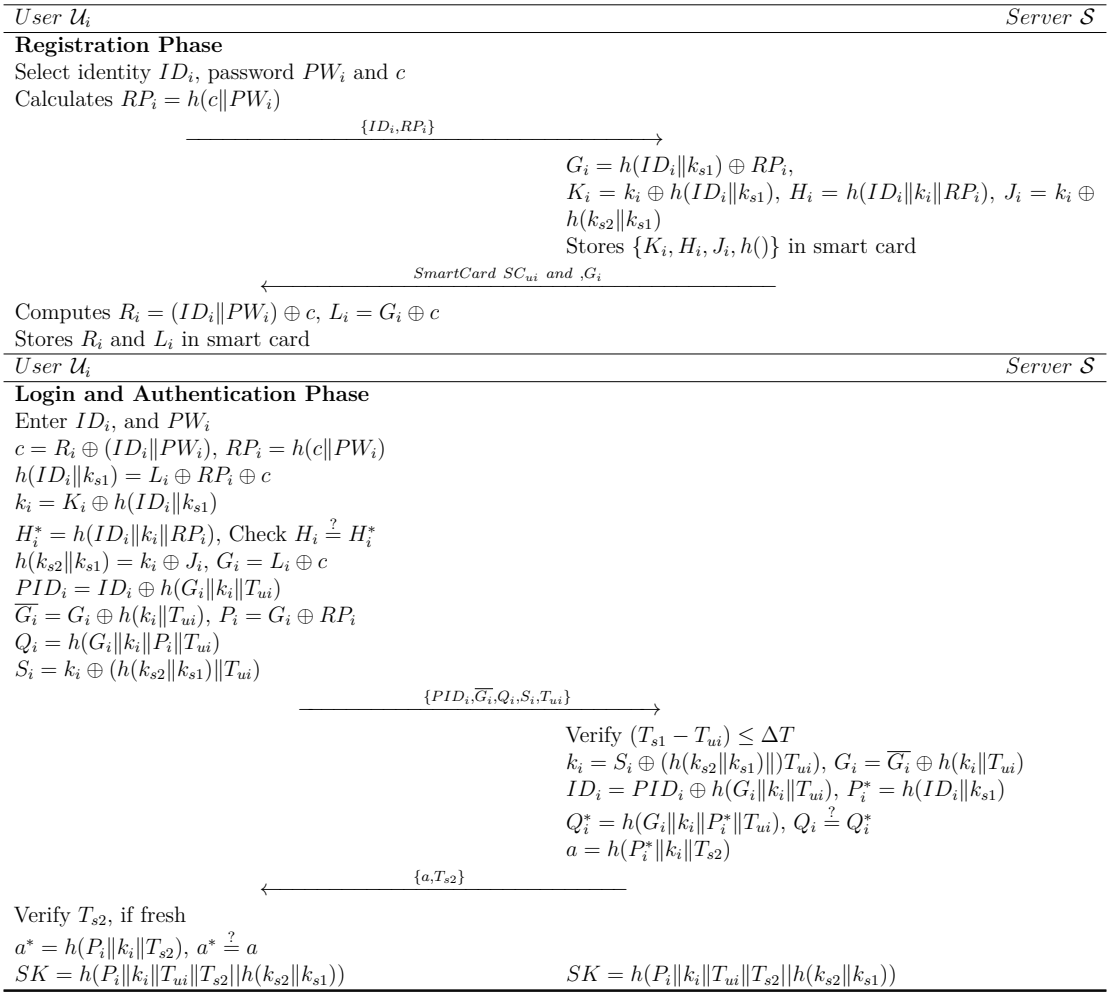


Figure 5.1: Kumari et al.'s Scheme

5.2 Cryptanalysis of Kumari et al.'s Scheme

This section defines the verdict that Kumari et al.'s scheme is vulnerable to user anonymity violation attack and smart card stolen attack. Before proceeding further, three common assumptions are made as follows:

1. An adversary $\mathcal{A}$ is having full control over public communication channel. $\mathcal{A}$ can intercept, modify, insert or delete any message.
2. $\mathcal{A}$ can steal $\mathcal{U}_i$'s smart card or get $\mathcal{U}_i$'s password but not both simultaneously.
3. Any one having possession of a smart card can extract information stored in that smart card [28, 29].

5.2.1 User anonymity violation attack

In current era of pervasive computing, user's personal information can be accessed by an adversary by analyzing the session information. In wireless communication, the adversary may become able to find the current location of a user or his moving history. An authentication scheme is said to provide anonymity if it can achieve two main goals: (1) real identity of user is not revealed to adversary and (2) the adversary cannot determine, either two different sessions are initiated by same user. In order to achieve both above mentioned goals Kumari et al.'s scheme employed dynamic ID technique. We show that the dynamic ID employed by Kumari et al. does not achieve both mentioned goals related to anonymity. A legal user $\mathcal{U}_j$ can break anonymity of another legal user $\mathcal{U}_i$ by performing the following steps:

Step AV 1: $\mathcal{U}_j$ extracts the information $\{K_j, H_j, J_j, h(), R_j, L_j\}$ stored on his smart card SC_{uj}, then computes following:

$$c = R_j \oplus (ID_j \| PW_j) \tag{5.2}$$

$$RP_j = h(c \| PW_j) \tag{5.3}$$

$$h(ID_j \| k_{s1}) = L_j \oplus RP_j \oplus c \tag{5.4}$$

$$k_j = K_j \oplus h(ID_j \| k_{s1}) \tag{5.5}$$

$$H_j^* = h(ID_j \| k_j \| RP_j) \tag{5.6}$$

Step AV 2: $\mathcal{U}_j$ further computes:

$$h(k_{s2}\|k_{s1}) = k_j \oplus J_j \tag{5.7}$$

Step AV 3: After computation of $h(k_{s2}\|k_{s1})$, $\mathcal{U}_j$ waits for $\mathcal{U}_i$ to initiate login and authentication request.

Step AV 4: When $\mathcal{U}_i$ initiates the login and authentication request by sending $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$ to $\mathcal{S}$. $\mathcal{U}_j$ intercepts the message and calculates:

$$k_i = S_i \oplus (h(k_{s2}\|k_{s1})\|T_{ui}) \tag{5.8}$$

$$G_i = \overline{G_i} \oplus h(k_i\|T_{ui}) \tag{5.9}$$

$$ID_i = PID_i \oplus h(G_i\|k_i\|T_{ui}) \tag{5.10}$$

In Eq. 5.10, ID_i is the real identity of $\mathcal{U}_i$. Hence $\mathcal{U}_j$ has successfully breached the anonymity of $\mathcal{U}_i$.

5.2.2 Smart card stolen attack

This section describes that Kumari et al.'s scheme is vulnerable to smart card stolen attack. A legal user $\mathcal{U}_j$ can impersonate as another legal user $\mathcal{U}_i$, if he becomes able to steal $\mathcal{U}_i$'s smart card. After possession of $\mathcal{U}_i$'s smart card, $\mathcal{U}_j$ performs following steps:

Step SC 1: Firstly, $\mathcal{U}_j$ calculates $h(k_{s2}\|k_{s1})$ from his own smart card and ID_i of remote user $\mathcal{U}_i$ after intercepting $\mathcal{U}_i$'s login and authentication request, as mentioned in subsection 5.2.1.

Step SC 2: $\mathcal{U}_j$ further calculates:

$$k_i = J_i \oplus h(k_{s2}\|k_{s1}) \tag{5.11}$$

$$P_i = K_i \oplus k_i \tag{5.12}$$

Step SC 3: $\mathcal{U}_j$ selects a random number G_i and computes:

$$\overline{G_i} = G_i \oplus h(k_i \| T_{ui}) \tag{5.13}$$
$$Q_i = h(G_i \| k_i \| P_i \| T_{ui}) \tag{5.14}$$
$$S_i = k_i \oplus (h(k_{s2} \| k_{s1}) \| T_{ui}) \tag{5.15}$$
$$PID_i = ID_i \oplus h(G_i \| k_i \| T_{ui}) \tag{5.16}$$

Step SC 4: $\mathcal{U}_j$ sends $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$ to $\mathcal{S}$.

Step SC 5: Upon receiving $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$ from $\mathcal{U}_j$, $\mathcal{S}$ first verifies the timestamp then performs the following steps:

$$k_i = S_i \oplus (h(k_{s2} \| k_{s1}) \| T_{ui}) \tag{5.17}$$
$$G_i = \overline{G_i} \oplus h(k_i \| T_{ui}) \tag{5.18}$$
$$ID_i = PID_i \oplus h(G_i \| k_i \| T_{ui}) \tag{5.19}$$
$$P_i^* = h(ID_i \| k_{s1}) \tag{5.20}$$
$$Q_i^* = h(G_i \| k_i \| P_i^* \| T_{ui}) \tag{5.21}$$

Step SC 6: $\mathcal{S}$ checks whether $Q_i \stackrel{?}{=} Q_i^*$,if it does not hold, $\mathcal{S}$ aborts the session. Otherwise, computes:

$$a = h(P_i^* \| k_i \| T_{s2}) \tag{5.22}$$

Step SC 7: $\mathcal{S}$ sends $\{a, T_{s2}\}$ to $\mathcal{U}_i$.

Step SC 8: $\mathcal{U}_j$ intercepts the message and calculate $a^* = h(P_i \| k_i \| T_{s2})$. Finally, both $\mathcal{U}_j$ and $\mathcal{S}$ computes the session key as follows:

$$SK = h(P_i \| k_i \| T_{ui} \| T_{s2} || h(k_{s2} \| k_{s1})) \tag{5.23}$$

Hence, $\mathcal{U}_j$ after stealing SC_{ui} successfully shared the session key with $\mathcal{S}$ on behalf of $\mathcal{U}_i$. Therefore, it has been shown that Kumari et al.'s scheme is vulnerable to smart card stolen attack.

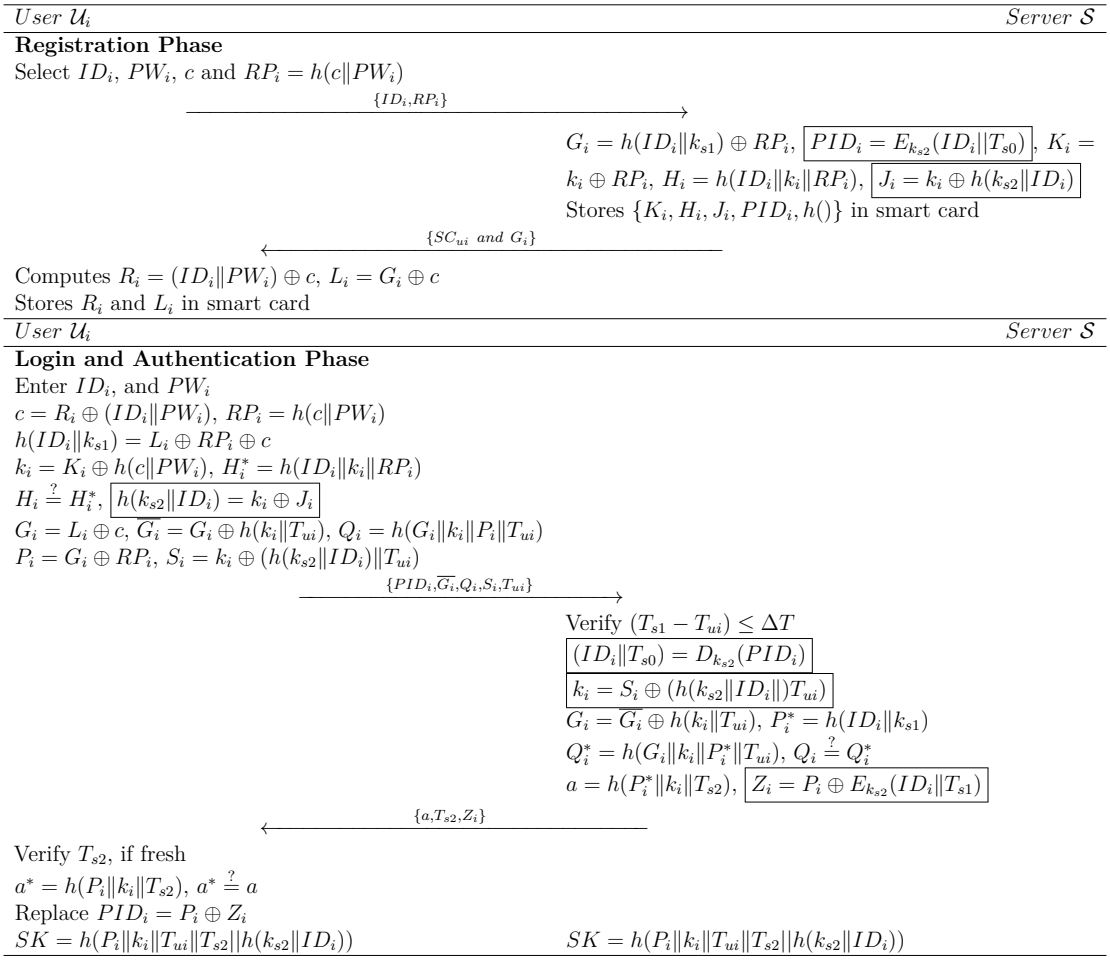


Figure 5.2: Proposed Scheme

5.3 Proposed Scheme

In this section, we elaborate the new enhanced scheme based on Kumari et al.'s scheme. The enhanced scheme is not only robust against all known attacks but also preserves the original merits of Kumari et al.'s scheme which specifically includes the lightweightness and no verifier tables stored on server. Like Kumari et al.'s scheme, the proposed scheme is also having three phases: the registration phase, login and authentication phase and password change phase. We have only modified the registration phase, login and authentication phase while the password change phase is as it is taken from Kumari et al.'s scheme. The proposed scheme is illustrated in Fig. 5.2 and explained in the following subsections:

5.3.1 Registration Phase

When $\mathcal{U}_i$ wants to register with $\mathcal{S}$, the operation performed by both $\mathcal{U}_i$ and $\mathcal{S}$ are as follows:

Step PR 1: $\mathcal{U}_i \rightarrow \mathcal{S} : \{ID_i, RP_i\}$
$\mathcal{U}_i$ selects ID_i, PW_i and a random number c, and computes $RP_i = h(c\|ID_i)$. Then it sends $\{ID_i, RP_i\}$ to $\mathcal{S}$ on a private channel.

Step PR 2: $\mathcal{S} \rightarrow \mathcal{U}_i : \{SC_{ui}, G_i\}$
$\mathcal{S}$ calculates pseudo identity $PID_i = E_{k_{s2}}(ID_i||T_{s0})$ for $\mathcal{U}_i$ then $\mathcal{S}$ further computes $G_i = h(ID_i\|k_{s1}) \oplus RP_i$, $K_i = k_i \oplus RP_i$, $H_i = h(ID_i\|k_i\|RP_i)$, and $J_i = k_i \oplus h(k_{s2}\|ID_i)$. $\mathcal{S}$ further stores $\{K_i, H_i, J_i, PID_i, h()\}$ in smart card SC_{ui} and sends SC_{ui} & G_i to $\mathcal{U}_i$ via some secure channel.

Step PR 3: Upon receiving $\{SC_{ui}, G_i\}$, $\mathcal{U}_i$ calculates $R_i = (ID_i\|PW_i) \oplus c$, $L_i = G_i \oplus c$ and stores both of these in SC_{ui}. Finally, the smart card SC_{ui} contains $\{K_i, H_i, J_i, PID_i, h(), R_i, L_i\}$.

5.3.2 Login and Authentication Phase

When $\mathcal{U}_i$ wants to login to remote server, he inserts SC_{ui} in card reader then inputs ID_i and PW_i. SC_{ui} and $\mathcal{S}$ performs following steps:

Step PL 1: SC_{ui} calculates $c = R_i \oplus (ID_i\|PW_i)$, $RP_i = h(c\|PW_i)$, $h(ID_i\|k_{s1}) = L_i \oplus RP_i \oplus c$, $k_i = K_i \oplus h(c\|PW_i)$ and $H_i^* = h(ID_i\|k_i\|RP_i)$.

Step PL 2: SC_{ui} checks $H_i \stackrel{?}{=} H_i^*$, if not true, the session is aborted by SC_{ui}.

Step PL 3: $SC_{ui} \rightarrow \mathcal{S} : \{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$
SC_{ui} computes $h(k_{s2}\|ID_i) = k_i \oplus J_i$, $G_i = L_i \oplus c$, $\overline{G_i} = G_i \oplus h(k_i\|T_{ui})$, $Q_i = h(G_i\|k_i\|P_i\|T_{ui})$, $P_i = G_i \oplus RP_i$ and $S_i = k_i \oplus (h(k_{s2}\|ID_i)\|T_{ui})$. Then SC_{ui} sends authentication request message $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$ to $\mathcal{S}$.

Step PL 4: After receiving authentication request message $\mathcal{S}$ first verifies the validity of T_{ui} then computes: $(ID_i\|T_{s0}) = D_{k_{s2}}(PID_i)$, $k_i = S_i \oplus (h(k_{s2}\|ID_i\|)T_{ui})$, $G_i = \overline{G_i} \oplus h(k_i\|T_{ui})$, $P_i^* = h(ID_i\|k_{s1})$, $Q_i^* = h(G_i\|k_i\|P_i^*\|T_{ui})$

Step PL 5: $\mathcal{S}$ checks $Q_i \stackrel{?}{=} Q_i^*$, if true, $\mathcal{U}_i$ is authenticated by $\mathcal{S}$.

Step PL 6: $\mathcal{S} \rightarrow \mathcal{U}_i : \{a, T_{s2}, Z_i\}$
$\mathcal{S}$ then computes $a = h(P_i^*\|k_i\|T_{s2})$, and $Z_i = P_i \oplus E_{k_{s2}}(ID_i\|T_{s1})$. Further, $\mathcal{S}$ sends

$\{a, T_{s2}, Z_i\}$ to $\mathcal{U}_i$.

Step PL 7: After receiving $\{a, T_{s2}, Z_i\}$ from $\mathcal{S}$, $\mathcal{U}_i$ verifies T_{s2} and computes $a^* = h(P_i \| k_i \| T_{s2})$ and compare it with $\mathcal{S}$'s signature a. If both are equal, $\mathcal{S}$ is treated as a legal server by $\mathcal{U}_i$.

Step PL 8: Both $\mathcal{S}$ and $\mathcal{U}_i$ compute the shared key as

$$SK = h(P_i \| k_i \| T_{ui} \| T_{s2} || h(k_{s2} \| ID_i)) \tag{5.24}$$

5.4 Security Analysis

In this section, we perform the informal as well as formal security analysis of our proposed scheme. We show that the proposed scheme is robust against known attacks which is evident from following subsections:

5.4.1 Informal Security Analysis

In this section, we analyze the security and correctness of proposed scheme under the same assumptions as discussed in section 5.2. Our analysis shows that the proposed scheme is robust against all known attacks, while slightly burdening the computation, communication and storing an extra parameter in the smart card. The main problem with Kumari et al.'s scheme was the use of $h(k_{s2} \| k_{s1})$ which can be computed by any legal user. Further, user specific secret k_i can be calculated by the use of $h(k_{s2} \| k_{s1})$ which ultimately results in user anonymity violation and smart card stolen attacks. Therefore, we use $h(k_{s2} \| ID_i)$ instead of $h(k_{s2} \| k_{s1})$, to make each computation user specific, and the pseudo identity PID_i of user is calculated by server at registration and during each authentication session. Table 5.2 summarizes the security analysis of proposed scheme with scheme's of Kumari et al., Chung et al., An and Wen et al. It is evident from the results that the proposed scheme resists all known attacks while all other schemes are vulnerable to user anonymity attack. Moreover, Kumari et al.'s scheme is vulnerable to smart card lost/stolen attack. Chung et al.'s scheme is vulnerable to insider, smart card stolen, impersonation, offline password guessing and DoS attacks. Furthermore, it does not provide proper mutual authentication and secure session key. An's scheme is vulnerable to DoS attack and lacking proper mutual authentication. The scheme of Wen et al. does not resist impersonation, offline password guessing and DoS

attacks. Furthermore, Wen et al.'s scheme does not provide forward secrecy, proper mutual authentication and secure session key.

5.4.1.1 Privileged Insider Attack

During registration phase, ID_i and $RP_i = h(c\|PW_i)$ are sent to $\mathcal{S}$, where password PW_i and c are protected by one way hash function. It is not possible for an insider to compute two values protected by hash function in polynomial time. Similarly, during login and authentication phase PW_i and c are not revealed to $\mathcal{S}$. Hence, the proposed scheme resists privileged insider attack.

5.4.1.2 Smart Card Lost/Stolen Attack

An adversary $\mathcal{A}$, whether a legal user or an outsider can steal $\mathcal{U}_i$'s smart card. Further $\mathcal{A}$ can get the parameters $\{K_i, H_i, J_i, PID_i, R_i, L_i\}$ stored on smart card. If $\mathcal{A}$ is a legal user then he can compute $h(k_{s2}\|ID_a)$ from his own smart card by following the method described in subsection 5.2.1. As we have modified the value of J_a to contain $h(k_{s2}\|ID_a)$ instead of $h(k_{s2}\|k_{s1})$. Therefore, the computation of $h(k_{s2}\|ID_a)$ is useless for $\mathcal{A}$ to find secret number k_i of $\mathcal{U}_i$, which indeed requires the knowledge of $h(k_{s2}\|ID_i)$. Furthermore, $\mathcal{A}$ can get k_i either from $K_i = k_i \oplus RP_i$ or $J_i = k_i \oplus h(k_{s2}\|ID_i)$. In-order to retrieve k_i from K_i, $\mathcal{A}$ needs to know RP_i, which can only be calculated by PW_i exclusively known to $\mathcal{U}_i$. For computing k_i from J_i, $\mathcal{A}$ should have the knowledge of $h(k_{s2}\|ID_i)$, which can be calculated by first getting k_i. Hence, the smart card contains no useful information for $\mathcal{A}$. Therefore, the lost/stolen smart card is having no bitter effects on the security of the proposed scheme.

5.4.1.3 User Anonymity Violation Attack

User anonymity is an important parameter while designing an authentication scheme. If anonymity is revealed to an adversary, he can access user's personal sensitive information like: preferences, social circle, current location, moving history etc. [84]. In registration phase of proposed scheme, the server $\mathcal{S}$ computes pseudo identity $PID_i = E_{k_{s2}}(ID_i||T_{s0})$ of $\mathcal{U}_i$ by encrypting ID_i concatenated by current timestamp. Moreover, during each successful authentication session $\mathcal{S}$ computes $\mathcal{U}_i$'s new pseudo identity PID_i concatenated with $\mathcal{S}$'s new timestamp and then encrypted by his own secret key k_{s2}. After this, $\mathcal{S}$ sends $Z_i = P_i \oplus E_{k_{s2}}(ID_i\|T_{s1})$. It can be clearly seen that $\mathcal{U}_i$'s pseudo identity is not sent in plaintext, but it is protected by bitwise exclusive-or with P_i. Upon receiving the message, $\mathcal{U}_i$ replaces

the previous PID_i with the received PID_i. The real identity can only be revealed to $\mathcal{A}$, if he can access $\mathcal{S}$'s secret key k_{s2}. Furthermore, the pseudo identity passes both requirements of anonymity: which are (i)the real identity is not revealed to $\mathcal{A}$ and (ii) no adversary can judge that two different sessions are initiated by same user, which is because of the dynamicity of pseudo identity.

5.4.1.4 User and Server Impersonation Attacks

An adversary $\mathcal{A}$ can impersonate as a legal user, if he is able to generate a valid login message. In proposed scheme the valid login message can only be generated by computing $H_i = h(ID_i \| k_i \| RP_i)$ which can only be generated by first knowing $\mathcal{U}_i$'s password PW_i and secret k_i. Similarly, if $\mathcal{A}$ wants to impersonate as a legal user $\mathcal{U}_i$ directly by sending authentication message to $\mathcal{S}$ then $\mathcal{A}$ has to calculate $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$. PID_i is the dynamic identity of $\mathcal{U}_i$, which is different in each session and is encrypted by $\mathcal{S}$'s secret key. Similarly, $\overline{G_i}, Q_i, S_i$ can only be computed by legal user $\mathcal{U}_i$. $\mathcal{A}$ can impersonate as a legal server $\mathcal{S}$ if he is able to generate $\mathcal{S}$'s valid signatures a, which can only be computed after calculation of $P_i = h(ID_i \| k_{s1})$ and k_i. Both of these values require $\mathcal{S}$'s secret keys k_{s1} and k_{s2}. Hence, an adversary cannot be impersonated as a legal user or as a legal server.

5.4.1.5 Online Password Guessing Attack

The inbuilt smart card login verification method is provisioned with a limited number of login attempts with wrong password and identity. After such wrong attempts the smart card gets blocked and asks for server intervention to unblock and re-activation.

5.4.1.6 Offline Password Guessing Attack

The smart card contains $\{K_i, H_i, J_i, PID_i, h(), R_i, L_i\}$ stored in its memory which can be revealed to an adversary if smart card is lost or stolen. Out of all these parameters only $R_i = (ID_i \| PW_i) \oplus c$, $K_i = k_i \oplus RP_i$,$H_i = h(ID_i \| k_i \| RP_i)$ contains $\mathcal{U}_i$'s password. The computation of PW_i from any of these parameters requires at least guessing three unknown values, which is not possible in polynomial time.

5.4.1.7 Replay Attack

When $\mathcal{S}$ receives authentication request it first checks the validity of $\mathcal{U}_i$'s timestamp. If timestamp is not valid then $\mathcal{S}$ aborts the session. Furthermore, the same timestamp is also embedded in $\mathcal{U}_i$'s signatures $\overline{G_i}, Q_i$ and S_i. So, if an adversary $\mathcal{A}$ replays a previous message, $\mathcal{S}$ can easily detect it and aborts the session. Similarly, $\mathcal{A}$ cannot replay $\mathcal{S}$'s reply message as it contains current timestamp T_{s2} and server signature $a = h(P_i^* \| k_i \| T_{s2})$, which is unique for each session as it contains new timestamp in each session. Hence, the proposed scheme resists the replay attack.

5.4.1.8 Denial of Services Attack

In proposed scheme the smart card contains inbuilt mechanism to verify the legality of a user. $\mathcal{U}_i$ submits his password and identity. Smart card then verifies the correctness of identity and password. If any of these two is wrong the smart card aborts the session. The login and authentication request is only send to $\mathcal{S}$, if $\mathcal{U}_i$ has been authenticated by smart card. Therefore, the proposed scheme resists denial of services attack.

5.4.1.9 Perfect Forward Secrecy

Forward secrecy ensures that if a session key or long term private key or password of any of the participants is disclosed then the secrecy of previous session keys remains intact. In proposed scheme each session key $SK = h(P_i \| k_i \| T_{ui} \| T_{s2} || h(k_{s2} \| ID_i))$ contains $\mathcal{U}_i$'s current timestamp T_{ui}, as well as $\mathcal{S}$'s timestamp T_{s2} along with secret number k_i, $\mathcal{U}_i$'s signature P_i and ID_i. Hence, even if long term private key of the server or user password is compromised, it will not provide aid to compute previous session keys.

5.4.1.10 Stolen Verifier Attack

In proposed scheme, the server does not maintain any verifier table to store user's password or other sensitive information. $\mathcal{S}$ computes $\mathcal{U}_i$'s ID_i, secret k_i, and P_i using his own secret keys and $\mathcal{U}_i$'s ID_i. Hence, no stolen verifier attack is possible on proposed scheme.

Table 5.2: Comparison of Security parameters

Scheme:	Proposed	[5]	[101]	[107]	[105]
Resists Insider attack	Yes	Yes	No	Yes	No
Resists Smart card lost attack	Yes	No	No	Yes	Yes
Resists User anonymity violation attack	Yes	No	No	No	No
Resists Impersonation attack	Yes	Yes	No	Yes	No
Resists Online password guessing attack	Yes	Yes	Yes	Yes	Yes
Resists Offline password guessing attack	Yes	Yes	No	Yes	No
Resists Replay attack	Yes	Yes	Yes	Yes	Yes
Resists DoS attack	Yes	Yes	No	No	No
Resists Stolen verifier attack	Yes	Yes	Yes	Yes	Yes
provides Forward secrecy	Yes	Yes	Yes	Yes	No
Provides Proper Mutual authentication	Yes	Yes	No	No	No
Provides Secure session key	Yes	Yes	No	Yes	No

5.4.2 Formal Security Analysis

In this section, we prove the security of our protocol in random oracle model. We start with formal security model and assumptions used in our proof.

5.4.2.1 Security model

To verify the resistance of proposed protocol against known attacks, we proceed using provable security. The adopted model is as follows:

- **Participants** A network having a number of interconnected participants is simulated in an authentication protocol Π. Each participant in the network is either a trusted server $S \in \mathcal{S}$ or a user $U \in \mathcal{U}$. There may be several instances of each participant termed as oracles and each of the oracles is involved in a distinct execution of Π. Referring to U's i-th instance (resp. S) in a session as Π_U^i (resp. Π_S^i). Π_U^i (resp. Π_S^j) is associated with mate ID pid_U^i (resp:pid_S^j), along with session ID sid_U^i (resp:sid_S^j), and a session key sk_U^i. pid_U^i (resp:pid_S^j). pid_U^i (resp:pid_S^j) represents the set of involved identities in the referred instance while sid_U^i (resp:sid_S^j) symbolizes the flows sent and received by Π_U^i (resp. Π_S^j). Π_U^i (resp. Π_S^i) is presumed to be *accepted*, if it griped the key sk_U^i (resp:sk_S^j). The identifiers sid_U^i (resp:sid_S^j), pid_U^i (resp:pid_S^j). Π_U^i and Π_S^j are said to be *partnered* if (1) both are accepted, (2) $pid_U^i = pid_S^j$, (3) $sid_U^i = sid_S^j$ and (4) $sk_U^i = sk_S^j$.
- **Long-lived keys** Each $U \in \mathcal{U}$ possesses a password PW_U, while each $S \in \mathcal{S}$ holds a vector $PW_S = \langle pw_U \rangle_{U \in \mathcal{U}}$ with an entry corresponding to each user.
- **Adversary model** An adversary $\mathcal{A}$ is assumed to fully control the channel. $\mathcal{A}$ plans and intercedes the sessions among communicating parties. $\mathcal{A}$ can execute succeeding

queries in any order:

Execute(Π_U^i, Π_S^j): This query enables $\mathcal{A}$ to perform passive attacks. This query is executed to eavesdrops on the honest executions among Π_U^i and Π_S^j by $\mathcal{A}$. It outputs the exchanged messages among participants.

SendClient(Π_U^i, m): This query provides $\mathcal{A}$ the facility to perform active attacks, where $\mathcal{A}$ intercepts and then modifies a message, generates a new one, or just forwards it to the Π_U^i. This query outputs the message generated by Π_U^i on receiving message m. $\mathcal{A}$ can also pledge Π by executing SendClient(Π_U^i, $Start$).

SendServer(Π_S^i, m): This query enables $\mathcal{A}$ to execute an active attack against an $S \in \mathcal{S}$. $\mathcal{A}$ performs it to acquire the message generated by Π_S^i upon reception of the message m.

Reveal(Π_U^i): By simulating this query $\mathcal{A}$ can obtain the session key of Π_U^i.

Corrupt(U): This query outputs the long lived key pw_U of participant U.

Test(Π_U^i): $\mathcal{A}$ can execute only one such query to a fresh oracle. It responses into a random bit $b \in \{0, 1\}$, if $b = 1$, then it returns the session key of Π_U^i. Otherwise, the query returns a random value.

- **fresh oracle** An oracle Π_U^i is said to be fresh if and only if: (1) Π_U^i is accepted, and (2) Reveal query is not invoked by Π_U^i or its partner after its acceptance.
- **Protocol Security** The security of Π is demonstrated by a game $Game(\Pi, \mathcal{A})$. During simulation of this game, $\mathcal{A}$ can execute a number of mentioned queries to Π_U^i and Π_S^j. If $\mathcal{A}$ asks a query Test(Π_U^i) and Π_U^i has *accepted* it and it is *fresh*, then $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ tries to guess b correctly. The advantage of $\mathcal{A}$ is defined as follows:

$$Advt_{\Pi,UD}(\mathcal{A}) = |2Pr[b' = b] - 1|.$$

Π is said to be secured if $Advt_{\Pi,D}(\mathcal{A})$ is negligible.

5.4.2.2 Security proof

Theorem 3. *UD is defined as a uniformly distributed dictionary of all possible passwords with size $|UD|$ and Π describes the improved authentication protocol. Suppose that hash*

function h is modeled as a random oracle. Then,

$$Advt_{\Pi,UD}(\mathcal{A}) \leq \frac{q_{hs}^2 + (q_{sd} + q_{ee})^2}{2^{ln}} + \frac{q_{hs}}{2^{ln}} + \frac{q_{sd}}{|UD|},$$

where q_{sd} denotes total Send *queries; q_{ee} the* Execute *queries and q_{hs} represents total number of hash queries to h.*

Proof. The proof consists of a game fusion, initiating by G_0 and terminating at G_3, while $\mathcal{A}$ is having no advantage. For each $G_x(0 \leq x \leq 3)$, $Succ_x$ is defined as an event that $\mathcal{A}$ guesses b correctly in test session.

Game G_0. In this game, all $U \in \mathcal{U}$ and $S \in \mathcal{S}$ are simulated in random oracle. By definition of event $Succ_x$ which means that $\mathcal{A}$ guesses b correctly in Test-query, we have:

$$Advt_{\Pi,D}(\mathcal{A}) = 2|\Pr[Succ_0] - \frac{1}{2}|. \tag{5.25}$$

Game G_1. It is the same game as of G_0 except the oracle h maintains a hash list h_{Llist}, where the records in h_{Llist} are of the form (IP, OP). G_1 returns OP, if a record (IP, OP) exists in h_{Llist}. Otherwise a random chosen $OP \in \{0,1\}^{ln}$ is sent to $\mathcal{A}$ and keeps new tuple (IP, OP) in h_{Llist}. All the user and server instances are simulated for Send, Execute, SendClient, SendServer, Reveal, Corrupt and Test queries. It is easily verifiable that game is perfectly indistinguishable from real attack. Hence, we have:

$$\Pr[Succ_1] = \Pr[Succ_0]. \tag{5.26}$$

Game G_2. This game involves simulation of all oracles in G_1. In addition, this game is canceled upon occurrence of collision on hash value h and partial transcripts S_i and a. Referring birthday paradox, the maximum collisions probability in output of hash oracles is maximum $q_{hs}^2/2^{ln+1}$, where q_{hs} is the maximum total of hash queries. Likewise, the maximum collision probability in transcripts is $(q_{sd} + q_{ee})^2/2^{ln+1}$ where as q_{sd} be the queries to Send oracle and q_{ee} be the queries to Execute oracle, and ln denotes bit length of the random numbers and the output of the hash function. So we have:

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{q_{hs}^2 + (q_{sd} + q_{ee})^2}{2^{ln+1}}. \tag{5.27}$$

Game G_3. For this game, the simulation of queris to SendClient oracle is again changed for

selected session in G_2. The computation of SK is amended to make it independent of password and related keys. When Send (Π_U^i, $\{a, T_{s2}, Z_i\}$) and Send (Π_S^j, $\{PID_i, \overline{G_i}, Q_i, S_i, T_{ui}\}$) are asked. We set $SK = h(P_i \| w \| T_{ui} \| T_{s2} || h(k_{s2} \| ID_i))$, where w is selected at random. The two possible cases where G_2 and G_3 are distinguishable as follows:

Case 1. $\mathcal{A}$ queries $(P_i \| w \| T_{ui} \| T_{s2} || h(k_{s2} \| ID_i))$ to h, the occurrence probability of this event is $q_{hs}/2^{ln}$.

Case 2. $\mathcal{A}$ asks Send query except Send(Π_U^i, $\{a, T_{s2}, Z_i\}$) and successfully impersonates U. $\mathcal{A}$ is not allowed to reveal static key PW_U. Thus, in order to impersonate U, the $\mathcal{A}$ has to get some password PW_U's information whose probability is $1/|UD|$, as at most there are q_{sd} such sessions, the occurrence probability of this event is less than $q_{sd}/|UD|$

The difference between G_2 and G_3 is as follows:

$$|\Pr[Succ_3] - \Pr[Succ_2]| \leq \frac{q_{hs}}{2^{ln}} + \frac{q_{sd}}{|UD|}. \tag{5.28}$$

On the other hand,

$$\Pr[Succ_3] = \frac{1}{2}. \tag{5.29}$$

Combining the equations Eqs. (5.25), (5.26), (5.27), (5.28) and (5.29), the result is as follows:

$$\begin{aligned} Advt_{\Pi,UD}(\mathcal{A}) &= 2|\Pr[Succ_0] - \frac{1}{2}| \\ &= 2|\Pr[Succ_0] - \Pr[Succ_3]| \\ &\leq 2(|\Pr[Succ_1] - \Pr[Succ_2] + \Pr[Succ_2] - \\ &\quad \Pr[Succ_3]| \\ &\leq \frac{q_{hs}^2 + (q_{sd} + q_{ee})^2}{2^{ln}} + \frac{q_{hs}}{2^{ln}} + \frac{q_{sd}}{|UD|}. \end{aligned}$$

□

5.5 Protocol verification through ProVerif

We have modeled the steps illustrated in subsection 5.3.2 and shown in Fig. 5.2. The modeled code in ProVerif is shown in Fig 5.3.

The verification is performed on ProVerif 1.88 (latest version), the results are as follows:

```
(***************** Channels *****************)
free insCh:channel.
free sCh:channel.
(********* Constants & Variables **********)
const PWi:bitstring [private].
(************** Constructors **************)
fun owh(bitstring):bitstring.
fun conncat(bitstring,bitstring):bitstring.
fun Exr(bitstring,bitstring):bitstring.
fun SyE(bitstring,bitstring):bitstring.
(******** Destructors & Equations *********)
reduc forall m:bitstring,key:bitstring; SyD(SyE
    (m,key),key)=m.
equation forall a:bitstring,b:bitstring; Exr(
    Exr(a,b),b)=a.
(***************** Events *****************)
event iniUser(bitstring).
event terminateUser(bitstring).
event iniServer(bitstring).
event terminateServer(bitstring).
```

(a) Declarations

```
(***************** processes ***************)
(****************** User ******************)
(*Processes*)
let UserProcess=
new IDi:bitstring;
new c:bitstring;
out(sCh,(IDi,owh(conncat(c,PWi))));
in(sCh,(yKi:bitstring,yHi:bitstring,yJi:
    bitstring, yPIDi:bitstring,yGi:bitstring))
    ;
event iniUser(IDi);
let Ri= Exr(conncat(IDi,PWi),c) in
let Li= Exr(yGi,c) in
(*Login*)
let C = Exr(Ri,conncat(IDi,PWi)) in
let RPI = owh(conncat(C,PWi)) in
let ki = Exr(yKi,owh(conncat(C,PWi))) in
let Hi' = owh(conncat(IDi,conncat(ki,RPI))) in
if yHi=Hi' then
let Gi = Exr(Li,C) in
new Tui:bitstring;
let GI = Exr(Gi,owh(conncat(ki,Tui))) in
let Pi = Exr(Gi,RPI) in
let Qi = owh(conncat(Gi,conncat(ki,conncat(Pi,
    Tui)))) in
let Si = Exr(ki,conncat(Exr(ki,yJi),Tui)) in
out(insCh,(yPIDi,GI,Qi,Si,Tui));
in(insCh,((ya:bitstring,yTs2:bitstring,yZi:
    bitstring)));
let a'=owh(conncat(Pi,conncat(ki,yTs2))) in
if a' = ya then
let sk=owh(conncat(Pi,conncat(ki,conncat(Tui,
    conncat(yTs2,owh(Exr(ki,yJi))))))) in
event terminateUser(IDi).
(*************** Server S ***************)
let ServerProcess=
new x:bitstring;
in(sCh,(xIDi:bitstring,xRPi:bitstring));
new Ts0:bitstring;
new ks1:bitstring;
new ks2:bitstring;
let Gi = Exr(owh(conncat(xIDi,ks1)),xRPi) in
let PIDi = SyE(conncat(xIDi,Ts0),ks2) in
new ki:bitstring;
let Ki = Exr(ki,xRPi) in
let Hi = owh(conncat(xIDi,conncat(ki,xRPi))) in
let Ji = Exr(ki,owh(conncat(ks2,xIDi))) in
out(sCh,(Ki,Hi,Ji,PIDi,Gi));
(*Login*)
event iniServer(xIDi);
in(insCh,(xyPIDi:bitstring,xGI:bitstring,xQi:
    bitstring,xSi:bitstring,xTui:bitstring));
let ki'=Exr(xSi,conncat(Exr(ki,Ji),xTui)) in
let Gi' = Exr(xGI,owh(conncat(ki,xTui))) in
let pi' = owh(conncat(xIDi,ks1)) in
let Qi' = owh(conncat(Gi,conncat(ki',conncat(pi
    ',xTui)))) in
if xQi=Qi' then
new Ts1:bitstring;
new Ts2: bitstring;
let a = owh(conncat(pi',conncat(ki',Ts2))) in
let Zi = Exr(pi',SyE(conncat(xIDi,Ts1),ks2)) in
out(insCh,(a,Ts2,Zi));
event terminateServer(xIDi).
```

(a) Processes

```
(***************** Main *****************)
process (!UserProcess) | (!ServerProcess))
free sk:bitstring [private].
query attacker(sk).
query id:bitstring; inj event(terminateUser(id)
    ) ==> inj event(iniUser(id)) .
query id:bitstring; inj event(terminateServer(
    id)) ==> inj event(iniServer(id)) .
```

(c) Main

Figure 5.3: ProVerif Validation

1. RESULT inj-event(terminateServer(id)) ==> inj-event(iniServer(id)) is true.
2. RESULT inj-event(terminateUser(id_17079)) ==> inj-event(iniUser(id_17079)) is true.
3. RESULT not attacker (sk[]) is true.

The results indicates that the both server and user events started and terminated successfully, while not *attacker (sk[]) is true.* verifies that attacker is not able to find session key. Hence, proposed scheme posses authentication property.

5.6 Performance Analysis

In this section, we execute performance comparison of the proposed scheme with related existing schemes [5, 101, 105, 107] with respect to memory requirements of smart card, communication cost and computation cost. Following notations are introduced to understand the performance comparisons:

- t_h : time to calculate Hash Function
- $t_{\oplus}$: time to perform Exclusive OR operation
- t_{me} : time to perform Modular Exponentiation
- t_{enc} : time to perform Symmetric Encryption
- t_{dec} : time to perform Symmetric Decryption

For simplicity, the ID, PW_i, timestamps $\{t_{i.}, t_{s.}\}$, random number k_i, output of one way hash function etc. are taken as 128 bit long. Table 5.3 summarizes memory requirements, communication and computation cost of proposed scheme with existing schemes. Proposed scheme requires $128 \times 7 = 896$ bits memory in smart card on the other hand Kumari et al.'s scheme requires $128 \times 6 = 768$. An's scheme require $128 \times 5 = 640$ bits and Chung et al. and Wen et al's schemes require $128 \times 3 = 384$ bits. The memory overhead of the proposed scheme is because it stores extra parameters to perform built in login verification by the smart card, so as to avoid denial of service attack and storage of pseudo identity. The communication overhead of proposed scheme is 896 bits which is less than Wen et al.'s scheme, equal to An's scheme and higher than Kumari et al. and Chung et al.'s scheme. It is due to the fact that in proposed scheme after each successful login and authentication phase, server sends new pseudo identity XORed with user's signatures inorder to avoid user anonymity violation attack. Total computation cost of the proposed scheme is also slightly higher than Kumari et

Table 5.3: Comparison of Computation cost, Communication cost & Memory Requirements

Scheme:	Proposed	Kumari et al. [5]	Chung et al. [101]	An et al. [107]	Wen et al. [105]
SC_{ui}'s Memory (in bits)	$128 \times 7 = 896$	$128 \times 6 = 768$	$128 \times 3 = 384$	$128 \times 5 = 640$	$128 \times 3 = 384$
Communication cost	$128 \times 7 = 896$	$128 \times 6 = 768$	$128 \times 6 = 768$	$128 \times 7 = 896$	$128 \times 9 = 1152$
Computational cost					
Registration SC_{ui}	$1t_h + 2t_{\oplus}$	$1t_h + 2t_{\oplus}$	Nil	$1t_h$	Nil
Registration $\mathcal{S}$	$3t_h + 3t_{\oplus} + 1t_{enc}$	$3t_h + 3t_{\oplus}$	$2t_h + 1t_{\oplus}$	$2t_h + 2t_{\oplus}$	$5t_h + 4t_{\oplus}$
Login-Authentication SC_{ui}	$5t_h + 10t_{\oplus}$	$5t_h + 10t_{\oplus}$	$5t_h + 3t_{\oplus}$	$3t_h + 7t_{\oplus} + 2t_{me}$	$10t_h + 9t_{\oplus}$
Login-Authentication $\mathcal{S}$	$6t_h + 4t_{\oplus} + 1t_{dec}$	$6t_h + 3t_{\oplus}$	$5t_h + 2t_{\oplus}$	$5t_h + 5t_{\oplus} + 2t_{me}$	$10t_h + 9t_{\oplus}$
Total Computation cost	$15t_h + 19t_{\oplus} + 1t_{enc} + 1t_{dec}$	$15t_h + 18t_{\oplus}$	$12t_h + 6t_{\oplus}$	$11t_h + 14t_{\oplus} + 4t_{me}$	$25t_h + 22t_{\oplus}$

al., Chung at.al. and Wen et al.'s schemes and is lower than An's scheme.

5.7 Chapter Summary

In this chapter, we have cryptanalyzed Kumari et al.'s remote user authentication scheme based on symmetric cryptography primitives. We have shown that Kumari et al.'s scheme is vulnerable to user anonymity violation attack and smart card stolen attack. Furthermore, we have proposed an enhanced remote user authentication scheme to overcome the weaknesses of Kumari et al.'s scheme. It is evident from security analysis that the proposed scheme is robust against all known attacks. The enhanced scheme also ensures privacy and anonymity. Although the scheme incurs some extra memory, communication and computation cost due to storage and communication of user's pseudo identity, yet it is only because of this additional burden that the proposed scheme is able to resist user anonymity violation and smart card stolen attacks.

Chapter 6

An ECC based Two-factor Authentication Protocol for TMIS

The recent development in the field of computing and communication enabled the remote health services to be a viable solution, while reducing the social and economic burdens also enhancing the quality and efficiency. Telecare medical information system (TMIS) facilitates medical practitioners and patients to establish communication over public network to provide health care services directly in patient's home. A general structure for TMIS is illustrated in Fig. 6.1, involving a number of entities like the patients, TMIS server, the doctors, health care staff, ambulance for emergency and so on. The telecare medical server maintains the patient's history and private information for remote health care purposes. The patient's history and private information is very critical and is typically accessed by authorized doctor/health care staff for efficient and remote diagnosis and treatment. To get TMIS services remotely, the patient can connect to TMIS server using some telecare application via public Internet, the TMIS server administrator/ healthcare staff can further decide to forward the request to some doctor/ambulance staff etc. Some useful TMIS services includes, pendant alarm, movement monitoring and telephone services.

Besides the usefulness of TMIS, the security and privacy are the main concerns as the only communication link between patient and medical practitioner is the public Internet, so all the threats applicable on Internet are also applicable to TMIS. In addition to traditional security requirements, patient's privacy and anonymity has become an important feature to be maintained during communication with healthcare staff. The anonymity and privacy enables a remote patient to get desired healthcare services without revealing his real name or identity, even the doctor can acquire only the needed patient's health related information,

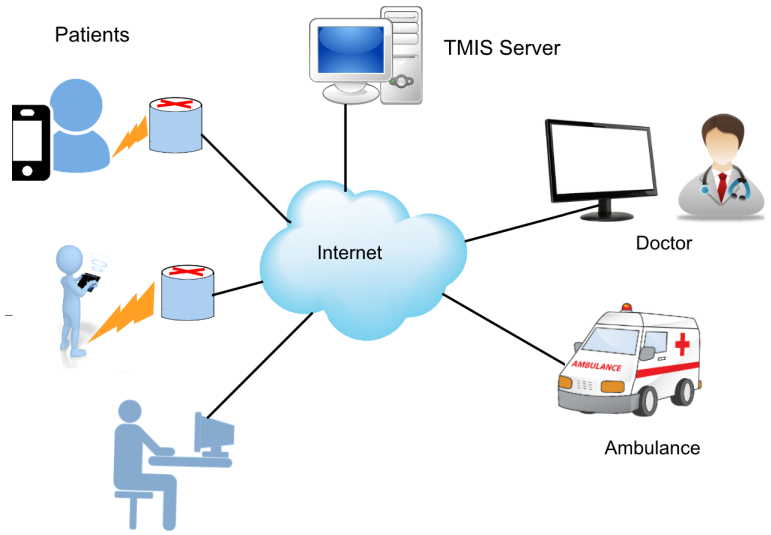


Figure 6.1: The Architecture of Remote Health Care Services

while the name and identity remains secret. The information exchanged between TMIS server and patient as well as the information stored in TMIS server is very critical and it is necessary to sort out who can access this information, failing which can expose the data to adversary. The adversary can use the data in some wrong way, such security breach can lead to distressing results on patient, even leading towards risking the patient's life, for example: The attacker can alternate pacemaker with deadly shocks [108]. To handle security issues usually password based two factor mutual authentication protocols are used to provide secure remote health-care facilities. Authenticated key agreement, if employed properly for TMIS may ensure security and privacy over insecure public network. The first such scheme was proposed by Diffie and Hellman [109] in 1976, although their scheme was vulnerable to different attacks [53, 110], but it provided a basis for authentication and key agreement research.

Password authenticated key agreements are of wide interest, as these provide confidentiality of the previous messages even if password of one party or a current session key is compromised. Recently a number of password based authentication protocols have been proposed [3, 6, 19, 44, 47, 49, 51, 52, 57, 58, 60, 65, 111–121]. Some of such schemes are vulnerable to different attacks [44, 47, 49, 51, 52, 111, 116, 117], while some other schemes do not preserve the privacy and anonymity of users [3, 19, 44, 51, 57, 65, 120, 121]. In 2010 Wu et al. [122] proposed an efficient key agreement protocol for TMIS, they introduced a pre-computation phase, but He et al. [123] proved their scheme to be vulnerable to impersonation and privileged inside attack, further He et al. [123] proposed an enhanced scheme but the scheme is still vulnerable to impersonation attack [116]. Wei et al. [116] proposed an improved scheme but Zhu et

Table 6.1: Notation Guide

Notations	Description	Notations	Description
E/F_p	Elliptic Curve	G	Base point over E/F_p
$\|\|$	String concatenation operator	$\oplus$	Bitwise XOR operation
$h(.)$	A one way hash function	$\mathcal{S}$	TMIS Server
$\mathcal{U}_i$, ID_i	Patient and his identity	PW_i	$\mathcal{U}_i$'s Password
$\mathcal{S}$	Legal TMIS Server	k_s	Master secret key of $\mathcal{S}$
PID_i	Pseudo identity of Patient $\mathcal{U}_i$	$\mathcal{A}$	The Adversary

al. [124] proved their scheme still vulnerable to offline password guessing attack. Khan et al. [125] proposed another enhanced authentication and key agreement scheme for TMIS, but Chen et al. [126] showed their scheme is vulnerable to privileged insider attack . The enhanced scheme of Chen et al. could not provide anonymity & untraceability as mentioned by Jiang et al. [111].

Very recently Xu et al. [61] proposed a two factor authenticated key agreement for TMIS. The protocol made an efficient use of elliptic curve cryptography, the protocol also ensured user anonymity, but Islam and Khan [6] proved their protocol failed in achieving strong authentication, failed to provide correct password change and unable for revocation of stolen smart card. They [6] also claimed that the protocol [61] is vulnerable to strong replay attack. Furthermore, in order to cope with the draw backs of Xu et al.'s protocol [61], Islam and Khan [6] proposed an improved protocol and claimed it to be secure against known attacks. This chapter proves that Islam et al.'s protocol [6] is vulnerable to server impersonation attack as well as user impersonation attack. Furthermore, an enhanced protocol is proposed to improve the security of Islam and Khan's protocol [6]. The rest of the chapter is organized as follows. Section 6.1 reviews Islam and Khan protocol [6]. The cryptanalysis of Islam and Khan's protocol is performed in section 6.2. The proposed enhanced protocol is described in section 6.3, while the comparative security and performance analysis is summarized in section 6.4. Finally, chapter's summary is solicited in section 6.5.

6.1 Review of Islam and Khan's Protocol

This section reviews Islam and Khan's two factor authentication protocol for TMIS, the protocol is illustrated in Fig. 6.2, which can be described by following three phases:

6.1.1 System Initialization Phase

TMIS Server $\mathcal{S}$ selects a prime number p and generates E/F_p, then choose a point G as base point over selected curve. $\mathcal{S}$ selects his master secret key $k_s \in Z_p^*$ and one way hash function $h(.) : 0,1 \rightarrow Z_p^*$. Finally, $\mathcal{S}$ publishes $\{F_p, E/F_p, p, G, h(.)\}$ and keeps k_s secret.

6.1.2 Registration Phase

Registration phase consists of two steps firstly the patient $\mathcal{U}_i$ selects his identity ID_i, password PW_i and a random number $r_i \in_R Z_p^*$. $\mathcal{U}_i$ further computes $l_i = h(ID_i||PW_i||r_i)$ and sends the tuple ID_i, l_i to $\mathcal{S}$ via some secure channel. $\mathcal{S}$ after receiving ID_i, l_i performs identity verification, if $\mathcal{U}_i$ is a new patient/user, it sets $N_i = 0$, otherwise sets $N_i = N_i + 1$ and stores (ID_i, N_i) in his database. Further, $\mathcal{S}$ selects a random number $b_s \in_R Z_p^*$ and computes $\alpha = \frac{b_s + k_s}{l_i} \mod p$, $B_i = b_s.G$ and $u_i = h(k_s.G||l_i)$. $\mathcal{S}$ stores $E/F_p, G, u_i, B_i, \alpha, h(), p, N_i$ in smart card. $\mathcal{S}$ handover the smart card to $\mathcal{U}_i$ through secure channel. Upon receiving smart card $\mathcal{U}_i$ stores r_i in smart card.

6.1.3 Login and Mutual Authentication with Key Exchange Phase

Step 1: $\mathcal{U}_i$ initiates authentication process by inserting his smart card in the specialized reader and entering his identity ID_i and password PW_i. The smart card computes $l_i = h(ID_i||PW_i||r_i)$, $k_s.G = (\alpha.l_i)G - B_i$, $u_i^* = h(k_s.G||l_i)$, and verifies $u_i^* \stackrel{?}{=} u_i$, aborts the session if invalid, otherwise generates a nonce $a_i \in_R Z_p^*$, T_{i1} and computes his pseudo identity $PID_i = ID_i \oplus h(k_s.G||T_{i1})$ and $C_i = a_i.(k_s.G)$, $G_i = h(ID_i||C_i||T_{i1}||k_s.G||N_i)$, then $\mathcal{U}_i$ sends $m_i = \{PID_i, C_i, G_i, T_{i1}\}$ to $\mathcal{S}$ as login message.

Step 2: Upon receiving m_i, $\mathcal{S}$ checks the validity of timestamp T_{i1}, aborts the session if timestamp is not valid, otherwise computes $ID_i^{'} = PID_i \oplus h(k_s.G||T_{i1})$, $G_i^{'} = h(ID_i^{'}||C_i||T_{i1}||k_s.G||N_i)$. $\mathcal{S}$ checks $G_i^{'} \stackrel{?}{=} G_i$, if false, session is aborted. Otherwise, $\mathcal{S}$ selects $c_s \in_R Z_p^*$, new timestamp for $\mathcal{U}_i$. Then $\mathcal{S}$ computes $C_s = c_s(k_s.G)$, $C_{si} = c_s(C_i)$. Then $\mathcal{S}$ calculates the session key $SK = h(ID_i^{'}||C_i||C_s||C_{si}||k_s.G)$ and $G_s = h(SK||C_s||T_{i2}||k_s.G)$. $\mathcal{S}$ stores (ID_i, N_i, T_{i1}) in his database and sends $m_s = \{C_s, G_s, T_{i2}\}$ to $\mathcal{U}_i$.

Step 3: $\mathcal{U}_i$ first verifies T_{i2}, abort the session if not valid otherwise Compute $C_{is} = a_i(C_s)$. $\mathcal{U}_i$ further calculates session key $SK' = h(ID_i||C_i||C_s||C_{is}||k_s.G)$ and $G_s^{'} = h(SK'||C_s||T_{i2}-$

Patient $\mathcal{U}_i$	*TMIS Server* $\mathcal{S}$
Registration Phase:	
Select identity ID_i, password PW_i and a number $r_i \in_R Z_p^*$	
Calculates $l_i = h(ID_i \| PW_i \| r_i)$	
$\xrightarrow{ID_i, l_i}$	
	Selects a random number $b_s \in_R Z_p^*$
	if fresh set $N_i = 0$, else $N_i = N_i + 1$
	Computes $\alpha = \frac{b_s + k_s}{l_i} \mod p$, $B_i = b_s.G$, $u_i = h(k_s.G \| l_i)$
	stores $E/F_p, G, u_i, B_i, r_i, \alpha, h(), p, N_i$ in smart card
$\xleftarrow{SmartCard}$	
stores r_i in smart card	
Login & Authentication Phase:	
Enter ID_i, and PW_i	
$l_i = h(ID_i \| PW_i \| r_i)$	
$k_s.G = (\alpha.l_i)G - B_i$	
$u_i^* = h(k_s.G \| l_i)$	
$u_i^* \stackrel{?}{=} u_i$, abort if not true	
Selects $a_i \in_R Z_p^*$, T_{i1}	
$PID_i = ID_i \oplus h(k_s.G \| T_{i1})$	
$C_i = a_i.(k_s.G)$	
$G_i = h(ID_i \| C_i \| T_{i1} \| k_s.G \| N_i)$	
$\xrightarrow{m_i = \{PID_i, C_i, G_i, T_{i1}\}}$	
	Check T_{i1}, abort if not valid
	$ID_i^{'} = PID_i \oplus h(k_s.G \| T_{i1})$
	$G_i^{'} = h(ID_i^{'} \| C_i \| T_{i1} \| k_s.G \| N_i)$
	Check $G_i^{'} \stackrel{?}{=} G_i$, abort if not valid
	Select $c_s \in_R Z_p^*$, T_{i2}
	$C_s = c_s(k_s.G)$
	$C_{si} = c_s(C_i)$
	$SK = h(ID_i^{'} \| C_i \| C_s \| C_{si} \| k_s.G)$
	$G_s = h(SK \| C_s \| T_{i2} \| k_s.G)$
	Store (ID_i, N, T_{i1})
$\xleftarrow{m_s = \{C_s, G_s, T_{i2}\}}$	
Check T_{i2}, abort if not valid	
$C_{is} = a_i(C_s)$	
$SK' = h(ID_i \| C_i \| C_s \| C_{is} \| k_s.G)$	
$G_s^{'} = h(SK' \| C_s \| T_{i2} \| k_s.G)$	
$G_s^{'} \stackrel{?}{=} G_s$ abort if not valid	
Accept SK as session key	

Figure 6.2: Islam and Khan's Protocol

$||k_s.G)$. Upon receiving m_s from $\mathcal{S}$, $\mathcal{U}_i$ further verifies $G_s^{'} \stackrel{?}{=} G_s$, if the relationship proves to be false, the session is aborted by $\mathcal{U}_i$. Otherwise, $\mathcal{U}_i$ accepts SK as shared key with $\mathcal{S}$.

6.1.4 Password Change Phase

In Islam and Khan's protocol, the patient/user $\mathcal{U}_i$ can change his/her password without involving any communication with TMIS server $\mathcal{S}$. Firstly, $\mathcal{U}_i$ enter his smart card into reader then inputs his ID_i and PW_i. The smart card computes $l_i = h(ID_i||PW_i||r_i)$, $k_s.G = (\alpha.l_i)G - B_i$ and $u_i^* = h(k_s.G||l_i)$. The smart card further verifies $u_i^* \stackrel{?}{=} u_i$, if not holds, smart card aborts the request. Otherwise, asks $\mathcal{U}_i$ for new password. $\mathcal{U}_i$ selects new $r_{i_{new}} \in_R Z_p^*$, $PW_{i_{new}}$ and submit these to smart card, which computes $l_{i_{new}} = h(ID_i||PW_{i_{new}}||r_{i_{new}})$, $\alpha_{new} = \frac{l_i\alpha}{l_{i_{new}}} = \frac{b_s+k_s}{l_{i_{new}}}$ & $u_{i_{new}} = h(k_s.G||l_{i_{new}})$. Smart card stores new values of $u_{i_{new}}, r_{i_{new}}, \alpha_{new}$.

6.2 Cryptanalysis of Islam and Khan's Protocol

This section shows Islam and Khan's protocol is vulnerable to TMIS server impersonation and patient/user impersonation attacks. We show that an adversary can easily masquerade as a legitimate TMIS server to share a session key with peer.

6.2.1 Server Impersonation Attack

This subsection shows that a legitimate patient can easily impersonate as a legal TMIS server. Let $\mathcal{U}_j$ be a legal patient, who wants to impersonate as legal TMIS server $\mathcal{S}$. $\mathcal{U}_j$ will perform following steps to impersonate him as $\mathcal{S}$.

Step 1: $\mathcal{U}_j$ extracts the information $E/F_p, G, u_j, B_j, r_j, \alpha, h(), p, N_j$ stored on his smart card by using the power analysis as mentioned in [28, 29]. $\mathcal{U}_j$ then enter his ID_j and PW_j, and computes following:

$$l_j = h(ID_j||PW_j||r_j) \tag{6.1}$$

$$k_s.G = (\alpha.l_j)G - B_j \tag{6.2}$$

Step 2: When patient $\mathcal{U}_i$ initiates the login and authentication process by sending $m_i = \{PID_i, C_i, G_i, T_{i1}\}$ to $\mathcal{S}$. $\mathcal{U}_j$ intercepts the message m_i.

Step 3: $\mathcal{U}_j$ retrieves $ID_i^{'} = PID_i \oplus h(k_s.G||T_{i1})$ and selects $c_j \in_R Z_p^*$, T_{i2} and computes:

$$C_j = c_j(k_s.G) \tag{6.3}$$

$$C_{ji} = c_j(C_i) \tag{6.4}$$

$$SK = h(ID_i^{'}||C_i||C_j||C_{ji}||k_s.G) \tag{6.5}$$

$$G_s = h(SK||C_j||T_{i2}||k_s.G) \tag{6.6}$$

Step 4: $\mathcal{U}_j$ sends $m_j = \{C_j, G_j, T_{i2}\}$ to $\mathcal{U}_i$.

Step 5: $\mathcal{U}_i$ verifies T_{i2} and computes:

$$C_{ij} = a_i.(C_j) \tag{6.7}$$

$$SK' = h(ID_i||C_i||C_j||C_{ij}||k_s.G) \tag{6.8}$$

$$G_j^{'} = h(SK'||C_j||T_{i2}||k_s.G) \tag{6.9}$$

Further, $\mathcal{U}_i$ checks $G_j^{'} \stackrel{?}{=} G_j$, as both are equal so $\mathcal{U}_i$ accept the session key SK' and $\mathcal{U}_j$ as legitimate TMIS server $\mathcal{S}$.

Hence, a legal patient/user $\mathcal{U}_j$ can impersonate himself as legitimate server $\mathcal{S}$ to all other legal users easily. So, it can be rightly said that Islam and Khan's scheme [6] is vulnerable to server impersonation attack.

6.2.2 User Impersonation Attack

This subsection shows that Islam and Khan's protocol is also vulnerable to user impersonation attack. A legal patient $\mathcal{U}_j$ can impersonate another legal patient $\mathcal{U}_i$ to a legal TMIS server $\mathcal{S}$. Let $\mathcal{U}_j$ be a legal patient who wants to impersonate as another legal patient $\mathcal{U}_i$. $\mathcal{U}_j$ will perform following steps to impersonate him as $\mathcal{U}_i$.

Step 1: $\mathcal{U}_j$ extracts the information $E/F_p, G, u_j, B_j, r_j, \alpha, h(), p, N_j$ stored on his smart card by using the power analysis as mentioned in [28, 29]. $\mathcal{U}_j$ then enters his ID_j and PW_j, and computes:

$$l_j = h(ID_j||PW_j||r_j) \tag{6.10}$$

$$k_s.G = (\alpha.l_j)G - B_j \tag{6.11}$$

Step 2: Let $\mathcal{U}_j$ takes access to N_i stored in server database using stolen verifier attack as

mentioned in [110, 127, 128].

Step 3: When another patient/user $\mathcal{U}_i$ initiates the login and authentication process by sending $m_i = \{PID_i, C_i, G_i, T_{i1}\}$ to $\mathcal{S}$. $\mathcal{U}_j$ intercepts the message m_i and passively computes $ID_i = PID_i \oplus h(k_s.G||T_{i1})$ and let the session terminate.

Step 4: After the session between $\mathcal{U}_i$ and $\mathcal{S}$ terminates, $\mathcal{U}_j$ generates a nonce $a_i^* \in_R Z_p^*$, new timestamp T_{i1}^* and computes:

$$PID_i = ID_i \oplus h(k_s.G||T_{i1}^*) \tag{6.12}$$

$$C_i^* = a_i^*(k_s.G) \tag{6.13}$$

$$G_i^* = h(ID_i||C_i^*||T_{i1}^*||k_s.G||N_i) \tag{6.14}$$

then $\mathcal{U}_j$ sends $m_i^* = \{PID_i, C_i^*, G_i^*, T_{i1}^*\}$ to $\mathcal{S}$ as login message.

Step 5: Upon receiving m_i^*, $\mathcal{S}$ checks the validity of timestamp T_{i1}^*, aborts the session if timestamp is not valid, otherwise computes:

$$ID_i^{'} = PID_i \oplus h(k_s.G||T_{i1}^*) \tag{6.15}$$

$$G_i^{'} = h(ID_i^{'}||C_i^*||T_{i1}^*||k_s.G||N_i) \tag{6.16}$$

then $\mathcal{S}$ checks $G_i^{'} \stackrel{?}{=} G_i^*$, if falsify, session is aborted, otherwise $\mathcal{S}$ selects $c_s \in_R Z_p^*$, new timestamp T_{i2} for $\mathcal{U}_i$. Then $\mathcal{S}$ computes:

$$C_s = c_s(k_s.G) \tag{6.17}$$

$$C_{si} = c_s(C_i^*) \tag{6.18}$$

$$SK = h(ID_i^{'}||C_i^*||C_s||C_{si}||k_s.G) \tag{6.19}$$

$$G_s = h(SK||C_j||T_{i2}||k_s.G) \tag{6.20}$$

$\mathcal{S}$ stores (ID_i, N_i^*, T_{i1}^*) in his database and sends $m_s = \{C_s, G_s, T_{i2}\}$ to $\mathcal{U}_i$. $\mathcal{U}_j$ intercepts the message.

Step 6: $\mathcal{U}_j$ computes:

$$C_{is} = a_i^*(C_s) \tag{6.21}$$

$$SK' = h(ID_i||C_i||C_s||C_{is}||k_s.G) \tag{6.22}$$

$$G_s^{'} = h(SK'||C_j||T_{i2}||k_s.G) \tag{6.23}$$

$\mathcal{U}_j$ keeps SK as shared key with $\mathcal{S}$.

Hence, a legal patient $\mathcal{U}_j$ can impersonate himself as another legal patient $\mathcal{U}_i$ to server $\mathcal{S}$. Therefore, Islam and Khan's [6] protocol is vulnerable to patient impersonation attack.

6.3 Proposed Scheme

The security of Islam and Khan's protocol relies upon a general parameter $k_s.G$, which can be easily calculated by any legal user/patient, so any legal patient after computing $k_s.G$ can easily calculate any other patient's ID when he initiates a session. So, instead of using a general value ($k_s.G$), if we use a unique value for each patient then the security of their protocol may be enhanced. Therefore, we are improving Islam and Khan's protocol by storing an extra unique value on smart card and some alterations in login and authentication phase as illustrated in Fig. 6.3. The improved proposed protocol works as follows:

6.3.1 Registration Phase

We enhanced registration phase by only storing an extra value in smart card. $\mathcal{S}$ after storing $E/F_p, G, u_i, B_s, \alpha$, $h(), p, N_i$ in smart card, also computes $O_i = h(ID_i||k_s) \oplus l_i$ and stores it in smart card, finally the smart card contains $\{E/F_p, G, u_i, B_i, \alpha, h(), p, N_i, O_i, r_i\}$.

6.3.2 Login and Mutual Authentication with Key Exchange Phase

When $\mathcal{U}_i$ wants to login, he initiates the process by inserting his smart card in the reader and entering his identity ID_i and password PW_i. Following steps will be performed by the smart card and the TMIS server.

Step 1: Smart card computes:

$$l_i = h(ID_i||PW_i||r_i) \tag{6.24}$$

$$k_s.G = (\alpha.l_i)G - B_i \tag{6.25}$$

$$u_i^* = h(k_s.G||l_i) \tag{6.26}$$

Patient $\mathcal{U}_i$	*TMIS Server* $\mathcal{S}$
Registration Phase:	
Select identity ID_i, password PW_i and a number $r_i \in_R Z_p^*$	
Calculates $l_i = h(ID_i \| PW_i \| r_i)$	
$\xrightarrow{ID_i, l_i}$	
	Selects a random number $b_s \in_R Z_p^*$
	if fresh set $N_i = 0$, else $N_i = N_i + 1$
	Computes $\alpha = \frac{b_s + k_s}{l_i} \bmod p$, $B_i = b_s.G$, $u_i = h(k_s.G \| l_i)$, $\boxed{O_i = h(ID_i \| k_s) \oplus l_i}$
	stores $E/F_p, G, u_i, B_i, r_i, \alpha, h(), p, N_i, O_i$ in smart card
$\xleftarrow{SmartCard}$	
stores r_i in smart card	
Login & Authentication Phase:	
Enter ID_i, and PW_i	
$l_i = h(ID_i \| PW_i \| r_i)$	
$k_s.G = (\alpha.l_i)G - B_i$	
$u_i^* = h(k_s.G \| l_i)$	
$u_i^* \stackrel{?}{=} u_i$, abort if not true	
Selects $a_i \in_R Z_p^*$, T_{i1}	
$\boxed{PID_i = ID_i \oplus a_i.G}$	
$C_i = a_i.(k_s.G)$	
$\boxed{G_i = h(ID_i \| O_i \oplus l_i \| C_i \| T_{i1} \| k_s.G \| N_i)}$	
$\xrightarrow{m_i = \{PID_i, C_i, G_i, T_{i1}\}}$	
	Check T_{i1}, abort if not valid
	$\boxed{ID_i^{'} = PID_i \oplus (C_i.k_s^{-1})}$
	$G_i^{'} = h(ID_i^{'} \| h(ID_i^{'} \| k_s) \| C_i \| T_{i1} \| k_s.G \| N_i)$
	Check $G_i^{'} \stackrel{?}{=} G_i$, abort if not valid
	Select $c_s \in_R Z_p^*$, T_{i2}
	$C_s = c_s(k_s.G)$
	$C_{si} = c_s(C_i)$
	$\boxed{SK = h(ID_i^{'} \| h(ID_i^{'} \| k_s) \| C_i \| C_s \| C_{si} \| k_s.G)}$
	$G_s = h(SK \| C_s \| T_{i2} \| k_s.G)$
	Store (ID_i, N, T_{i1})
$\xleftarrow{m_s = \{C_s, G_s, T_{i2}\}}$	
Check T_{i2}, abort if not valid	
$C_{is} = a_i(C_s)$	
$\boxed{SK' = h(ID_i \| O_i \oplus l_i \| C_i \| C_s \| C_{is} \| k_s.G)}$	
$G_s^{'} = h(SK' \| C_s \| T_{i2} \| k_s.G)$	
$G_s^{'} \stackrel{?}{=} G_s$ abort if not valid	
Accept SK as session key	

Figure 6.3: Proposed Protocol

Then smart card checks:

$$u_i^* \stackrel{?}{=} u_i \tag{6.27}$$

The smart card aborts the session if $u_i^* \neq u_i$. Otherwise, generates a nonce $a_i \in_R Z_p^*$ and a timestamp T_{i1} then computes:

$$PID_i = ID_i \oplus a_i.G \tag{6.28}$$

$$C_i = a_i.(k_s.G) \tag{6.29}$$

$$G_i = h(ID_i||O_i \oplus l_i||C_i||T_{i1}||k_s.G||N_i) \tag{6.30}$$

$\mathcal{U}_i$ sends $m_i = \{PID_i, C_i, G_i, T_{i1}\}$ to $\mathcal{S}$.

Step 2: Upon receiving m_i, $\mathcal{S}$ checks the validity of timestamp T_{i1}, aborts the session if timestamp is not valid. Otherwise, computes:

$$ID_i^{'} = PID_i \oplus (C_i.k_s^{-1}) \tag{6.31}$$

$$G_i^{'} = h(ID_i^{'}||h(ID_i^{'}||k_s)||C_i||T_{i1}||k_s.G||N_i) \tag{6.32}$$

$\mathcal{S}$ then checks $G_i^{'} \stackrel{?}{=} G_i$, aborts if $G_i^{'} \neq G_i$, Otherwise selects $c_s \in_R Z_p^*$ and a new timestamp T_{i2}, then computes:

$$C_s = c_s(k_s.G) \tag{6.33}$$

$$C_{si} = c_s(C_i) \tag{6.34}$$

$$SK = h(ID_i^{'}||h(ID_i||k_s)||C_i||C_s||C_{si}||k_s.G) \tag{6.35}$$

$$G_s = h(SK||C_s||T_{i2}||k_s.G) \tag{6.36}$$

$\mathcal{S}$ Store (ID_i, N_i, T_{i2}) in his database and sends $m_s = \{C_s, G_s, T_{i2}\}$ to $\mathcal{U}_i$.

Step 3: Upon receiving m_s, $\mathcal{U}_i$ first verifies T_{i2}, abort the session if not valid. Otherwise, $\mathcal{U}_i$ computes:

$$C_{is} = a_i(C_s) \tag{6.37}$$

$$SK' = h(ID_i||O_i \oplus l_i||C_i||C_s||C_{is}||k_s.G) \tag{6.38}$$

$$G_s^{'} = h(SK'||C_s||T_{i2}||k_s.G) \tag{6.39}$$

Then $\mathcal{U}_i$ check $G_s^{'} \stackrel{?}{=} G_s$, aborts if $G_s^{'} \neq G_s$. Otherwise, $\mathcal{U}_i$ accepts SK as shared key with $\mathcal{S}$.

Table 6.2: Security Analysis

Protocol→ Security Properties↓	Wei et al. [116]	Xu et al. [61]	Islam and Khan [6]	Proposed
Mutual Authentication	Yes	Yes	Yes	Yes
User/Patient Anonymity	No	Yes	Yes	Yes
Perfect Forward Secrecy	Yes	Yes	Yes	Yes
Wrong Password Detection at Login Phase	No	No	Yes	Yes
Replay Attack	Insecure	Insecure	Secure	Secure
Impersonation Attack	Insecure	Secure	Insecure	Secure
Privileged Insider Attack	Secure	Insecure	Secure	Secure
Man-in-Middle Attack	Insecure	Insecure	Insecure	Secure
offline Password Guessing Attack	Insecure	Secure	Secure	Secure

6.3.3 Password Change Phase

Similar to Islam and Khan protocol, in our improved protocol $\mathcal{U}_i$ changes his password without involvement of the TMIS server $\mathcal{S}$. Firstly, $\mathcal{U}_i$ enters his smart card into reader then inputs his ID_i and PW_i. The smart card computes $l_i = h(ID_i||PW_i||r_i)$, $k_s.G = (\alpha.l_i)G - B_i$ and $u_i^* = h(k_s.G||l_i)$. The smart card further verifies $u_i^* \stackrel{?}{=} u_i$, if it doesn't hols, the smart card aborts the request. Otherwise, $\mathcal{U}_i$ enters new password $PW_{i_{new}}$ and new $r_{i_{new}} \in_R Z_p^*$. The smart card further computes $l_{i_{new}} = h(ID_i||PW_{i_{new}}||r_{i_{new}})$, $\alpha_{new} = \frac{l_i\alpha}{l_{i_{new}}} = \frac{b_s+k_s}{l_{i_{new}}}$, $u_{i_{new}} = h(k_s.G||l_{i_{new}})$. and $O_{i_{new}} = O_i \oplus l_i \oplus l_{i_{new}}$, and stores new values of $u_{i_{new}}, r_{i_{new}}, \alpha_{new}, O_{i_{new}}$ and discards the old values.

6.4 Comparative Analysis

6.4.1 Security Analysis

This section analyzes the security of proposed scheme. The scheme provides mutual authentication, resist user and server impersonation attacks. Furthermore, the proposed scheme is secure against privileged insider, stolen verifier, man-in-middle and offline password guessing attacks. The scheme also provides perfect forward secrecy.

6.4.1.1 Mutual Authentication

The enhanced protocol provides mutual authentication. $\mathcal{S}$ authenticates $\mathcal{U}_i$ by computing $G_i^{'}$. The computation of $G_i^{'}$ involves the computation of $O_i \oplus l_i$, where O_i is stored in smart card while l_i requires user password PW_i, so in order to generate valid $O_i \oplus l_i$, the adversary needs

$\mathcal{U}_i$'s password. Furthermore, $\mathcal{S}$ is authenticated by $\mathcal{U}_i$ by verifying G_s, which is computed using session key $SK = h(ID_i||h(ID_i||k_s)||C_i||C_s||C_{si}||k_s.G)$ and the computation of G_s. Here, SK requires secret key k_s of $\mathcal{S}$. Adversary without having the secret key of $\mathcal{S}$ can not compute the session key SK and G_s.

6.4.1.2 User Anonymity

The enhanced protocol ensures user/patient anonymity. During login session the ID_i of patient is not sent over public media rather a pseudo identity PID_i, freshly generated for session is sent to $\mathcal{S}$, further ID_i can only be revealed by the use of $\mathcal{S}$'s secret key k_s. Hence, the proposed protocol provides strong user anonymity.

6.4.1.3 Replay Attack

The enhanced protocol prevents replay attack. Similar to Islam and Khan protocol a timestamp T_{i1} is sent by $\mathcal{U}_i$ to $\mathcal{S}$ as plain text as well as within G_i which is protected by hash function. For every new session, the fresh timestamp is generated and verified. Similar procedure is performed at user side to ensure freshness. Furthermore, for each session a new user pseudo identity PID is generated by user selected parameter a_i. Therefore, in-order to generate valid G_i and C_i the adversary needs the knowledge of a_i and $h(ID_i||k_s)$. Extracting a_i from $a_i(k_s.G)$ is untraceable elliptic curve discrete logarithm problem and finding k_s from G_i is also ECDLP and protected by a hash function. Therefore, the proposed protocol prevents replay attack.

6.4.1.4 Impersonation Attack

In proposed protocol, an adversary $\mathcal{A}$ can impersonate as legitimate TMIS server if $\mathcal{A}$ is able to calculate $h(ID_i||k_s)$ because the computation of both the session key SK and server signature G_s require $h(ID_i||k_s)$ to be calculated first, where k_s is secret key of server. Whereas, ID_i can only be extracted from $\mathcal{U}_i$'s pseudo identity PID_i, the extraction of ID_i from PID_i is also done through $\mathcal{S}$'s secret key k_s. So adversary $\mathcal{A}$ cannot impersonate himself as legitimate TMIS server $\mathcal{S}$ without knowing $\mathcal{S}$'s secret key. An adversary $\mathcal{A}$ can impersonate as a legal patient $\mathcal{U}_i$, if $\mathcal{A}$ can generate valid signature G_i, which requires to compute l_i, further l_i can only be computed by knowing user password PW_i. Hence, an adversary $\mathcal{A}$ cannot impersonate as a legal patient $\mathcal{U}_i$ without knowledge of $\mathcal{U}_i$'s password PW_i. Therefore, the proposed scheme resists impersonation attacks.

6.4.1.5 Privileged Insider Attack

During registration $\mathcal{U}_i$ sends ID_i and $l_i = h(ID_i||PW_i||r_i)$, where random r_i is generated by $\mathcal{U}_i$, computing PW_i and r_i from l_i is protected by a hash function. Furthermore, no verifier table is maintained for $\mathcal{U}_i$'s password, $\mathcal{S}$ uses his secret key k_s for authentication. Therefore, no privileged insider can ever access user password, hence the proposed scheme is secure against privileged insider attack and stolen verifier attacks.

6.4.1.6 Man-in-Middle Attack

An adversary $\mathcal{A}$ can launch man-in-middle attack if and only if he can pass through the authentication from $\mathcal{S}$ and $\mathcal{U}_i$. However, it has been proved in subsection 6.4.1.1 that no adversary can pass the authentication from $\mathcal{S}$ without having $\mathcal{U}_i$'s password and smart card. Similarly, the adversary can not pass authentication from $\mathcal{U}_i$ without having $\mathcal{S}$'s secret key k_s.

6.4.1.7 Offline Password Guessing Attack

If by any means an adversary $\mathcal{A}$ gets $\mathcal{U}_i$'s smart card and reveals the information stored in it. Even then he will not be able to guess $\mathcal{U}_i$'s password, as the only parameter stored in smart card related to password is $O_i = h(ID_i||k_s) \oplus h(ID_i||PW_i||r_i)$. The adversary $\mathcal{A}$ can get r_i, but ID_i, PW_i and k_s are not known to him. The correct estimation of three values protected by hash cannot be determined in polynomial time [129]. Therefore, the proposed protocol resists offline password guessing attack.

6.4.1.8 Perfect Forward Secrecy

An authentication and key agreement is said to posses perfect forward secrecy if the adversary $\mathcal{A}$ having both $\mathcal{U}_i$'s password PW_i as well as $\mathcal{S}$'s secret key k_s, but still not be able to get previously generated session keys. For computing a session key in proposed protocol $\mathcal{U}_i$ chooses a new random a_i and $\mathcal{S}$ selects a new random c_s unique for each session, so freshness of session key is guaranteed, the adversary having PW_i as well as k_s still need to know the session specific a_i and c_s to calculate that session's key. Therefore, the proposed protocol provides perfect forward secrecy.

Table 6.3: Computation Cost Analysis

Protocol	Patient $\mathcal{U}_i$	TMIS Server $\mathcal{S}$
Wei et al. [116]	$T_{me} + 5T_h \approx 385ms$	$T_{me} + T_{mi} + 5T_h \approx 3.51ms$
Xu et al. [61]	$3T_{pm} + 6T_h \approx 396ms$	$3T_{pm} + 5T_h \approx 3.56ms$
Islam and Khan [6]	$2T_{pm} + 6T_S \approx 266ms$	$T_{pm} + 3T_h \approx 1.20ms$
Proposed	$3T_{pm} + 5T_h \approx 395ms$	$T_{pm} + 3T_h \approx 1.20ms$

6.4.2 Performance Analysis

This subsection performs the comparative performance analysis of proposed scheme with existing schemes [6, 61, 116] with respect to computation cost, communication overhead and the storage required.

6.4.2.1 Computation Cost Analysis

For computational cost analysis following notations are introduced:

- T_{me} : time for modular exponentiation
- T_{pm} : time for point multiplication
- T_{mi} : time for modular inversion
- T_h : time for hash operation

Modular exponentiation (T_{me}), modular inversion (T_{mi}) and point multiplication (T_{pm}) are the major operations in proposed and existing schemes, which takes 380 ms, 30 ms and 130 ms respectively, on Philips HiPersmartcard with clock speed 36 MHz [75], while hash operation takes $1ms$. Similarly, for server side Pentium IV processor with clock speed 3GHz, the execution time for these operations are 3.16 ms, 1.17 ms, 0.3 ms and 0.01 ms respectively. Table 6.3 summarizes the computation cost comparison of proposed protocol with existing protocols [6, 61, 116]. The proposed protocol achieves same computation cost as of Islam and Khan's protocol at TMIS server side, while it is slightly heavier at patient side, which is due to an extra point multiplication to compute the pseudo identity of patient ($PID_i = ID_i \oplus a_iG$) in proposed protocol to ensure resistance to impersonation attack.

6.4.2.2 Communication Cost and Memory Requirements Analysis

Table 6.4 summarizes total bytes transmitted by both $\mathcal{U}_i$ and $\mathcal{S}$ and the memory requirements to store security parameters in bytes. For comparison, it has been assumed that identity,

Table 6.4: Comparison of Communication Cost and Memory Requirements

Bytes Transmitted	Proposed	Islam and Khan [6]	Xu et al. [61]	Wei et al. [116]
Login Phase	80	80	80	164
Authentication Phase	60	60	60	164
Memory Requirements	140	120	100	404

hash digest and timestamps are 160 bit long, while ECC recommended size by NIST for key is 160 bits and for RSA same is 1024 bits.

During login and authentication phase of proposed, Islam and Khan and Xu et al. protocols, $\mathcal{U}_i$ sends $\{PID_i, C_i,\ G_i, T_{i1}\}$ each of 160 bit/20 bytes long so total bytes sent by $\mathcal{U}_i$ are $20 \times 4 = 80\ bytes$, while $\mathcal{S}$ sends $\{C_s, G_s, T_{i2}\}$, total bytes sent by $\mathcal{S}$ are $20 \times 3 = 60\ bytes$. During login and authentication phase of Wei et al.'s protocol $\mathcal{U}_i$ sends $\{ID_i, B', R_1\}$, where B' is of 1024 bits/128 bytes long, so total bytes transmitted by $\mathcal{U}_i$ are $20 \times 2 + 128 = 168\ bytes$.

The smart card for Wei et al. protocol stores $\{ID_i, \beta,\ g, b\}$, which is $20 + 128 \times 3 = 404\ bytes$. Xu et al.'s smart card stores $\{G, Y, B_i, r_i, p\}$ and is $20 \times 5 = 100\ bytes$. Islam and Khan's smart card stores $\{G, u_i, B_i, r_i, \alpha, p\}$ and is $20 \times 6 = 120\ bytes$. The smart card in proposed protocol stores an extra parameter O_i as compared to Islam and Khan's protocol so its storage overhead is $20 \times 7 = 140\ bytes$. The proposed protocol achieves same communication overhead as of Islam and Khan's protocol, while ensuring resistance to all known attacks.

6.5 Chapter Summary

In this chapter, we analyzed Islam and Khan's two factor authentication protocol for TMIS based on ECC, our analysis revealed that Islam and Khan's protocol is vulnerable to user and sever impersonation attacks. In-order to enhance the security, we have proposed an improved protocol. Although, proposed protocol incurs some extra storage and computation cost at user side but having communication overhead same as Islam and Khan's protocol. The proposed protocol while maintaining all the merits of Islam and Khan's protocol is also robust against all known attacks.

Chapter 7

A Biometric Based three-factor Authentication Scheme for TMIS

Two factor authentication schemes [6, 44, 47, 49, 51, 52, 58, 104, 111, 116, 117] or not having the notion of user anonymity and privacy [3, 3, 19, 44, 51, 53, 57, 120, 121] for TMIS are nowadays converging into three factor authentication schemes, because massive amount of open issues has been exploded over three factor authentication that are getting attention of the huge research community. All this happen, because soon it was realized that two factor authentication can be easily deceived. As two factor authentication depends upon knowledge and ownership factors and therefore, it is supposed to be extra secure than commonly used single factor authentication. The knowledge factor refers to the knowledge of the user such as password or pin codes. On the other hand ownership factor refers to what a user own such as smart or ATM cards. ATM transaction is the simplest example for utilization of two factor authentication as it demands user or customer to know his/her PIN code or password and also user must have specific ATM card. However, soon it was realized that information stored in the smart card can be easily retrieved, therefore smart card based authentication schemes are not reliable and need to be reconsidered due to impersonation and password guessing attacks [29, 61, 130–132]. Similarly, smart card can be stolen/ lost and vulnerable to differential power analysis [28, 29]. The factors pertaining to three factor authentication are (1) what user knows (i.e password), (2) what user has (i.e smart card), and (3) what user is (i.e biometrics).

Consequently, introduction of biometrics not only resolved the issues related to two factor authentication because biometrics provides recognition on an inherent feature of human being. It also ensures the presence of a person to be authenticated at the time of authentication.

Moreover, password and smart card were incapable to differentiate between the attacker and the authentic user but the advent of biometrics makes it possible to differentiate between the two. Abundant biometric based schemes [133–137] has been presented that have syndicated the password and smart card as well. Awasthi et al. [138] presented biometric authentication scheme with nonce for TMIS. Lately, Mishra et al. [134] find out that offline password guessing attack is likely on this scheme and moreover their scheme fails to offer an appropriate password change option. Tan et al. [139] also declared that Awasthi et al.'s scheme is insecure against reflection attack and does not fulfill the criteria for delivering three factor security and user anonymity. Therefore, Tan et al. introduced an enhanced three factor authentication scheme and declared that their scheme is invincible against said attacks. Lately, Arshad and Nikooghadam [140] claimed that Tan et al.'s scheme is susceptible to replay and denial of service attacks. Consequently, Arshad and Nikooghadam proposed an authentication based scheme on elliptic curve cryptography in order to offer invincibility against replay and denial of service attacks. Unfortunately, the scheme proposed by Arshad and Nikooghadam is proved to be insecure against offline password guessing and patient impersonation attacks by Lu et al. [7]. Lu et al. then, put forwarded a biometric based three factor authentication scheme and claimed their scheme to offer irresistible security. However, in this chapter we establish that Lu et al.'s scheme is vulnerable to numerous attacks including (1) Patient anonymity violation attack, (2) Patient impersonation attack, and (3) TMIS server impersonation attack. Furthermore, their scheme does not provide patient untraceability. We then, proposed an improvement of Lu et al.'s scheme. To prove the security of proposed biometric based three factor authentication scheme, we have adopted the automated formal tool ProVerif. Rest of the chapter is organized as follows. Section 7.1 describes some fundamental concepts pertaining to this chapter. Section 7.2 reviews Lu et al.'s scheme. We perform cryptanalysis of Lu et al.'s scheme in section 7.3. Proposed three factor authentication scheme is illustrated in section 7.4, while the security validation of proposed scheme using automated tool ProVerif is performed in section 7.5. We perform security and performance comparisons of proposed scheme with related existing schemes in sections 7.6 and 7.7. Finally, chapter's summary is solicited in section 7.8.

7.1 Preliminaries

This section explains the notation guide, some fundamental concepts relating to biohashing and the common adversarial model.

7.1.1 Notation Guide

The notation guide pertaining to this chapter is illustrated in Table 7.1.

7.1.2 BioHashing

The biometrics offers a unique and quantifiable method for identification of a particular human. The use of biometrics is now very common for authentication. Although, the inherited problem of using biometrics is the noise encountered in each imprint resulting into false rejection of same biometrics. Fortunately, a number of biohashing techniques [30, 31, 141, 142] are proposed to cope with false rejection problem. BioHashing is a mapping of user's biometrics and specified pseudo random number tokens. BioHashing is verified to be the most suitable and compatible technique that can be utilized in tiny smart devices such as smart card and smart phone etc [30, 31, 141, 143].

7.1.3 Adversarial Model

In this chapter, we consider the common adversarial model as mentioned in [25–27]. Where according to capabilities of the adversary $\mathcal{A}$, following assumptions are made:

1. $\mathcal{A}$ fully controls the public communication channel. $\mathcal{A}$ can capture, replay, modify, insert a new message and can delete any message.
2. $\mathcal{A}$ can either access patient $\mathcal{U}_i$'s password or can steal his smart card, but not both alongside.
3. Any one having possession of a smart card can extract information stored in that smart card [28, 29].
4. $\mathcal{A}$ knows the public identities of all the users and the server.

7.2 Review of Lu et al.'s Scheme

This section elaborates Lu et al.'s authentication scheme. The scheme is illustrated in Fig. 7.1 and is explained following four phases:

Table 7.1: Notation Guide

Notations	Description	Notations	Description
$\|\|, \oplus$	Concatenation and XOR operators	$H(.)$	BioHashing operator
$h_1(.),\ h_2(.)$	two one-way hash functions	$\mathcal{S},\ \mathcal{U}_i$	TMIS Server, Patient
$ID_i,\ PW_i,\ B_i$	$\mathcal{U}_i$'s identity, Password, Biometrics	$x,\ K_{pub} = xP$	$\mathcal{S}$'s private/public key pair
AID_i	Dynamic identity of patient $\mathcal{U}_i$	$\mathcal{A}$	Adversary

7.2.1 Initialization

In this phase, the server $\mathcal{S}$ sets up its parameters, initially an elliptic curve $E_p(a, b)$ is selected, then, $\mathcal{S}$ selects an arbitrary base point P and two one way hash functions $h_1(.),\ h_2(.)$ along with bio-hashing operator $H(.)$. $\mathcal{S}$ then, generates his private key x. Finally, $\mathcal{S}$ publicizes $\{E_p(a, b)P, h_1(.), h_2(.), H(.)\}$, while he retains his private key x as secret.

7.2.2 Registration

During registration phase, the patient $\mathcal{U}_i$ selects his identity ID_i, password PW_i, then he imprints his biometrics B_i in specialized reader, further $\mathcal{U}_i$ computes and sends $MP_i = PW_i \oplus H(B_i)$ along with ID_i to TMIS server $\mathcal{S}$ using some secure channel. $\mathcal{S}$ upon reception of ID_i, MP_i computes $V_i = h_1(ID_i \| MP_i)$ and $AID_i = ID_i \oplus h_2(x)$, then $\mathcal{S}$ customizes a smart card with V_i, AID_i, h_1, h_2, H and sends the smart card to the patient $\mathcal{U}_i$ on some secure channel.

7.2.3 Login and Authentication

Step 1: $\mathcal{U}_i$ inserts his smart card in reader and inputs his password PW_i and identity ID_i. Then he imprints his biometric B_i. The smart card computes $h_1(ID_i || PW_i \oplus H(B_i))$ and checks its equivalence with stored V_i, if invalid smart card aborts the session. Otherwise, the smart card generates d_u to compute $K = h_1(ID_i \| ID_i \oplus AID_i)$, $M_1 = K \oplus d_u P$ and $M_2 = h_1(ID_i \| d_u P \| T_1)$. Then the smart card sends login/authentication request message $m_{i1} = \{AID_i, M_1, M_2, T_1\}$ to TMIS server $\mathcal{S}$.

Step 2: For the received login/authentication message m_{i1}, TMIS server $\mathcal{S}$ checks the freshness of timestamp T_1 by comparing it with current timestamp T_c. $\mathcal{S}$ terminates the session if T_1 is not fresh. Otherwise, $\mathcal{S}$ using his private key x computes $ID_i = AID_i \oplus h_2(x)$ and $d_u P = h_1(ID_i \| h_2(x)) \oplus M_1$. $\mathcal{S}$ checks $M_2 \stackrel{?}{=} h_1(ID_i \| d_u P \| T_1)$, if it is false $\mathcal{S}$ aborts the session. Otherwise, $\mathcal{S}$ generates random d_s and fresh timestamp T_2

Patient $\mathcal{U}_i$	*TMIS Server* $\mathcal{S}$
Registration Phase:	
Choose ID_i, PW_i, imprint Biometric B_i	
Compute $MP_i = PW_i \oplus H(B_i)$	
$\xrightarrow{ID_i, MP_i}$	
	Computes $V_i = h_1(ID_i \| MP_i)$
	Computes $AID_i = ID_i \oplus h_2(x))$
	stores V_i, AID_i in $smart-card$
$\xleftarrow{smart-card}$	
$Smart-card = \{V_i, AID_i, h_1, h_2, H\}$	
Login & Authentication Phase:	
Imprints Biometric B_i and enters ID_i,PW_i	
Verifies $V_i \stackrel{?}{=} h_1(ID_i \| PW_i \oplus H(B_i))$, abort if not true	
$K = h_1(ID_i \| ID_i \oplus AID_i)$	
Generates random d_u	
$M_1 = K \oplus d_u P$	
$M_2 = h_1(ID_i \| d_u P \| T_1)$	
$\xrightarrow{m_{i1}=\{AID_i, M_1, M_2, T_1\}}$	
	$T_c - T_1 \leq \Delta T$, abort if not fresh
	$ID_i = AID_i \oplus h_2(x)$
	$d_u P = h_1(ID_i \| h_2(x)) \oplus M_1$
	Check $M_2 \stackrel{?}{=} h_1(ID_i \| d_u P \| T_1)$, aborts if not valid
	Select d_s, T_2
	$SK = d_s d_u P$
	$M_3 = K \oplus d_s P$
	$M_4 = h_1(K \| d_u P \| SK \| T_2)$
$\xleftarrow{m_s=\{M_3, M_4, T_2\}}$	
$T_c - T_2 \leq \Delta T$, abort if not fresh	
$d_s P = M_3 \oplus K$	
$SK = d_u d_s P$	
$M_4 \stackrel{?}{=} h_1(K \| d_u P \| SK \| T_2)$	
Generate T_3, $M_5 = h_1(K \| d_s P \| SK \| T_3)$	
Accept SK as session key	
$\xrightarrow{m_{i2}=\{M_5, T_3\}}$	
	$T_c - T_3 \leq \Delta T$, abort if not fresh
	$M_5 \stackrel{?}{=} h_1(K \| d_s P \| SK \| T_3)$

Figure 7.1: Lu et al.'s Authentication Scheme

then, computes $SK = d_s d_u P$, $M_3 = K \oplus d_s P$ and $M_4 = h_1(K \| d_u P \| SK \| T_2)$. $\mathcal{S}$ then sends challenge message $m_s = \{M_3, M_4, T_2\}$ to $\mathcal{U}_i$.

Step 3: For the received message m_s, the patient $\mathcal{U}_i$ first verifies the freshness of T_2. Then computes $d_s P = M_3 \oplus K$ and session key $SK = d_u d_s P$. Further, $\mathcal{U}_i$ computes $h_1(K \| d_u P \| SK \| T_2)$ and checks its equivalence with M_4. if it is true, $\mathcal{U}_i$ accepts SK as shared session key and computes $M_5 = h_1(K \| d_s P \| SK \| T_3)$. Finally, $\mathcal{U}_i$ sends response message $m_{i2} = \{M_5, T_3\}$ to TMIS server $\mathcal{S}$.

Step 4: For the received response message m_{i2}, TMIS server $\mathcal{S}$ verifies freshness of T_3, if it is fresh, $\mathcal{S}$ computes $h_1(K \| d_s P \| SK \| T_3)$ and checks its equivalence with M_5, if it is valid $\mathcal{S}$ trusts $\mathcal{U}_i$ as legal patient and keeps SK as the shared key.

7.2.4 Password Change

In Lu et al.'s scheme, the patient $\mathcal{U}_i$ can freely change his password without intervention of TMIS server $\mathcal{S}$. For changing password, $\mathcal{U}_i$ inserts his smart card, imprints his biometrics and enters his password and identity. The smart card then checks $V_i \stackrel{?}{=} h_1(ID_i || PW_i \oplus H(B_i))$, if correct, smart card asks for new password, $\mathcal{U}_i$ enters PW_i^{new} and the smart card computes $V_i^{new} = h_1(ID_i \| PW_i^{new} \oplus H(B_i))$ and assigns V_i^{new} to V_i.

7.3 Cryptanalysis of Lu et al.'s Scheme

This section presents the weaknesses of Lu et al.'s authentication scheme. We prove that Lu et al.'s scheme is vulnerable to a number of attacks. We also show that Lu et al.'s scheme does not fulfill patient untraceability. Following subsections describe the weaknesses of Lu et al.'s scheme.

7.3.1 Patient Anonymity Violation Attack

This section shows that a dishonest patient $\mathcal{A}$ can easily break patient's anonymity. $\mathcal{A}$ will perform following steps to launch patient anonymity violation attack:

Step PAV 1: An adversary $\mathcal{A}$ registers with the system and gets personalized smart card containing $\{V_a, AID_a, h_1, h_2, H\}$.

Step PAV 2: $\mathcal{A}$ extracts $\{V_a, AID_a, h_1, h_2, H\}$ stored in his smart card by means of power analysis [28, 29]. $\mathcal{A}$ then submits his password PW_a, identity ID_a and biometrics B_a and computes:

$$h_2(x) = AID_a \oplus ID_a \tag{7.1}$$

Step PAV 3: When an honest patient $\mathcal{U}_i$ pledges the authentication request message $m_{i1} = \{AID_i, M_1, M_2, T_1\}$. $\mathcal{A}$ captures the request message m_{i1}, and computes:

$$ID_i = AID_i \oplus h_2(x) \tag{7.2}$$

In Eq. 7.2, ID_i is the real identity of $\mathcal{U}_i$. Hence, $\mathcal{A}$ has successfully violated $\mathcal{U}_i$'s anonymity.

7.3.2 Patient Impersonation Attack

This subsection presents the verdict that a dishonest patient $\mathcal{A}$ can easily impersonate as another legal patient $\mathcal{U}_i$. Following steps will be executed between $\mathcal{A}$ and $\mathcal{S}$ for a successful impersonation attack:

Step PIA 1: $\mathcal{A}$ first gets patient $\mathcal{U}_i$'s identity ID_i as mentioned in subsection 7.3.1.

Step PIA 2: $\mathcal{A}$ generates random d_a and computes:

$$K_a = h_1(ID_i \| ID_i \oplus AID_i) \tag{7.3}$$

$$M_1 = K_a \oplus d_a P \tag{7.4}$$

$$M_2 = h_1(ID_i \| d_a P \| T_{a1}) \tag{7.5}$$

Step PIA 3: $\mathcal{A}$ sends authentication request message $m_{a1} = \{AID_i, M_1, M_2, T_{a1}\}$ to $\mathcal{S}$.

Step PIA 4: $\mathcal{S}$ upon reception of m_{a1} checks the validity of timestamp T_{a1}, as it is freshly generated by $\mathcal{A}$, so $\mathcal{S}$ computes:

$$ID_i = h_2(x) \oplus AID_i \tag{7.6}$$

$$d_a P = h_1(ID_i \| h_2(x)) \oplus M_1 \tag{7.7}$$

Step PIA 5: $\mathcal{S}$ further verifies $M_2 \stackrel{?}{=} h_1(ID_i \| d_a P \| T_{a1})$ and finds it correct. $\mathcal{S}$ then generates

d_s, T_2 and computes:

$$SK = d_s d_a P \tag{7.8}$$

$$M_3 = K_a \oplus d_s P \tag{7.9}$$

$$M_4 = h_1(K \| d_a P \| SK \| T_2) \tag{7.10}$$

Step PIA 6: $\mathcal{S}$ sends challenge message $m_s = \{M_3, M_4, T_2\}$ to $\mathcal{U}_i$.

Step PIA 7: $\mathcal{A}$ captures m_s and computes:

$$d_s P = M_3 \oplus K_a \tag{7.11}$$

$$SK = d_u d_s P \tag{7.12}$$

$$M_5 = h_1(K_a \| d_s P \| SK \| T_{a3}) \tag{7.13}$$

Step PIA 8: $\mathcal{A}$ then sends response message $m_{a2} = \{M_5, T_{a3}\}$ to $\mathcal{S}$.

Step PIA 9: For the received message m_{a2}, $\mathcal{S}$ checks freshness of T_{a3} and $M_5 \stackrel{?}{=} h_1(K_a \| d_s P \| SK \| T_{a3})$, as both are valid. $\mathcal{S}$ accepts the adversary $\mathcal{A}$ as a legal patient $\mathcal{U}_i$. Hence, the adversary $\mathcal{A}$ has successfully impersonated to $\mathcal{S}$ on behalf of $\mathcal{U}_i$.

7.3.3 TMIS Server Impersonation Attack

This subsection elaborates the vulnerability of Lu et al.'s scheme to TMIS server $\mathcal{S}$'s impersonation attacks. We show that a dishonest patient $\mathcal{A}$ can easily impersonate as TMIS server $\mathcal{S}$ to deceive other legal patients. Following steps will be executed between a legal patient $\mathcal{U}_i$ and $\mathcal{A}$ for successful impersonation attack:

Step SIA 1: $\mathcal{A}$ extracts $h_2(x)$ from his smart card as described in subsection 7.3.1, and waits for authentication request message by some other legal patient.

Step SIA 2: When a legal patient $\mathcal{U}_i$ initiates the authentication request message $m_{i1} = \{AID_i, M_1, M_2, T_1\}$ to $\mathcal{S}$. The adversary $\mathcal{A}$ captures the message and generates random

number d_a, timestamp T_a and computes:

$$ID_i = AID_i \oplus h_2(x) \tag{7.14}$$
$$d_uP = h_1(ID_i \| h_2(x)) \tag{7.15}$$
$$SK = d_a d_u P \tag{7.16}$$
$$M_3 = K \oplus d_aP \tag{7.17}$$
$$M_4 = h_1(h_1(ID_i \| ID_i \oplus AID_i) \| d_uP \| SK \| T_a) \tag{7.18}$$

Then $\mathcal{A}$ sends challenge message $m_a = \{M_3, M_4, T_a\}$ to $\mathcal{U}_i$.

Step SIA 3: $\mathcal{U}_i$ upon reception of m_a, first verifies the freshness of timestamp T_a, as it was freshly generated by adversary, so $\mathcal{U}_i$ will compute:

$$d_aP = M_3 \oplus K \tag{7.19}$$
$$SK = d_u d_a P \tag{7.20}$$

$\mathcal{U}_i$ then checks the validity of $M_4 \stackrel{?}{=} h_1(h_1(ID_i \| ID_i \oplus AID_i) \| d_uP \| SK \| T_a)$, and finds it correct. Hence, $\mathcal{A}$ passes this test.

Step SIA 4: Finally, $\mathcal{U}_i$ accepts SK as shared session key and will generate T_3 to compute $M_5 = h_1(K \| d_aP \| SK \| T_3)$ and sends $m_{i2} = \{M_3, T_3\}$ to TMIS server, which in turns accepts SK as session key.

Hence, the adversary $\mathcal{A}$ has impersonated as legal TMIS server $\mathcal{S}$ and deceived $\mathcal{U}_i$.

7.3.4 Patient Untraceability

An authentication scheme is said to provide user/patient untraceability, if no adversary can recognize whether two different sessions are initiated by the same user. In Lu et al.'s scheme, the patient sends AID_i as his pseudo identity, AID_i remains same for several sessions, so an adversary by just analyzing the channel can differentiate whether or not two sessions are initiated by same user by just comparing the pseudo identities sent in each session. Hence, Lu et al.'s scheme does not provide patient untraceability.

7.4 Proposed Scheme

In this section, we explain our improved biometric based three factor authentication scheme. While designing our improvement, we consider the reasons effecting the security of Lu et al.'s scheme, as it can be easily verified that the security of Lu et al.'s scheme is relied on a generic parameter $h_2(x)$, so any adversary $\mathcal{A}$ registered to the system can easily extract $h_2(x)$ from his own smart card. Then $\mathcal{A}$ can easily launch numerous attacks as mentioned in section 7.3 on Lu et al.'s scheme. Hence, to improve Lu et al.'s scheme, we have alternated the use of $h_2(x)$ by $h_2(ID_i\|x)$. Further, we have modified some of the steps in registration and authentication phases, while password change phase is taken from Lu et al.'s scheme in its present form. We have illustrated the proposed scheme in Fig. 7.2 as well as in following subsections:

7.4.1 Initialization

The initialization phase is very similar to Lu et al.'s initialization phase, where TMIS server $\mathcal{S}$ selects an elliptic curve $E_p(a,b)$, an arbitrary base point P and two one way hash functions $h_1(.),\ h_2(.)$ along with biohashing operator $H(.)$ and his private key x. Then $\mathcal{S}$ publicizes $\{E_p(a,b), P, h_1(.), h_2(.), H(.)\}$, additionally in the proposed scheme $\mathcal{S}$ also computes and publishes his public key $K_{pub} = xP$.

7.4.2 Registration

Patient $\mathcal{U}_i$ initiates this phase to register with the system in order to acquire remote healthcare services. $\mathcal{U}_i$ first selects his identity ID_i, password PW_i, then he imprints his biometrics B_i in specialized reader, further $\mathcal{U}_i$ computes and sends $MP_i = PW_i \oplus H(B_i)$ along with ID_i to TMIS server $\mathcal{S}$ using some secure channel. $\mathcal{S}$ upon reception of ID_i, MP_i computes $V_i = h_1(ID_i\|MP_i)$ and $W_i = MP_i \oplus h_2(ID_i\|x)$. Then $\mathcal{S}$ customizes a smart card with $\{V_i, W_i, K_{pub}h_1, h_2, H\}$ and sends the smart card to the patient $\mathcal{U}_i$ via some secure channel.

7.4.3 Login and Authentication

Step PA 1: $\mathcal{U}_i$ inserts his smart card in the reader and inputs his password PW_i and identity ID_i. Then he imprints his biometric B_i. The smart card computes $h_1(ID_i||PW_i \oplus H(B_i))$ and checks its equivalence with stored V_i, if invalid smart card aborts the session.

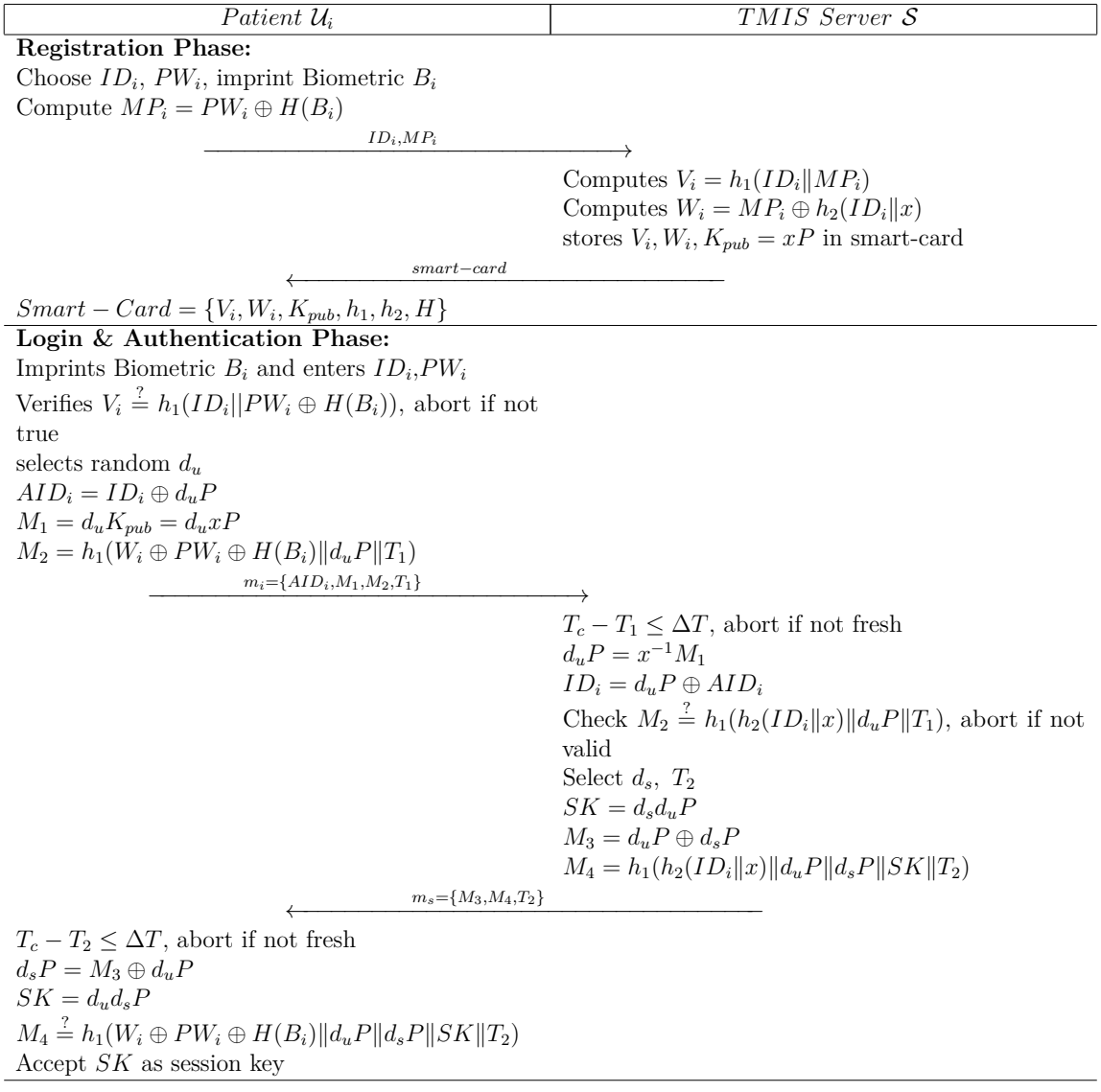


Figure 7.2: Proposed Authentication Scheme

Otherwise, smart card generates d_u and fresh timestamp T_1 to compute $AID_i = ID_i \oplus d_uP$, $M_1 = d_uK_{pub} = d_uxP$ and $M_2 = h_1(W_i \oplus PW_i \oplus H(B_i)\|d_uP\|T_1)$. Then smart card sends login/ authentication request message $m_i = \{AID_i, M_1, M_2, T_1\}$ to TMIS server $\mathcal{S}$.

Step PA 2: For the received login/authentication message m_i, TMIS server $\mathcal{S}$ checks the freshness of timestamp T_1 by comparing it with current timestamp T_c, terminates the session if T_1 is not fresh. Otherwise, $\mathcal{S}$ using his private key x computes $d_uP = x^{-1}M_1$ and $ID_i = AID_i \oplus d_uP$. Furthermore, $\mathcal{S}$ checks $M_2 \stackrel{?}{=} h_1(h_2(ID_i\|x)\|d_uP\|T_1)$, if it is false $\mathcal{S}$ aborts the session. Otherwise, $\mathcal{S}$ generates random d_s and fresh timestamp T_2, then computes $SK = d_sd_uP$, $M_3 = d_uP \oplus d_sP$ and $M_4 = h_1(h_2(ID_i\|x)\|d_uP\|d_sP\|SK\|T_2)$. $\mathcal{S}$ then sends challenge message $m_s = \{M_3, M_4, T_2\}$ to $\mathcal{U}_i$.

Step PA 3: For the received message m_s, the patient $\mathcal{U}_i$ first verifies the freshness of T_2. Then $\mathcal{U}_i$ computes $d_sP = M_3 \oplus d_uP$ and session key $SK = d_ud_sP$. Further, $\mathcal{U}_i$ computes $h_1(W_i \oplus PW_i \oplus H(B_i)\|d_uP\|d_sP\|SK\|T_2)$ and checks its equivalence with received M_4, if true, $\mathcal{U}_i$ accepts SK as shared session key and $\mathcal{S}$ as the intended legal TMIS server.

7.4.4 Password Change

Similar to Lu et al.'s scheme, the password in proposed scheme can be freely changed without intervention of TMIS server $\mathcal{S}$. For changing password, $\mathcal{U}_i$ inserts his smart card, imprints his biometrics and enters his password and identity. The smart card then computes $MP_i = PW_i \oplus H(B_i)$ and checks $V_i \stackrel{?}{=} h_1(ID_i||MP_i)$, if it is correct, the smart card asks for new password. $\mathcal{U}_i$ enters PW_i^{new} smart card then computes $MP_i^{new} = PW_i^{new} \oplus HB_i$ and $V_i^{new} = h_1(ID_i\|MP_i^{new})$ and $W_i^{new} = W_i \oplus MP_i \oplus\ MP_i^{new}$. Finally, smart card assigns V_i^{new} to V_i and W_i^{new} to W_i.

7.5 Formal Security Validation using ProVerif

In this subsection, we prove the security of proposed scheme using automated formal tool ProVerif [34, 35, 68]. To demonstrate proposed scheme's security, we have modeled the steps illustrated in section 7.4 and Fig. 7.2. The formal verifier model of ProVerif is consisting of three parts (1) declaration part; (2) process part; and (3) main part as shown in Fig. 7.3. In declaration part all the names, variables and channels along with cryptographic functions are defined. All processes and subprocesses are modeled in process part while the investigating

Table 7.2: Security Analysis

Scheme→ Security Properties↓	Our	[7]	[140]	[139]	[138]	[144]
Mutual Authentication	Yes	Yes	No	Yes	Yes	Yes
User/Patient Anonymity	Yes	No	Yes	Yes	No	Yes
Perfect Forward Secrecy	Yes	Yes	Yes	No	Yes	Yes
Replay Attack	Yes	Yes	Yes	No	Yes	Yes
Impersonation Attack	Yes	No	No	Yes	Yes	Yes
Privileged Insider Attack	Yes	Yes	Yes	Yes	Yes	Yes
Man-in-Middle Attack	Yes	No	Yes	Yes	Yes	Yes
Offline Password Guessing Attack	Yes	Yes	No	Yes	Yes	No

scheme is modeled in main part. We have defined two channels, names and variables while cryptographic functions are also defined as constructors and equations in declaration part. In Process part, we define two processes *PatientUi* and *TMISserverS* for each participant i.e. the patient and TMIS server respectively. In main part, we model the start and end event of each patient and server process. Further, we also model the scheme as parallel execution of both patient and server processes. Finally, to verify the correctness of the proposed scheme and secrecy of the session key, we applied three queries and the results are as follows:

1. RESULT inj-event(end˙Server(id)) ==> inj-event(begin˙Server(id)) is true.
2. RESULT inj-event(end˙Patient(id˙2124)) ==> inj-event(begin˙Patient(id˙2124)) is true.
3. RESULT not attacker(SK[]) is true.

The results (1) and (2) verifies that the server and patient processes started and terminated successfully, which confirms the correctness of proposed scheme. The result (3) shows that attacker query on session key SK is not successful, which confirms the secrecy property of the proposed scheme.

7.6 Security Analysis

In this section, we analyze the security of our improved authentication scheme considering the same adversarial model as described in subsection 2.2.6. In following subsections, we show that the proposed scheme is robust against all known attacks.

```
(***************** Channels ***************)
free ChSec:channel [private].
free ChPub:channel.
(************ Names & Variables ************)
const P:bitstring.
free x:bitstring [private].
free Kpub:bitstring.
free IDi:bitstring.
free PWi:bitstring [private].
free Bi:bitstring [private].
free SK:bitstring [private].
(**Constructors*destructors*Equations **)
fun H(bitstring):bitstring.
fun h1(bitstring):bitstring.
fun h2(bitstring):bitstring.
fun ECPM(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun Inverse(bitstring):bitstring.
fun Mult(bitstring,bitstring):bitstring.
equation forall a:bitstring; Inverse(Inverse(a)
    )=a.
equation forall a:bitstring,b:bitstring; XOR(
    XOR(a,b),b)=a.
(******************events******************)
event begin_Patient(bitstring).
event end_Patient(bitstring).
event begin_Server(bitstring).
event end_Server(bitstring).
```

(a) Declarations

```
(********Process Replication************)
process ( (!TMISServerS) | (!PatientUi) )
(************* *queries* *************)
query attacker(SK).
query id:bitstring; inj event(end_Patient(id))
    ==> inj event(begin_Patient(id)) .
query id:bitstring; inj event(end_Server(id))
    ==> inj event(begin_Server(id)) .
```

(c) Main

```
(***************Patient Process***************)
let PatientUi =
(***************** Registration***************)
let MPi = XOR(PWi,H(Bi)) in
out(ChSec,(IDi,MPi));
in(ChSec,(xVi:bitstring,xWi:bitstring,xKpub:
    bitstring));
(************Login & Authentication***********)
event begin_Patient (IDi);
let Vi = h1(XOR(Concat(IDi,PWi),H(Bi))) in
if(Vi = xVi) then
new du:bitstring;
new T1:bitstring;
let AIDi = XOR(IDi,ECPM(du,P)) in
let M1 = ECPM(du,Kpub) in
let M2 = Concat(XOR(xWi,(PWi,H(Bi))),(ECPM(du,P
    ),T1)) in
out(ChPub,(AIDi,M1,M2,T1));
in(ChPub,(xM3:bitstring,xM4:bitstring,xT2:
    bitstring));
let dsP = XOR(xM3,ECPM(du,P)) in
let SK = ECPM(du,dsP) in
let M4 = h1(Concat(XOR(xWi,(PWi,H(Bi))),(ECPM(
    du,P),SK,xT2))) in
if(M4 = xM4) then
event end_Patient(IDi);    0.
(***************** TMIS Server *****************)
(***************** Registration*****************)
let TMISServerS =
(*Initialization*)
let Kpub = Mult(x,P)in
event begin_Server(x);
in(ChSec,(xIDi:bitstring,xMPi:bitstring));
let Vi = h1(Concat(xIDi,xMPi)) in
let Wi = XOR(xMPi,h2(Concat(xIDi,x))) in
out(ChSec,(Vi,Wi,Kpub));
(************Login & Authentication***********)
in(ChPub,(xAIDi:bitstring,xM1:bitstring,xM2:
    bitstring,xT1:bitstring));
let duP = Mult(Inverse(x),xM1) in
let IDi' = XOR(duP,xAIDi) in
let M2 = h1(Concat(h2(Concat(IDi',x)),(duP,xT1)
    )) in
if(M2 = xM2) then
new ds:bitstring;
new T2:bitstring;
let SK = ECPM(ds,duP) in
let M3 = XOR(duP,ECPM(ds,P)) in
let M4 = h1(Concat(h2(Concat(IDi',x)),(duP,ECPM
    (ds,P),SK,T2))) in
out(ChPub,(M3,M4,T2));
event end_Server(x)
else 0.
```

(b) Processes

Figure 7.3: ProVerif Validation

7.6.1 Mutual Authentication

In proposed scheme, $\mathcal{S}$ validates $\mathcal{U}_i$ by verifying $M_2 \stackrel{?}{=} h_1(h_2(ID_i\|x)\|d_uP\|T_1)$, which requires to compute $h_2(ID_i\|x)$. Furthermore, $h_2(ID_i\|x)$ can be computed as $h_2(ID_i\|x) = W_i \oplus PW_i \oplus H(B_i)$, where W_i is stored in smart card. Hence, to generate valid $h_2(ID_i\|x)$ the adversary needs $\mathcal{U}_i$'s smart card as well as his password PW_i and biometrics B_i. $\mathcal{U}_i$ authenticates $\mathcal{S}$ by verifying $M_4 \stackrel{?}{=} h_1(h_2(ID_i\|x)\|d_uP\|SK\|T_2)$. Only intended legal TMIS server $\mathcal{S}$ can pass this test because $h_2(ID_i\|x)$ requires secret key x of TMIS server also d_uP can be extracted from M_1 using $\mathcal{S}$'s secret key. Hence, both $\mathcal{U}_i$ and $\mathcal{S}$ mutually authenticates each other.

7.6.2 User Anonymity

In proposed schemes patients, instead of patient's identity ID_i, a dynamic pseudo identity AID_i is sent in authentication request message to $\mathcal{S}$. Further, patient's pseudo identity AID_i is dynamically generated in each session. Patient's real identity ID_i can only be extracted by the use of TMIS server $\mathcal{S}$'s private key x. Hence, the improved scheme ensures patient's anonymity and untraceability.

7.6.3 Replay Attack

Very similar to Lu et al.'s scheme, in authentication request message a timestamp T_1 is sent in plaintext as well as embedded in M_2. If an adversary replays a previous message, $\mathcal{S}$ can easily detect replay attack by just verifying freshness of T_1. If an adversary sent fresh timestamp T_a along with previously generated M_2 then the request message will not pass the test $M_2 \stackrel{?}{=} h_1(h_2(ID_i\|x)\|d_uP\|T_1)$. Same is the case if the attacker captures patient's request message and replays TMIS server's previously sent message. Hence, proposed scheme withstand the replay attack.

7.6.4 Impersonation Attack

To impersonate as $\mathcal{U}_i$, the adversary $\mathcal{A}$ must generate valid login message $m_i = \{AID_i, M_1, M_2, T_1\}$. Similarly, to impersonate as TMIS server, $\mathcal{A}$ must be able to generate valid response message $m_s = \{M_3, M_4, T_2\}$. As we have already described in subsection 6.4.1.1 that m_i can be generated by using $\mathcal{U}_i$'s smart card as well his password PW_i and biometrics B_i. Similarly,

to generate valid m_s, $\mathcal{A}$ needs $\mathcal{S}$'s secret key x. Hence, proposed scheme resists patient as well as TMIS server impersonation attacks.

7.6.5 Privileged Insider Attack

For registration, the patient $\mathcal{U}_i$ sends $MP_i = PW_i \oplus H(B_i)$, no insider is having access to patient's password PW_i or biometrics B_i. Moreover, $\mathcal{S}$ does not store any verifier table, for authentication purposes $\mathcal{S}$ uses his own private key x. Hence, privileged insider and stolen verifier attacks are not viable on proposed scheme.

7.6.6 Man-in-middle Attack

$\mathcal{A}$ can execute man in middle attack if he becomes able to pass authentication test from both $\mathcal{S}$ and $\mathcal{U}_i$. Since it has been shown in subsection 6.4.1.1 that the adversary can only pass authentication from $\mathcal{U}_i$, if he holds $\mathcal{S}$'s secret key x. Likewise, $\mathcal{A}$ can pass authentication from $\mathcal{S}$, if he possesses $\mathcal{U}_i$'s smart card, password PW_i and biometrics B_i. Therefore, proposed scheme resists man in middle attack.

7.6.7 Offline Password Guessing Attack

Suppose $\mathcal{A}$ by some means got $\mathcal{U}_i$'s smart card and read the information $\{W_i,\ V_i\}$ from $\mathcal{U}_i$'s smart card. Then, to guess PW_i, he needs to know identity ID_i, the biometric B_i. Hence, offline password guessing attack is not viable on proposed scheme.

7.6.8 Perfect Forward Secrecy

In proposed scheme, if an adversary becomes able to acquire $\mathcal{U}_i$'s password or $\mathcal{S}$'s secret key, he will still be unable to compute previous session keys, as in proposed scheme the computation of session key $SK = d_u d_s P$ requires session specific d_u entered by $\mathcal{U}_i$ and d_s contributed by $\mathcal{S}$, without knowing session specific d_u and d_s, the adversary could not find session key. Hence, proposed scheme ensures perfect forward secrecy.

Table 7.3: Computation, communication and Memory Analysis

Scheme→	Our	[7]	[140]	[139]	[138]	[144]
Memory	60	40	140	80	80	296
Communication	140	180	200	100	100	484
Messages	2	3	3	2	2	3
Computation						
Registration	$3T_{owh}$	$3T_{owh}$	$4T_{owh}$	$3T_{owh}$	$3T_{owh}$	$3T_{owh}$
Authentication	$4T_{epm} + 7T_{owh}$	$4T_{epm} + 11T_{owh}$	$4T_{epm}$+$15T_{owh}$+ $2T_{mm} + 1T_{min}$	$6T_{epm} + 11T_{owh}$	$6T_{epm} + 9T_{owh}$	$1T_{mm} + 4T_{sde} + 8T_{me} + 1T_F + 5T_{owh}$
Password Change	$3T_{owh}$	$3T_{owh}$	$4T_{owh}$	$4T_{owh}$	$4T_{owh}$	$4T_{owh}$

7.7 Performance Analysis

Here we perform the comparative performance analysis of our scheme with recent related existing schemes [7, 138–140, 144]. For performance evaluation, we consider the memory required in smart card, the computation and communication overheads.

7.7.1 Comparative Computation Analysis

Following notations are introduced for comparative performance analysis:

- T_{owh} : time to compute a one-way hash function
- T_{me} : time to compute a modular exponentiation operation
- T_{epm} : time to compute a ECC point multiplication
- T_{min} : time to compute a modular inversion
- T_{sde} : time to compute a symmetric encryption/ decryption operation

We have summarized comparative computational cost analysis in Table 7.3. Proposed scheme has reduced $4T_{owh}$ operation during login and authentication scheme as compared with Lu et al.'s scheme. It is well understood that $T_{me} >> T_{epm}$. Hence, proposed scheme incurs the least computational overhead when compared with related recent existing schemes [7, 138–140, 144].

7.7.2 Communication Overhead and Smart Card Memory Analysis

We have summarized the memory (bytes) required in smart card and bytes exchanged during authentication phase of proposed and related schemes [7, 138–140, 144] in Table 7.3. For

simplicity, we have assumed the size of identities, timestamps and hash digest as 20 bytes. We have also taken into consideration the NIST recommended sizes for ECC and RSA which are 20 bytes and 128 bytes, respectively. The proposed scheme requires 20 extra bytes to store server's public key K_{pub} in smart card, while it saves transmission of 40 bytes during login and authentication phase. Furthermore, the proposed schemes achieves the security and privacy in only 2 messages. Hence, proposed scheme is more efficient in terms of communication overhead as compared with Lu et al.'s scheme.

7.8 Chapter Summary

In this chapter, we have investigated Lu et al.'s biometric based three factor authentication scheme for TMIS. We have proved that Lu et al.'s scheme cannot resist (1)patient anonymity violation attack, (2) patient impersonation attack, and (3) TMIS server impersonation attack. Furthermore, Lu et al.'s scheme does not provide patient untraceability. To overcome the weaknesses of Lu et al.'s scheme, we have proposed an improved scheme. We have analyzed the security of proposed scheme using formal automated tool ProVerif. The proposed scheme while resisting all known attacks is also more lightweight in terms of computation and communication overheads.

Chapter 8

A Biometric Based Three-factor Authentication Scheme using Symmetric Key Cryptography for TMIS

Due to demerits of password based authentication schemes using smart card as discussed in chapter 7 and [29, 36, 42, 42, 50, 59, 61, 63, 130–132, 145–147]. Various three-factor biometrics based authentication schemes are adopted in [140, 148–150]. three-factor based authentication schemes ensure improved security and also offer authenticity and integrity of exchanged information [138, 151–153]. Since asymmetric key based authentication schemes [36, 42, 50, 59, 63, 146, 147] are considered to be more secure in which each user holds different but dependent keys. This element of dependency ensure secure and protected communication against well-known attacks. However, asymmetric scheme's basic operations such as exponentiation, point multiplication and pairing etc. makes it impractical for resource constrained scenarios or systems. On the other hand symmetric key based authentication schemes in which shared key is used [94, 101, 154] are considered to be more suitable for resource constrained systems due to lightweight and less computation intensive basic operations such as symmetric encryption, MAC, XOR and hash etc. But such schemes are either vulnerable to different attacks or having correctness problems [84, 85].

In 2013, Yan et al. [155] presented an enhanced scheme against denial of services (DoS) attack because they find Tan's [149] scheme susceptible against DoS attack. Very soon Mir and Nikooghadam [8] pointed out that Yan et al. scheme is insecure due to various possible attacks

such as impersonation and offline password guessing attacks. Moreover, Yan et al's scheme fails to provide forward secrecy and its password change phase is unexpectedly inefficient. Mir and Nikooghadam [8] then proposed an improved anonymous three-factor authentication scheme based on lightweight symmetric key primitives and claimed their scheme to be secure. However, in this chapter, we have presented an in-depth analysis of Mir and Nikooghadam's scheme and find out that smart card stolen/lost attack is possible. Moreover, despite the claim of Mir and Nikooghadam that their scheme provides user anonymity we prove that user anonymity violation attack is still possible on Mir and Nikooghadam's scheme. Then we proposed an improved three-factor authentication scheme based on only lightweight symmetric key primitives. The proposed scheme is provably secure against active adversaries. We have also substantiated the security of proposed scheme using automated formal tool ProVerif [34–36].

Rest of the chapter is organized as follows, in section 8.1 we review Mir and Nikooghadam's scheme, while cryptanalysis of Mir and Nikooghadam's scheme is performed in section 8.2. Proposed supplemented scheme is described in Section 8.3, we have analyzed the security of our scheme in Section 8.4. Section 8.5 verifies the security using automated tool ProVerif, the performance comparison is performed in Section 8.6. Finally, chapter's summary is solicited in Section 8.7.

8.1 Review of Mir and Nikooghadam's Scheme

In this section, we present a review of Mir and Nikooghadam's scheme [8]. The scheme of Mir and Nikooghadam is comprised of four phases: (1) The Registration Phase; (2) The Login Phase; (3) The Authentication and Key agreement; and (4)The Password and Biometrics Change Phase. These phases are discussed in details as under:

8.1.1 The Registration Phase

The registration of a particular patient is a three step process. The patient $\mathcal{P}_i$ selects his unique identity ID_{pi} and password PW_{pi} besides random number N_{pi}. Patient $\mathcal{P}_i$ engrave his/her biometrics B_{pi} and then it determines $\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$. The patient then transmits registration request $\{\overline{PW}_{pi}, ID_{pi}\}$ to server $\mathcal{S}$ via secure channel. The server $\mathcal{S}$ calculates $X_{pi} = h(ID_{pi}\|x_s)$,$Y_{pi} = X_{pi} \oplus \overline{PW}_{pi}$ and $M_{pi} = h(\overline{PW}_{pi}\|X_{pi}\|ID_{pi})$ in response to registration request. The server $\mathcal{S}$ also produces parameter a that is secret and finds

Table 8.1: Notation Guide

Notations	Description	Notations	Description
$\mathcal{S}$	Server	$\mathcal{P}_i$	The legal client
ID_{pi}	Identity of P_i	$\mathcal{A}$	The Adversary
PW_{pi}	Password of patient N_{pi}	b_i	Unique random number of $\mathcal{P}_i$
a	Secret keys of $\mathcal{S}$	$\|$	String concatenation operator
t_1	timestamp of $\mathcal{P}_i$	t_3	timestamp of $\mathcal{S}$
$\oplus$	Bitwise XOR operation	$h(.)$	A one way hash function
SC_{pi}	$\mathcal{P}_i$'s smart card	x_s	Server's private key

out $C_{pi} = h(a\|x_s) \oplus \overline{PW}_{pi}$. The server $\mathcal{S}$ also keeps the status bit to show the status of patient. At the end the server $\mathcal{S}$ inserts $\{Y_{pi}, C_{pi}, h(.), M_{pi}\}$ into smart card SC_{pi}, which is then sent back to $\mathcal{P}_i$ through secure channel. The patient $\mathcal{P}_i$ gets the SC_{pi} and computes $g_{pi} = B_{pi} \oplus h(PW_{pi}\|ID_{pi})$ and $E_{pi} = N_{pi} \oplus h(ID_{pi}\|PW_{pi})$. The registration phase ends up when patient $\mathcal{P}_i$ inserts g_{pi} and E_{pi} into SC_{pi}. Hence, SC_{pi} contains the values of $\{Y_{pi}, h(.), g_{pi}, E_{pi}, C_{pi}, M_{pi}\}$.

8.1.2 Login Phase

The login process completes in two phases that are as follows

Step LP 1: $\mathcal{P}_i$ pushes his/her SC_{pi} into card reader and enters his her identity ID_{pi} and password PW_{pi} and provides his/her biometric scan B^*_{pi}. The smart card SC_{pi} calculates $B_{pi} = g_{pi} \oplus h(ID_{pi}\|PW_{pi})$, then checks $d(B_{pi}, B^*_{pi}) \geq \gamma$ if condition is true then session is aborted, otherwise computes $N_{pi} = E_{pi} \oplus h(ID_{pi}\|PW_{pi})$, $\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$, $X_{pi} = Y_{pi} \oplus \overline{PW}_{pi}$, $M'_{pi} = h(\overline{PW}_{pi}\|X_{pi}\|ID_{pi})$. Further, it checks $M'_{pi} \stackrel{?}{=} M_{pi}$, if it does not hold, the session terminates. Otherwise, ID_{pi} and PW_{pi} are detected as valid information.

Step LP 2: SC_{pi} produces b_i as random number and calculates $Z = \overline{PW}_{pi} \oplus C_{pi} = h(a\|x_s)$, $DID_{pi} = ID_{pi} \oplus h(Z)$, $G_{pi} = b_i \oplus h(X_{pi}\|ID_{pi}\|Z)$, $H_{pi} = h(ID_{pi}\|X_{pi}\|Z\|b_i\|t_1)$. At the end, patient $\mathcal{P}_i$ sends login request $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ to server $\mathcal{S}$.

8.1.3 Authentication Phase

On getting the login request from particular patient $\mathcal{P}_i$, the server $\mathcal{S}$ follows the steps that are as under:

Step AA 1: Server $\mathcal{S}$ obtain timestamp t_1, checks the transmission delay time interval and then calculates $ID_{pi} = DID_{pi} \oplus h(Z)$, $X_{pi} = h(ID_{pi} \| x_s)$, $b_i = G_{pi} \oplus h(ID_{pi} \| X_{pi} \| Z)$, $H_{pi}^{'} = h(ID_{pi} \| Z \| X_{pi} \| t_1 \| b_i)$ and checks $H_{pi}^{'} \stackrel{?}{=} H_{pi}$, if it does not match, the session is aborted. Otherwise, $\mathcal{P}_i$ is considered as an authorized patient $\mathcal{P}_i$. The server $\mathcal{S}$ then produce f_s as random number and calculates $M_s = f_s \oplus h(ID_{pi} \| X_{pi} \| Z)$, $H_s = h(f_s \| X_{pi} \| Z \| ID_{pi} \| t_3)$. After that $\mathcal{S}$ responds to $\mathcal{P}_i$ by sending $\{M_s, H_s, t_3\}$.

Step AA 2: Patient $\mathcal{P}_i$ obtain timestamp t_3 and checks the time interval, if the condition does not hold, the login request is denied. Otherwise, $\mathcal{P}_i$ calculates $f_s = M_s \oplus h(ID_{pi} \| X_{pi} \| Z)$, $H_s^{'} = h(f_s \| X_{pi} \| Z \| ID_{pi} \| t_3)$ and checks $H_s^{'} \stackrel{?}{=} H_s$ if it is not true the session is aborted. Otherwise, $\mathcal{P}_i$ calculates the session key $SK = h(X_{pi} \| Z \| ID_{pi} \| f_s \| b_i)$ and $H_{pi2} = h(SK)$. Then $\mathcal{P}_i$ sends $\{H_{pi2}\}$ towards server $\mathcal{S}$.

Step AA 3: Server $\mathcal{S}$ determines session key as given below, when it gets $\{H_{pi2}\}$ from $\mathcal{P}_i$. Then computes $H_{pi2}^{'} = h(SK)$ and checks $H_{pi2}^{'} \stackrel{?}{=} H_{pi2}$ if it is not true, the session terminates. Otherwise, $\mathcal{S}$ and $\mathcal{SK}$ are valid.

$$SK = h(X_{pi} \| Z \| ID_{pi} \| f_s \| b_i) \tag{8.1}$$

8.1.4 Password and Biometrics Change Phase

In this phase, the server $\mathcal{S}$ does not intervene. The patient $\mathcal{P}_i$ can change his/her biometrics and password by following steps:

Step PB 1: The patient enters his/her smart card SC_{pi} into card reader and provides his/her identity ID_{pi} and password PW_{pi}, then scans his/her biometric B_{pi}^{*} at sensor. In first step the patient $\mathcal{P}_i$ performs login similar to one discussed in login phase as step 1.

Step PB 2: $\mathcal{P}_i$ chooses new password and imprints his/her new biometric $B_{i_{new}}$. The smart card computes $E_{i_{new}} = N_{pi} \oplus h(ID_{pi} \| PW_{pi_{new}})$, $g_{i_{new}} = B_{i_{new}} \oplus h(PW_{pi_{new}} \| ID_{pi})$, $\overline{PW}_{pi_{new}} = h(ID_{pi} \| PW_{pi_{new}} \| N_{pi} \| B_{i_{new}})$, $Y_{pi_{new}} = Y_{pi} \oplus \overline{PW}_{pi_{new}} \oplus \overline{PW_{pi}}$, $C_{pi_{new}} = C_{pi} \oplus \overline{PW}_{pi_{new}} \oplus \overline{PW}_{pi}$ and $M_{pi_{new}} = h(X_{pi} \| \overline{PW}_{pi_{new}} \| ID_{pi})$. So, at the end smart card updates the new information $Y_{pi_{new}}$, $C_{pi_{new}}$, $E_{pi_{new}}$, $g_{pi_{new}}$ and $M_{pi_{new}}$.

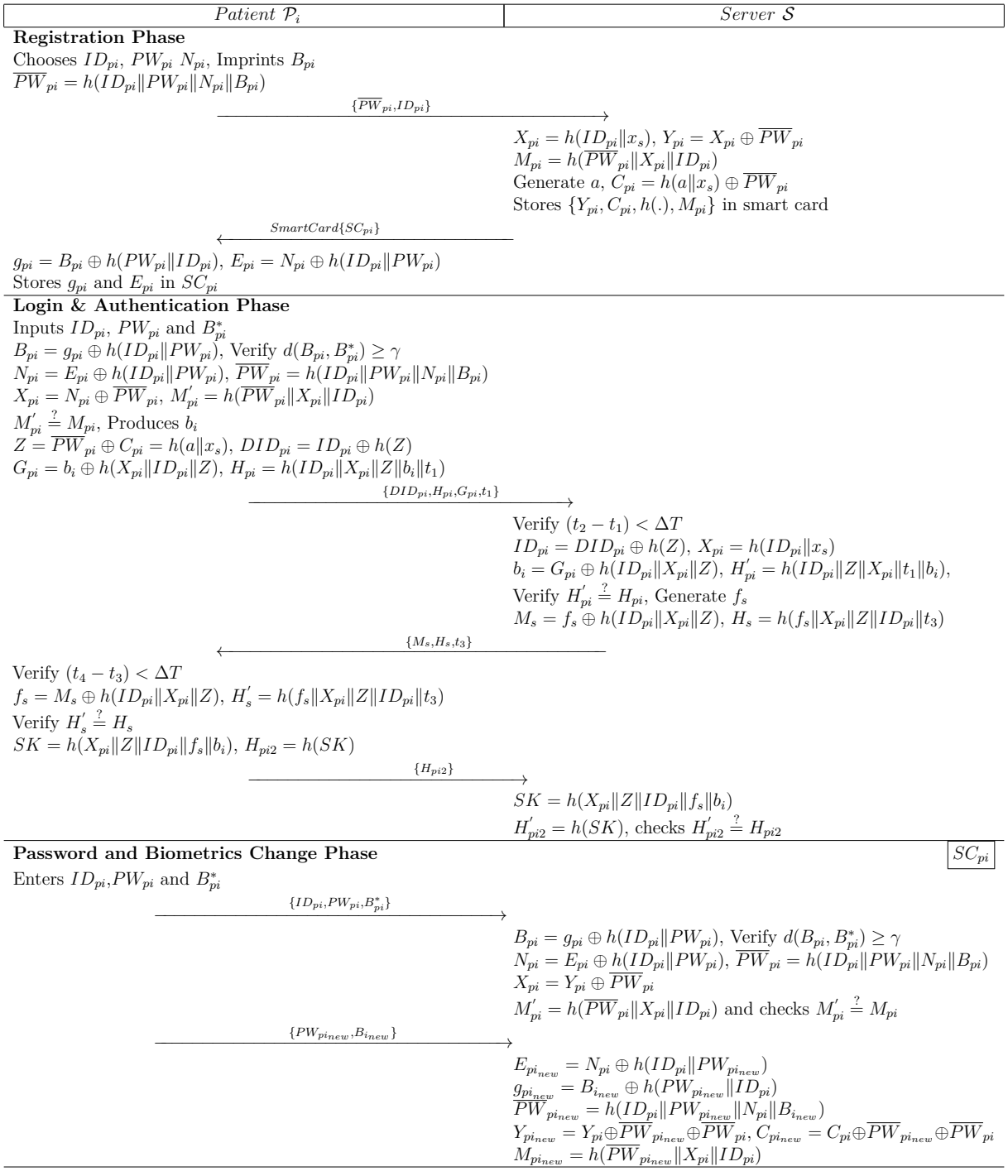


Figure 8.1: Mir and Nikooghadam's Scheme

8.2 Cryptanalysis of Mir and Nikooghadam's Scheme

In this section, it is shown that Mir and Nikooghadam's Scheme is susceptible to user anonymity violation and smart card stolen attacks. Before going into details, we made the three necessary assumptions first.

1. Public communication channel is fully accessible to an adversary $\mathcal{A}$. $\mathcal{A}$ can perform various operation over this channel such as he can inject, delete, edit and intercept any message.
2. $\mathcal{A}$ can get smart card or guess the password of a particular patient $\mathcal{P}_i$ but $\mathcal{A}$ cannot obtain them at the same time.
3. Smart card information can be easily extracted, once it is stolen [28, 29].

8.2.1 User Anonymity Violation Attack

Open architecture of ubiquitous computing enable the adversary to analyze the communication sessions and steal private information. Even adversary can keep track of particular user's location and its tours or movement history. Particular authentication scheme can only ensure anonymity if it fulfills the two main requirements. First requirement is that adversary cannot guess the identity of a particular user, and as per second requirement adversary even fails to guess that the same user has initiated two different sessions. Although, Mir and Nikooghadam claimed that their scheme fulfills the two said requirements by utilizing dynamic ID. In this subsection, we prove that their scheme is still vulnerable to user anonymity violation attack. Anonymity of legal user $\mathcal{P}_i$ can be breached by another legal user $\mathcal{P}_j$ by following the given steps.:

Step UA 1: $\mathcal{P}_j$ retrieves the values $\{N_{pj}, h(.), f_{pj}, O_{pj}, L_{pj}\}$ stored on smart card SC_{pi}, and then computes the following:

$$B_{pj} = f_{pj} \oplus h(ID_{pj} \| PW_{pj}) \tag{8.2}$$

$$D_{pj} = G_{pj} \oplus h(ID_{pj} \| PW_{pj}) \tag{8.3}$$

$$\overline{PW}_{pj} = h(ID_{pj} \| PW_{pj} \| D_{pj} \| B_{pj}) \tag{8.4}$$

$$M_{pj} = N_{pj} \oplus \overline{PW}_{pj} \tag{8.5}$$

$$L_j^{'} = h(\overline{PW}_{pj} \| M_{pj} \| ID_{pj}) \tag{8.6}$$

Step UA 2: $\mathcal{P}_j$ then calculates

$$Z = \overline{PW}_{pj} \oplus O_{pj} = h(a \| x_s) \tag{8.7}$$

Step UA 3: Then $\mathcal{P}_j$ waits until $\mathcal{P}_i$ sends request for login and authentication.

Step UA 4: When $\mathcal{P}_i$ sends request for the login and authentication by sending information $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ to server $\mathcal{S}$.

Step UA 5: Meanwhile, $\mathcal{P}_j$ intercepts the in coming message and computes:

$$ID_{pi} = DID_{pi} \oplus h(Z) \tag{8.8}$$

Hence, $\mathcal{P}_j$ can find the real identity ID_{pi} of $\mathcal{P}_i$. In this way $\mathcal{P}_j$ can easily breach the P_i's anonymity.

8.2.2 Smart Card Stolen Attack

In this subsection, it is discussed that smart card stolen attack is possible on Mir and Nikooghadam's scheme. If some legal user $\mathcal{P}_j$ is able to successfully steal $\mathcal{P}_i$'s smart card he/she can easily impersonate the legal user $\mathcal{P}_i$, and this can be done by the following steps:

Step SC 1: At first step, $\mathcal{P}_j$ computes Z by using the information stored on his own smart card and computes the ID_{pi} of remote user $\mathcal{P}_i$ by intercepting request from $\mathcal{P}_i$ of login and authentication, as discussed earlier in subsection 8.2.1.

Step SC 2: $\mathcal{P}_j$ extracts the information $\{Y_{pi}, N_{pi}, C_{pi}, h(.), X_{pi}\}$ stored on $\mathcal{P}_i$'s smart card. $\mathcal{P}_j$ then computes:

$$\overline{PW}_{pi} = C_{pi} \oplus h(a \| x_s) \tag{8.9}$$

$$X_{pi} = Y_{pi} \oplus \overline{PW}_{pi} \tag{8.10}$$

$$DID_{pi} = ID_{pi} \oplus h(Z) \tag{8.11}$$

Step SC 3: Then $\mathcal{P}_j$ chooses b_i as a random number and computes:

$$G_{pi} = b_i \oplus h(X_{pi} \| ID_{pi} \| Z) \tag{8.12}$$

$$H_{pi} = h(ID_{pi} \| X_{pi} \| Z \| b_i \| t_1) \tag{8.13}$$

Step SC 4: After that $\mathcal{P}_j$ transmits $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ to the server $\mathcal{S}$.

Step SC 5: On getting $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ from $\mathcal{P}_j$, $\mathcal{S}$ initially confirms the timestamp, and then proceeds as follows:

$$ID_{pi} = DID_{pi} \oplus h(Z) \tag{8.14}$$
$$X_{pi} = h(ID_{pi} \| x_s) \tag{8.15}$$
$$b_i = G_{pi} \oplus h(ID_{pi} \| X_{pi} \| Z) \tag{8.16}$$
$$H_{pi}^{'} = h(ID_{pi} \| Z \| X_{pi} \| t_1 \| b_i) \tag{8.17}$$

Step SC 6: The server $\mathcal{S}$ verifies whether $H_{pi}^{'} \stackrel{?}{=} H_{pi}$, and session is terminated by the $\mathcal{S}$ if it is not true. Otherwise, $\mathcal{S}$ selects a random number f_s and computes:

$$M_s = f_s \oplus h(ID_{pi} \| X_{pi} \| Z) \tag{8.18}$$
$$H_s = h(f_s \| X_{pi} \| Z \| ID_{pi} \| t_3) \tag{8.19}$$

Step SC 7: Then server $\mathcal{S}$ transmits $\{M_s, H_s, t_3\}$ to $\mathcal{P}_i$.

Step SC 8: $\mathcal{P}_j$ intercepts $\{M_s, H_s, t_3\}$ and computes:

$$f_s = M_s \oplus h(ID_{pi} \| X_{pi} \| Z) \tag{8.20}$$
$$H_s = h(f_s \| X_{pi} \| Z \| ID_{pi} \| t_3) \tag{8.21}$$
$$SK = h(X_{pi} \| Z \| ID_{pi} \| f_s \| b_i) \tag{8.22}$$
$$H_{pi2} = h(SK) \tag{8.23}$$

Step SC 9: Further, $\mathcal{P}_j$ sends H_{pi2} to $\mathcal{S}$. Upon reception $\mathcal{S}$ computes:

$$SK = h(X_{pi} \| Z \| ID_{pi} \| f_s \| b_i) \tag{8.24}$$
$$H_{pi}^{'} = h(SK) \tag{8.25}$$

Step SC 10: Finally, $\mathcal{S}$ checks $H_{pi}^{'} = H_{pi}$, because it holds. $\mathcal{S}$ considers $\mathcal{P}_j$ as legitimate $\mathcal{P}_i$ and keeps SK as shared session key with $\mathcal{P}_i$. Hence, $\mathcal{P}_j$ successfully impersonated on behalf of $\mathcal{P}_i$ to deceive $\mathcal{S}$. The shared session key between $\mathcal{P}_j$ adn $\mathcal{S}$ is as follows:

$$SK = h(X_{pi} \| Z \| ID_{pi} \| f_s \| b_i)$$

Hence, Mir and Nikooghadam's scheme can be easily compromised by smart card stolen attack.

8.3 Proposed Scheme

In this section the proposed scheme is discussed as follows:

8.3.1 The Registration Phase

The registration of a particular patient is a three step process. The patient $\mathcal{P}_i$ selects his unique identity ID_{pi} and password PW_{pi} besides random number N_{pi}. Patient $\mathcal{P}_i$ engrave his/her biometrics B_{pi} and then it determines $\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$. The patient then tranmits registration request $\{\overline{PW}_{pi}, ID_{pi}\}$ to server $\mathcal{S}$ via secure channel. The server $\mathcal{S}$ calculates $X_{pi} = h(ID_{pi}\|x_s)$,$Y_{pi} = X_{pi} \oplus \overline{PW}_{pi}$ and $M_{pi} = h(\overline{PW}_{pi}\|X_{pi}\|ID_{pi})$ in response to registration request. $\mathcal{S}$ generates random number r_0 and computes $PID_{pi} = E_{x_s}(ID_{pi}\|r_0) \oplus \overline{PW}_{pi}$. The server $\mathcal{S}$ also keeps the status bit to show the status of patient. At the end the server $\mathcal{S}$ inserts $\{Y_{pi}, PID_{pi}, h(.), M_{pi}\}$ into smart card SC_{pi}, which is then sent back to $\mathcal{P}_i$ through secure channel. The patient $\mathcal{P}_i$ gets the SC_{pi} and computes $g_{pi} = B_{pi} \oplus h(PW_{pi}\|ID_{pi})$ and $E_{pi} = N_{pi} \oplus h(ID_{pi}\|PW_{pi})$. The registration phase ends up when patient $\mathcal{P}_i$ inserts g_{pi} and E_{pi} into SC_{pi}. Hence, SC_{pi} contains the values of $\{Y_{pi}, h(.), g_{pi}, E_{pi}, PID_{pi}, M_{pi}\}$.

8.3.2 Login Phase

The login process completes in two phases that are as follows:

Step LP 1: $\mathcal{P}_i$ pushes his/her SC_{pi} into card reader and enters his her identity ID_{pi} and password PW_{pi} and provides his/her biometric scan B^*_{pi}. The smart card SC_{pi} calculates $B_{pi} = g_{pi} \oplus h(ID_{pi}\|PW_{pi})$, then checks $d(B_{pi}, B^*_{pi}) \geq \gamma$ if condition is true then session is aborted, otherwise computes $N_{pi} = E_{pi} \oplus h(ID_{pi}\|PW_{pi})$, $\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$, $X_{pi} = Y_{pi} \oplus \overline{PW}_{pi}$, $M'_{pi} = h(\overline{PW}_{pi}\|X_{pi}\|ID_{pi})$. Further it checks $M'_{pi} \stackrel{?}{=} M_{pi}$, if it does not hold then session terminates. Otherwise, ID_{pi} and PW_{pi} are detected as valid information. Then SC_{pi} computes $DID_{pi} = PID_{pi} \oplus \overline{PW}_{pi}$

Step LP 2: SC_{pi} produces b_i as random number and calculates $G_{pi} = b_i \oplus h(X_{pi}\|ID_{pi})$, $H_{pi} = h(ID_{pi}\|X_{pi}\|b_i\|t_1)$. Finally, patient $\mathcal{P}_i$ sends login request $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ to server $\mathcal{S}$.

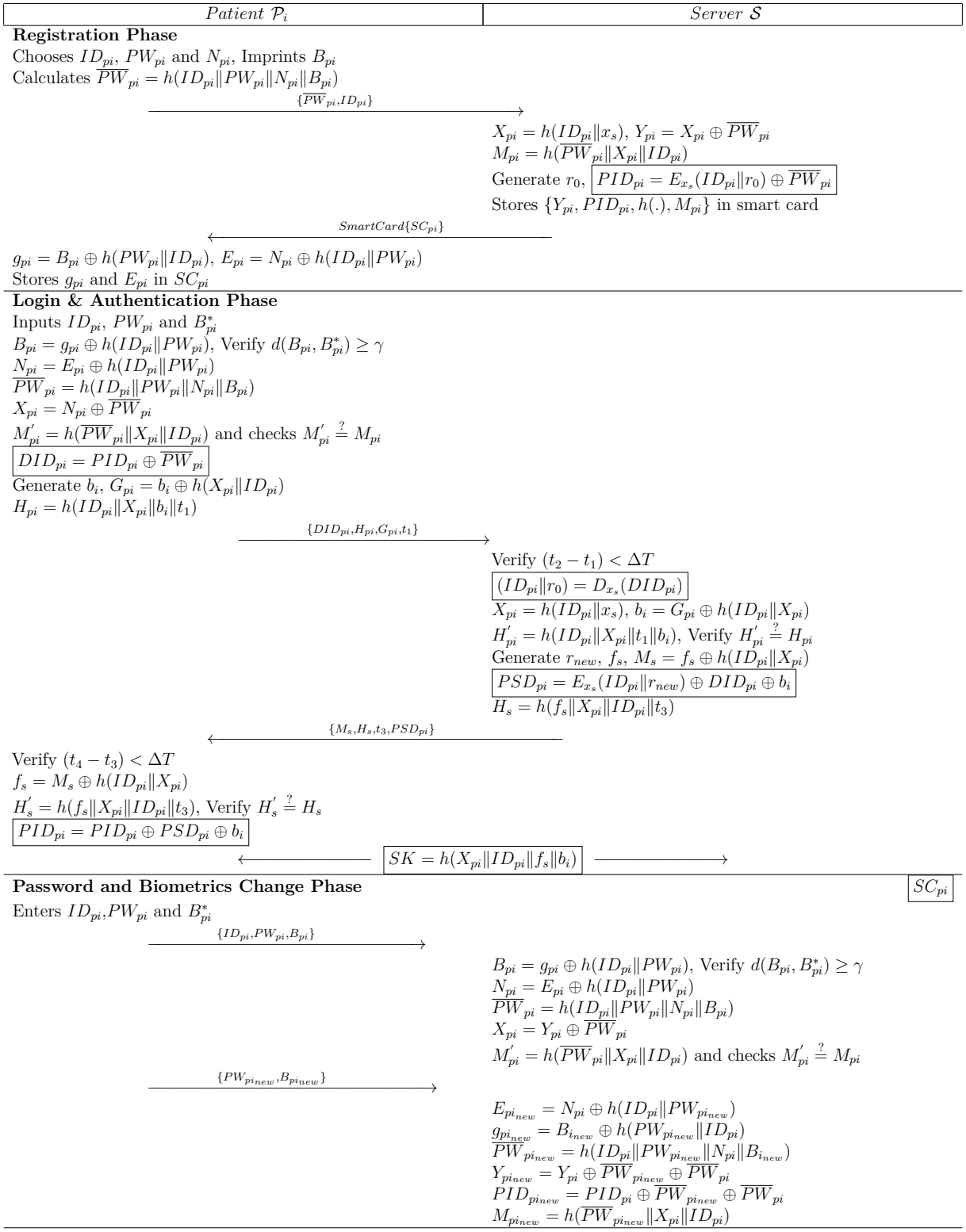


Figure 8.2: Our proposed scheme

8.3.3 Authentication Phase

On getting the login request from particular patient $\mathcal{P}_i$ the server $\mathcal{S}$ follows the steps that are as under:

Step AA 1: Server $\mathcal{S}$ obtain timestamp t_1, checks the transmission delay time interval first, and then calculates $(ID_{pi}\|r_0) = D_{x_s}(DID_{pi})$, $X_{pi} = h(ID_{pi}\|x_s)$,$b_i = G_{pi} \oplus h(ID_{pi}\|X_{pi})$, $H'_{pi} = h(ID_{pi}\|X_{pi}\|t_1\|b_i)$. Then $\mathcal{S}$ checks $H'_{pi} \stackrel{?}{=} H_{pi}$, if it does not match, the session is aborted. Otherwise, $\mathcal{P}_i$ is considered as an authorized patient. The server $\mathcal{S}$ then produces f_s, r_{new} as random numbers and calculates $M_s = f_s \oplus h(ID_{pi}\|X_{pi})$, $PSD_{pi} = E_{x_s}(ID_{pi}\|r_{new})$
$\oplus\, DID_{pi} \oplus b_i$, $H_s = h(f_s\|X_{pi}\|ID_{pi}\|t_3)$. After that server $\mathcal{S}$ responds to $\mathcal{P}_i$ by sending $\{M_s, H_s, t_3, PSD_{pi}\}$.

Step AA 2: Patient $\mathcal{P}_i$ obtains timestamp t_3 and checks the time interval, if the condition does not hold the request is denied. Otherwise, $\mathcal{P}_i$ calculates $f_s = M_s \oplus h(ID_{pi}\|X_{pi})$, $H'_s = h(f_s\|X_{pi}\|ID_{pi}\|t_3)$, checks $H'_s \stackrel{?}{=} H_s$ if it is not true then the session is aborted. Otherwise, $\mathcal{P}_i$ considers $\mathcal{S}$ as authenticated. $\mathcal{P}_i$ further calculates $PID_{pi} = PID_{pi} \oplus PSD_{pi} \oplus b_i$ and session key SK. Shared session key is given below:

$$SK = h(X_{pi}\|ID_{pi}\|f_s\|b_i) \tag{8.26}$$

8.3.4 Password and Biometrics Change Phase

In this phase the server $\mathcal{S}$ does not intervene. The patient $\mathcal{P}_i$ can change his/her biometrics and password by following steps

Step PB 1: Patient enters his/her smart card SC_{pi} into card reader and provides his/her identity ID_{pi} and password PW_{pi} and scans his/her biometrics B^*_{pi} at sensor. In first step the patient $\mathcal{P}_i$ performs login similar to one discussed in login phase as step 1.

Step PB 2: $\mathcal{P}_i$ chooses new password and imprints his/her new biometric $B_{i_{new}}$, the smart card then calculates $E_{i_{new}} = N_{pi} \oplus h(ID_{pi}\|PW_{pi_{new}})$, $g_{i_{new}} = B_{i_{new}} \oplus h(PW_{pi_{new}}\|ID_{pi})$, $\overline{PW}_{pi_{new}} = h(ID_{pi}\|PW_{pi_{new}}\|N_{pi}\|B_{i_{new}})$, $Y_{pi_{new}} = Y_{pi} \oplus \overline{PW}_{pi_{new}} \oplus \overline{PW}_{pi}$, and $M_{pi_{new}} = h(X_{pi}\|\overline{PW}_{pi_{new}}\|ID_{pi})$. So, at the end smart card updates the new information $Y_{pi_{new}}$, $C_{pi_{new}}$, $E_{pi_{new}}$, $g_{pi_{new}}$ and $M_{pi_{new}}$ with older values.

8.4 Security Analysis

In this section, formal and informal security analysis is performed of the proposed scheme. It is shown that the proposed scheme is invincible against potential known attacks, which are described in the subsequent section. These attacks are considered on the basis of supposition or hypothesis that an adversary $\mathcal{A}$ has complete control over the communication channel. So, $\mathcal{A}$ can easily change, eavesdrop, add or drop any message that is transmitted through public channel.

8.4.1 Formal Security

To demonstrate that proposed scheme is provably secure, we adopted the same analysis as mentioned in [8, 94]. Following oracles are defined for analysis purpose:

- ***Reveal:*** This oracle outputs an input string Str to the hash function $t = h(Str)$.
- ***Extract:*** This oracle unconditionally outputs plaintext P out of cipher text $C = E_{x_s}(P)$ without knowing the private key x_s.

Theorem 4. *The proposed scheme is provably secure against an adversary $\mathcal{A}$ for stemming $\mathcal{P}_i$'s identity ID_{pi}, password PW_{pi}, the session key SK and server $\mathcal{S}$'s private key x_s considering one way hash function as a random oracle.*

Proof. Let $\mathcal{A}$ be an adversary with capabilities to derive $\mathcal{P}_i$'s ID_{pi}, the session key SK and $\mathcal{S}$'s private key x_s. $\mathcal{A}$ simulates *Reveal* oracle to run algorithmic experiment $EXPE1^{SYMENC,HASH}_{\mathcal{A},AMFAS}$ against our proposed anonymous multi-factor authentication scheme ($AMFAS$). the experiment's ($EXPE1^{SYMENC,HASH}_{\mathcal{A},AMFAS}$) success probability is defined as:

$$Succe_1 = |Pr[EXPE1^{SYMENC,HASH}_{\mathcal{A},AMFAS} = 1] - 1|$$

The advantage carried by adversary is solicited as

$$Advt^{SYMENC,HASH}_{\mathcal{A},AMFAS}(t_e, q_{rev}, qext) = max_{\mathcal{A}}(Succe_1)$$

. We define t_e as the maximum execution time for $\mathcal{A}$ while q_{rev} and q_{ext} are defined as the maximum number of *Reveal* and *Extract* queries, respectively. According to the experiment, $\mathcal{A}$ can derive ID_{pi}, the server's private key x_s and the session key SK. If he can invert hash function and the symmetric encryption with knowing server's private key x_s. However, it is computationally infeasible to find Str out of $t = h(Str)$. Similarly, it is computationally infeasible to

obtain P out of $E_{x_s}(P)$ without knowing private key x_s. So, we have $Advt_A^{HASH}(t_e) \leq \epsilon$ and $Advt_A^{SYMENC}(t_e) \leq \epsilon$. As it is clearly seen that $Advt_{\mathcal{A},AMFAS}^{SYMENC,HASH}(t_e, q_{rev}, qext)$ depends on both $Advt_A^{HASH}(t_e)$ and $Advt_A^{SYMENC}(t_e)$. Therefore, $Advt_{\mathcal{A},AMFAS}^{SYMENC,HASH}(t_e, q_{rev}, qext) \leq \epsilon$. Hence, proposed anonymous multi-factor authentication scheme is secure against an adversary $\mathcal{A}$ to expose ID_{pi}, x_s and the session key SK. □

Algorithm 2 $EXPE_{\mathcal{A},AMFAS}^{SYMENC,HASH}$

1: Eavesdrop the authentication message $(DID_{pi}, H_{pi}, G_{pi}, t_1)$, Where $DID_{pi} = E_{x_s}(ID_{pi}\|r_0)$, $H_{pi} = h(ID_{pi}\|X_{pi}\|b_i\|t_1)$ and $G_{pi} = b_i \oplus h(X_{pi}\|ID_{pi})$
2: Call Extract oracle on DID_{pi} to get $(ID'_{pi}\|r'_0) \leftarrow Extract(DID_{pi})$
3: Call Reveal oracle on H_{pi} and get $(ID''_{pi}\|X'_{pi}\|b'_i\|t'_1) \leftarrow Reveal(H_{pi})$
4: **if** $(ID'_{pi} = ID''_{pi}$ and $t_1 = t'_1)$ **then**
5: Call Reveal on $b'_i \oplus G_{pi}$ and get $(X''_{pi}\|ID''_{pi}) \leftarrow Reveal(b'_i \oplus G'_{pi})$
6: **if** $(ID'_{pi} = ID'''_{pi}$ and $X'_{pi} = X''_{pi})$ **then**
7: Call Reveal on X'_{pi} and get $(ID''''_{pi}\|x'_s) \leftarrow Reveal(X'_{pi})$
8: **if** $(ID'_{pi} = ID''''_{pi})$ **then**
9: Accept x'_s as $\mathcal{S}$'s private key
10: Eavesdrop the response message $(M_s, H_s, t_3, PSD_{pi})$, Where $M_s = f_s \oplus h(ID_{pi}\|X_{pi})$, $H_s = h(f_s\|X_{pi}\|ID_{pi}\|t_3)$ and $PSD_{pi} = E_{x_s}(ID_{pi}\|r_{new}) \oplus DID_{pi} \oplus b_i$
11: Call Reveal on H_s to get $(f_s^*\|X_{pi}^*\|ID_{pi}^*\|t_3^*) \leftarrow Reveal(H_s)$
12: **if** $(t_3 = t_3^*$ and $ID'_{pi} = ID_{pi}^*$ and $X'_{pi} = X_{pi}^*)$ **then**
13: Accept f_s^* and compute session key as
14: $SK = h(X_{pi}\|ID_{pi}\|f_s\|b_i)$
15: **else**
16: **return** Fail
17: **end if**
18: **else**
19: **return** Fail
20: **end if**
21: **else**
22: **return** Fail
23: **end if**
24: **else**
25: **return** Fail
26: **end if**

8.4.2 Discussion on Functional Security

In this section correctness and security of our scheme is evaluated under the same circumstances or supposition as conversed earlier in section 8.2. Investigation indicate that our scheme is robust and efficient enough to prevent all recognized potential attacks. Use of $h(a\|x_s)$ in Mir and Nikooghadam's scheme creates the main problem because it can easily be computed by any legitimate user. Moreover, ID_{pi} and $\overline{PW}_{pi}$ can be computed by $h(a\|x_s)$ consequently results in smart card stolen and user anonymity attacks. Hence, we replaced $h(a\|x_s)$ by $E_{x_s}(ID_i\|r_0)$ in order to keep user specific calculations. Further, PID_{pi} of user is calculated as the pseudo identity by the server, not only at registration but also during each authentication session. We have illustrated the security comparison of proposed scheme with related existing scheme [8, 149, 155, 156] in table 8.2. It can be clearly seen that proposed scheme is robust

against all known attacks, whereas all other cited schemes are vulnerable to different attacks. The security of proposed scheme is explained in following subsection:

8.4.2.1 Patient Anonymity and Untraceability

Patient anonymity and untraceability is considerably important factor when designing an authentication scheme. If patient anonymity is braked, the adversary can retrieve patient's personal sensitive information like: his medical record, moving tracks, social circle and his current location etc. During registration $\mathcal{S}$ computes a dynamic identity $E_{x_s}(ID_{pi}\|r_0)$ containing his identity and a random number, encrypted by his own private key (x_s). Further, this dynamic identity is not directly stored in smart card but concealed by $\overline{PW}_{pi}$. Therefore, even if an adversary acquires the $\mathcal{P}_i$'s smart card, he will still be unable to obtain $\mathcal{P}_i$'s dynamic identity. Furthermore, in each login session server computes $\mathcal{P}_i$'s new identity $E_{x_s}(ID_{pi}\|r_n)$. It can be easily seen that $\mathcal{S}$ does not send the dynamic identity instead it first conceal it using previous identity DID_{pi} and the random nonce (b_i) and then sends $PID_{pi} = PID_{pi} \oplus PSD_{pi} \oplus b_i$ to $\mathcal{P}_i$. So, the real identity is not exposed to any adversary, further the dynamic identity is changed in each login session. Hence, proposed schemes provides patient anonymity as well as untraceability.

8.4.2.2 Privileged Insider Attack

$\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$ and ID_{pi} are sent to server $\mathcal{S}$ during registration phase, where ID_{pi}, PW_{pi}, N_{pi} and B_{pi} are concatenated and secured by one way hash function. An insider cannot calculate these hash-secured values in polynomial time. These hash concealed values are also not publicized to server $\mathcal{S}$. Therefore, we can conclude that proposed scheme successfully prevent privilege insider attack.

8.4.2.3 Replay Attack

In proposed scheme, if an adversary replays a past login message $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$. The server $\mathcal{S}$ upon reception of the message will check the freshness of timestamp t_1. As the timestamp is out dated $\mathcal{S}$ recognizes the message is a replay and simply ignore the message. Hence, proposed scheme withstands replay attack.

8.4.2.4 Stolen Verifier Attack

In proposed improved scheme, $\mathcal{S}$ makes use of his own private key (x_s) to handle login and authentication request. There is no verifier table stored by $\mathcal{S}$. Hence, no verifier table implies no stolen verifier attack possibility.

8.4.2.5 Denial of Services Attack

The smart card checks the validity of password (PW_{pi}), identity (ID_{pi}) and biometrics (B_{pi}). For any of these entries, If user enters incorrect value. The smart card simply discard the request. Hence the patient will never face denial of services due to a mistakenly entered value.

8.4.2.6 Password Guessing Attack

Let the adversary able to extract the information $\{Y_{pi}, PID_{pi}, M_{pi}, g_{pi}, E_{pi}\}$ stored on $\mathcal{P}_i$'s smart card. Then he needs to compute N_{pi} and biometrics B_{pi}. Further, the adversary has computed $\overline{PW}_{pi} = h(ID_{pi}\|PW_{pi}\|N_{pi}\|B_{pi})$. Even if the adversary gets hold of N_{pi} and B_{pi}, he has to guess two values, the identity ID_{pi} and PW_{pi}. Hence, to launch guessing attack, the adversary has to guess four different values secured by a one way hash function. Similarly, the limits on the number of incorrect login request makes it infeasible to launch online password guessing attack. Therefore, online/ offline guessing attack is infeasible in proposed scheme.

8.4.2.7 Impersonation Attack

To impersonate as a TMIS server $\mathcal{S}$, An adversary $\mathcal{A}$ needs the private key x_s of the server along with $X_{pi} = h(ID_{pi}\|x_s)$. Because the patient's real ID_{pi} is concealed in his pseudo identity $PID_{pi} = E_{x_s}(ID_{pi}\|r_0)$ and the computation of session key $SK = h(X_{pi}\|ID_{pi}\|-f_s\|b_i)$ requires to first compute $X_{pi} = h(ID_{pi}\|x_s)$. Furthermore, X_{pi} is also involved in construction of server's signature $H_s = h(f_s\|X_{pi}\|ID_{pi}\|t_3)$. Therefore, $\mathcal{A}$ cannot impersonate himself as $\mathcal{S}$ without knowing private key x_s of $\mathcal{S}$. Similarly, $\mathcal{A}$ can impersonate himself as $\mathcal{P}_i$ if he can generate valid login request $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ and response $\{H_{pi2}\}$ messages. All of these values require $\mathcal{P}_i$'s password and biometrics. Hence, proposed scheme withstands patient as well as TMIS server's impersonation attack.

Table 8.2: Comparison of Security Parameters

Scheme:	Proposed	[8]	[155]	[149]	[156]
Anonymity and privacy	Yes	No	No	Yes	Yes
Mutual authentication and key agreement	Yes	Yes	Yes	Yes	Yes
Resists impersonation attack	Yes	No	No	Yes	No
Resists smart card theft attack	Yes	Yes	Yes	Yes	No
Resists replay attack	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	No	Yes	Yes
Resists insider/ stolen verifier attacks	Yes	Yes	Yes	Yes	Yes
Resists password guessing attack	Yes	Yes	No	Yes	Yes
Resists denial of service attack	Yes	Yes	No	No	Yes

8.4.2.8 Perfect Forward Secrecy

In proposed scheme, the shared session key contains random nonceâĂŹs contributed by both the patient and TMIS server. Therefore, even if server's private key x_s is exposed to some adversary he will not be able to compute previously shared session keys. Hence, proposed scheme posses prefect forward secrecy.

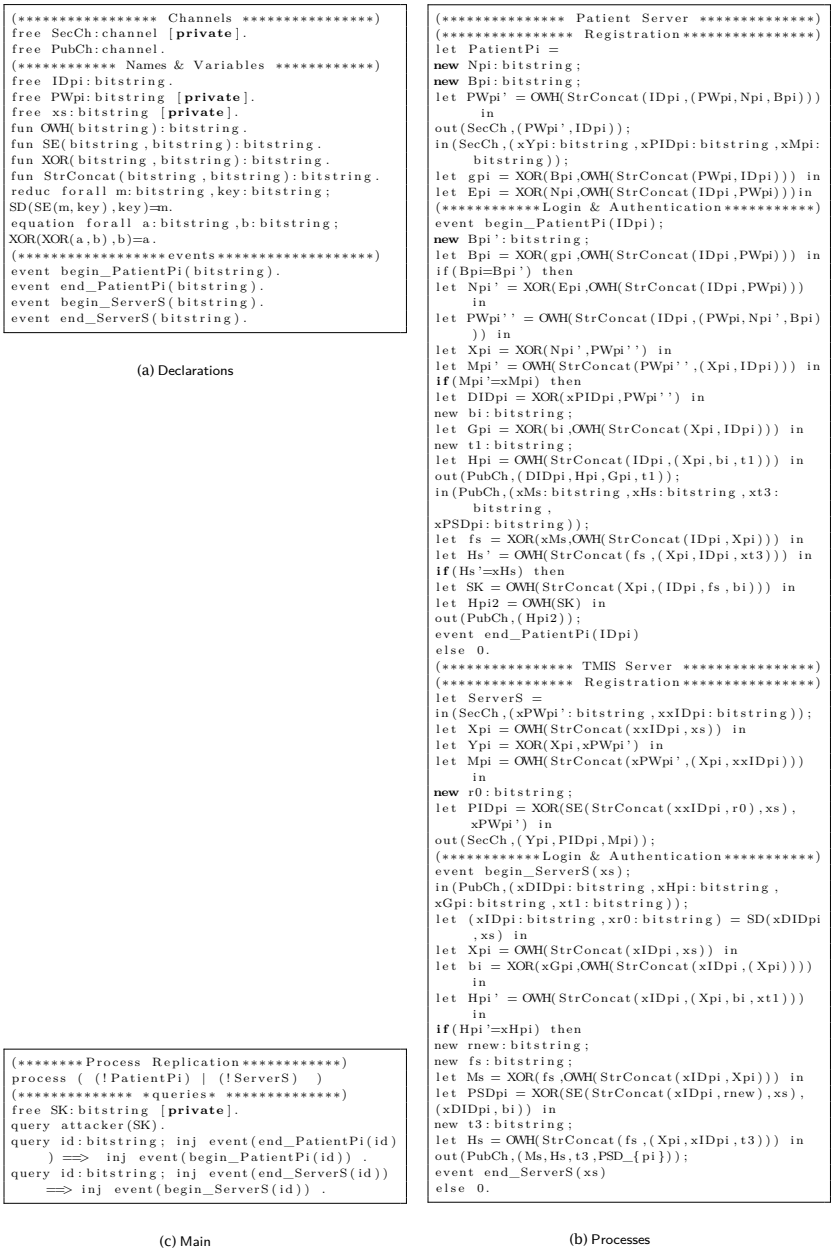
8.5 Formal Validation using ProVerif

In this section, the proposed scheme's security analysis is discussed, which is evaluated using the automated and pervasive tool ProVerif [34–36]. Security of proposed technique is proved by performing the steps given in section 8.3 and as shown in Fig 8.2. ProVerif is consisting of three parts namely: (1) Declaration; (2) Process; and (3) Main. The ProVerif code for the proposed scheme is illustrated in Fig. 8.3.

The results are as under:

1. inj-event(end˙ServerS(id)) ==> inj-event(begin˙ServerS(id)) is true.
2. inj-event(end˙PatientPi(id˙2059)) ==> inj-event(begin˙PatientPi(id˙2059)) is true.
3. not attacker(SK[]) is true.

It is verified from 1 and 2 that both patient and server processes begin and end successfully which confirms the reachability property, whereas 3 proves that session key (SK) is not exposed to adversary and secrecy is also preserved.



```
(***************** Channels ****************)
free SecCh:channel [private].
free PubCh:channel.
(************ Names & Variables ************)
free IDpi:bitstring.
free PWpi:bitstring [private].
free xs:bitstring [private].
fun OWH(bitstring):bitstring.
fun SE(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun StrConcat(bitstring,bitstring):bitstring.
reduc forall m:bitstring,key:bitstring;
SD(SE(m,key),key)=m.
equation forall a:bitstring,b:bitstring;
XOR(XOR(a,b),b)=a.
(******************events*******************)
event begin_PatientPi(bitstring).
event end_PatientPi(bitstring).
event begin_ServerS(bitstring).
event end_ServerS(bitstring).
```

(a) Declarations

```
(******** Patient Server **********)
(********** Registration**********)
let PatientPi =
new Npi:bitstring;
new Bpi:bitstring;
let PWpi' = OWH(StrConcat(IDpi,(PWpi,Npi,Bpi)))
    in
out(SecCh,(PWpi',IDpi));
in(SecCh,(xYpi:bitstring,xPIDpi:bitstring,xMpi:
    bitstring));
let gpi = XOR(Bpi,OWH(StrConcat(PWpi,IDpi))) in
let Epi = XOR(Npi,OWH(StrConcat(IDpi,PWpi)))in
(***********Login & Authentication**********)
event begin_PatientPi(IDpi);
new Bpi':bitstring;
let Bpi = XOR(gpi,OWH(StrConcat(IDpi,PWpi))) in
if(Bpi=Bpi') then
let Npi' = XOR(Epi,OWH(StrConcat(IDpi,PWpi)))
    in
let PWpi'' = OWH(StrConcat(IDpi,(PWpi,Npi',Bpi)
    )) in
let Xpi = XOR(Npi',PWpi'') in
let Mpi' = OWH(StrConcat(PWpi'',(Xpi,IDpi))) in
if(Mpi'=xMpi) then
let DIDpi = XOR(xPIDpi,PWpi'') in
new bi:bitstring;
let Gpi = XOR(bi,OWH(StrConcat(Xpi,IDpi))) in
new t1:bitstring;
let Hpi = OWH(StrConcat(IDpi,(Xpi,bi,t1))) in
out(PubCh,(DIDpi,Hpi,Gpi,t1));
in(PubCh,(xMs:bitstring,xHs:bitstring,xt3:
    bitstring,
xPSDpi:bitstring));
let fs = XOR(xMs,OWH(StrConcat(IDpi,Xpi))) in
let Hs' = OWH(StrConcat(fs,(Xpi,IDpi,xt3))) in
if(Hs'=xHs) then
let SK = OWH(StrConcat(Xpi,(IDpi,fs,bi))) in
let Hpi2 = OWH(SK) in
out(PubCh,(Hpi2));
event end_PatientPi(IDpi)
else 0.
(*************** TMIS Server ***************)
(*************** Registration***************)
let ServerS =
in(SecCh,(xPWpi':bitstring,xxIDpi:bitstring));
let Xpi = OWH(StrConcat(xxIDpi,xs)) in
let Ypi = XOR(Xpi,xPWpi') in
let Mpi = OWH(StrConcat(xPWpi',(Xpi,xxIDpi)))
    in
new r0:bitstring;
let PIDpi = XOR(SE(StrConcat(xxIDpi,r0),xs),
    xPWpi') in
out(SecCh,(Ypi,PIDpi,Mpi));
(***********Login & Authentication**********)
event begin_ServerS(xs);
in(PubCh,(xDIDpi:bitstring,xHpi:bitstring,
xGpi:bitstring,xt1:bitstring));
let (xIDpi:bitstring,xr0:bitstring) = SD(xDIDpi
    ,xs) in
let Xpi = OWH(StrConcat(xIDpi,xs)) in
let bi = XOR(xGpi,OWH(StrConcat(xIDpi,(Xpi))))
    in
let Hpi' = OWH(StrConcat(xIDpi,(Xpi,bi,xt1)))
    in
if(Hpi'=xHpi) then
new rnew:bitstring;
new fs:bitstring;
let Ms = XOR(fs,OWH(StrConcat(xIDpi,Xpi))) in
let PSDpi = XOR(SE(StrConcat(xIDpi,rnew),xs),
(xDIDpi,bi)) in
new t3:bitstring;
let Hs = OWH(StrConcat(fs,(Xpi,xIDpi,t3))) in
out(PubCh,(Ms,Hs,t3,PSD_{pi}));
event end_ServerS(xs)
else 0.
```

(b) Processes

```
(********Process Replication***********)
process ( (!PatientPi) | (!ServerS) )
(************* *queries* *************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj event(end_PatientPi(id)
    ) ==> inj event(begin_PatientPi(id)) .
query id:bitstring; inj event(end_ServerS(id))
    ==> inj event(begin_ServerS(id)) .
```

(c) Main

Figure 8.3: ProVerif Validation

8.6 Performance Evaluation

This section is about performance evaluation of proposed scheme with the former relevant schemes. Recently Mir and Nikooghadam introduced an authentication scheme in which they tried to mitigate the security weaknesses and performance concerns of Yan et al.'s scheme. In this chapter, it is proved that the Mir and Nikooghadam's scheme is vulnerable to user anonymity violation and smart card stolen attacks. The performance of the proposed scheme is compared with the schemes of Tan, Yan et al., Mishra et al. and Mir and Nikooghadam in Table 8.3. Notations are demarcated according to Kilinc and Yanik [69] and are given as under:

- T_{owh} represents total running time of hash operation, that takes $0.0023ms$.
- T_{sen} represents total running time of block cipher encryption takes $0.0046ms$.

Table 8.3: Computation Cost Comparison

Scheme	User Side	Server Side	Total
Mishra et al. [156]	$6T_{owh} \approx 0.0138ms$	$5T_{owh} + 1T_{sen} \approx 0.0161ms$	$11T_{owh} + 1T_{sen} \approx 0.0299ms$
Tan [149]	$7T_{owh} + 1T_{sen} \approx 0.0207ms$	$5T_{owh} + 1T_{sen} \approx 0.0161ms$	$12T_{owh} + 2T_{sen} \approx 0.0431ms$
Yan et al. [155]	$6T_{owh} \approx 0.0138ms$	$5T_{owh} \approx 0.0115ms$	$11T_{owh} \approx 0.0253ms$
Mir and Nikooghadam [8]	$9T_{owh} \approx 0.0207ms$	$8T_{owh} \approx 0.0184ms$	$17T_{owh} \approx 0.0391ms$
Proposed Scheme	$7T_{owh} \approx 0.0161ms$	$5T_{owh} + 2T_{sen} \approx 0.02ms$	$12T_{owh} + 2T_{sen} \approx 0.0368ms$

Comparison demonstrates that the proposed scheme performs better than Mir and Nikooghadam's and Tan's schemes but is slight expensive than the rest of the related schemes and these related schemes (including Mir and Nikooghadam's and Tan's schemes), are vulnerable to potential security attacks such as smart card stolen, offline password guessing and user anonymity violation attacks etc. On the hand, proposed scheme is more robust and is invincible against the said attacks.

Table 8.4: Communication Cost Comparison

Scheme	Messages	Transmitted Bits
Mishra et al. [156]	3	1280 bits
Tan [149]	3	842 bits
Yan et al. [155]	3	960 bits
Mir & Nikooghadam [8]	3	1024 bits
Proposed Scheme	2	1024 bits

Table 8.4 depicts the comparison of communication cost. This communication cost is derived from the number of messages exchanged and the total bandwidth utilized during login and authentication phases. Assuming one-way hash function output, the random numbers, user identity are all 160 bits each. Whereas, timestamps consume 32 bits. The proposed

scheme's login step $\{DID_{pi}, H_{pi}, G_{pi}, t_1\}$ needs (160 + 160 + 160 + 32)= 512 bits whereas the authentication step of proposed scheme $\{M_s, H_s, t_3, PSD_{pi}\}$ needs (160 + 160 + 32 + 160) = 512 bits. So the cumulative requirement of the proposed scheme comes out to be 1024 bit. So, the communication cost of the proposed scheme is slightly higher than Tan, Yan et al.'s schemes and equal to Mir and Nikooghadam's scheme but it is more secure than the rest of the schemes as proved earlier in this chapter.

8.7 Chapter Summary

In this chapter, we briefly reviewed Mir and Nikooghadam's symmetric key based authentication scheme for TMIS. We analyze that Mir and Nikooghadam's scheme cannot withstand patient anonymity violation attack as well as stolen smart card attack. Then we define an improved scheme to fix the weaknesses of Mir and Nikooghadam's scheme. The proposed scheme is more robust than Mir and Nikooghadam and related schemes which is evident from rigorous formal and informal security analysis. The proposed scheme is also more lightweight than Mir and Nikooghadam's scheme. We have also validated the security of proposed scheme by its simulation in popular security analysis tool ProVerif.

Chapter 9

A Privacy Aware Handover Authentication Scheme using ECC

The rapid development of information and communication technologies enabled mobile users to communicate with each other from anywhere. A mobile node ($\mathcal{MN}$) expects scuffle free connectivity while ensuring secure and seamless roaming over multiple access points. A general handover authentication scenario is illustrated in Fig. 9.1 involving three types of entities: mobile nodes ($\mathcal{MN}$), access points ($\mathcal{AP}$) and an authentication server ($\mathcal{AS}$). A $\mathcal{MN}$ gets register with $\mathcal{AS}$ before entering into network, then $\mathcal{MN}$ connects to an $\mathcal{AP}$ to benefit network services. When the $\mathcal{MN}$ moves out from the transmission range of an $\mathcal{AP}$ and enters in the range of another $\mathcal{AP}$, a handover authentication is needed between $\mathcal{AP}$ and $\mathcal{MN}$, to protect both from illegal access. Additionally, privacy has emerged as of wide interest, if the privacy of the user is compromised the adversary becomes able to access remote user's location, identity and roaming route, such information is very sensitive and can be sneaked and used by many companies to promote their businesses. Without guarantee of privacy users are hesitant to opt many mobile services.

9.1 Models and Goals

This subsection describes the system model, adversarial model and design goals.

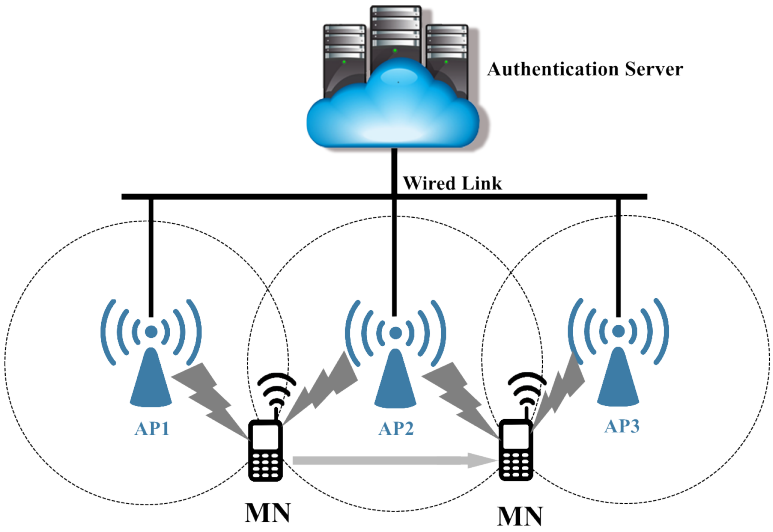


Figure 9.1: A typical Handover authentication process in wireless networks

9.1.1 System Model

A typical system model of handover authentication is illustrated in the Fig. 9.1, which involves three entities, an authentication server $\mathcal{AS}$, the access point $\mathcal{AP}_h$ and mobile node $\mathcal{MN}_j$. Initially, $\mathcal{AP}_h$ and $\mathcal{MN}_j$ both get register with $\mathcal{AS}$ to obtain the identity based long term keys, the $\mathcal{MN}_j$ can then connect with $\mathcal{AP}_h$ to get desired services. A handover authentication is performed when $\mathcal{MN}_j$ roams from the coverage range of $\mathcal{AP}_h$ to $\mathcal{AP}_i$, in this case both $\mathcal{MN}_j$ and $\mathcal{AP}_i$ authenticates each other and generates a shared session key. The session key is used to protect the confidentiality of communication between them. We assume that each $\mathcal{AP}_i$ is having a high quality tempered proof device, which restricts the adversary to extract long term secret keys of $\mathcal{AP}_i$.

9.1.2 Adversarial Model

Here, we considered different adversaries based on their capabilities to highlight privacy preservation:

1. **Non global adversary:** An adversary with limited capabilities, this type of adversary can only eavesdrop the communication between $\mathcal{MN}_j$ and $\mathcal{AP}_i$. However, non global adversary is not able to extract whole information exchanged between $\mathcal{MN}_j$ and $\mathcal{AP}_i$.
2. **Weak global adversary:** This type of adversary is more powerful as compared to non global adversary. Weak global adversary can passively eavesdrop the whole

communication between $\mathcal{MN}_j$ and $\mathcal{AP}_i$, which can be useful for adversary to trace $\mathcal{MN}_j$'s movement route.

3. **Strong global adversary:** A strong global adversary is also having the ability to compromise some of the $\mathcal{AP}_i$'s. The threat model considered in case of strong global adversary is of course stronger than the real scenarios. However, strong global adversary is not able to compromise the secret keys of any $\mathcal{AP}_i$, because in reality the secret keys are protected by temper proof devices.

9.1.3 Design Goals

The design goals of this research is to propose a privacy-aware handover authentication protocol which can achieve following objectives:

1. **Fast handover:** The handover authentication protocol should be fast enough to cope with time limitations of handover. It should have lightweight cryptographic primitives, the number of such operations should be minimum.
2. **Mutual Authentication:** The protocol must be able to provide mutual authentication and a fresh session key. The session key ensures confidentiality and integrity. The session key must contain secret parameters from both $\mathcal{MN}_j$ and $\mathcal{AP}_i$ to ensure forward secrecy.
3. **Anonymity & Privacy protection:** The protocol must ensures $\mathcal{MN}_j$'s anonymity and privacy, real identity of $\mathcal{MN}_j$ should not even revealed to $\mathcal{AP}_i$, in addition no strong global adversary is able to detect $\mathcal{MN}_j$'s movement route.

9.2 Literature Review

A lot of research has been carried out focusing faster handover authentication with privacy protection and recently, many handover authentication protocols have been proposed using different techniques [157, 158]. In 2012, He et al. [159] proposed a novel privacy-preserving authentication protocol for faster handover in wireless networks. In their protocol, $\mathcal{AS}$ assigns a set of pseudo identities to each $\mathcal{MN}$ to preserve anonymity and untraceability. They also claimed that their protocol is computation and communication efficient than earlier protocols while resisting all known attacks. However, He et al. [160] demonstrated that the protocol designed in [159] is lacking the claimed security, where an adversary can easily figure out

the session key by intercepting the transmitted message. Furthermore, they proposed an improved protocol in [160] and argued that the protocol is able remove the security weakness of the protocol proposed in [159]. However, Yeo et al. [161] pointed out that the protocol proposed in [160] is still vulnerable to the key compromise attack. In 2013, Tsai et al. [162] proposed another improvement of the protocol explained in [160] and claimed that the proposed protocol can achieve better performance than the existing protocols [163]. In 2014, a provably secure handover authentication protocol for wireless mobile networks is proposed by Islam and Khan [163]. It is to be noted that their protocol is free from time consuming bilinear pairing and map-to-point hash function. In addition, their handover authentication protocol achieved the provable security in the random oracle model. In 2015, He et al. [17] showed that the handover authentication protocol proposed in [160] is vulnerable to the private key compromised attack. Then they proposed an enhanced handover authentication protocol in wireless networks using elliptic curve cryptography and bilinear pairing. However, their protocol is not provably secured in the random oracle model.

In 2015, Li et al. [9] identified that the existing handover authentication protocols [157, 159, 160, 162, 164] are either inefficient or insecure, and such protocols are not suitable for fast moving mobile nodes. Then, Li et al. proposed a new privacy-aware handover authentication protocol for wireless networks and claimed that it can provide mutual authentication between mobile node and access point, while achieving low computation and communication costs. However in this chapter, we show that Li et al.'s protocol [9] is vulnerable to access point impersonation attack. Furthermore, we proposed an improved protocol, which can resist all known attacks. The improved protocol has the following merits:

- Our protocol is provably secure in the random oracle model against the hardness assumptions of the elliptic curve discrete logarithm problem and elliptic curve computational Diffie-Hellman problem.
- Our protocol is secured based on the analysis of automated tool ProVerif.
- Our protocol achieves low computation costs than other existing and related protocols.
- Our protocol provides the mutual authentication and fast handover authentication between mobile node and access point along with anonymity and untraceability of the mobile node.

Table 9.1: Notation Guide

Notations	Description	Notations	Description
p, q	Two large prime number of k-bit, $p = 2q + 1$	E/F_p	Elliptic Curve
G	Elliptic curve group, $G = \{E/F_p\} \cup \{O\}$	P	Base point over E/F_p
$\mathcal{MN}_j$	j^{th} Mobile node	$\mathcal{MN}_j$	j^{th} Mobile node
$\mathcal{AS}$	Authentication server	$H_1(\cdot)$, $H_2(\cdot)$	Two one way hash functions
$\mathcal{F}$	Key generation function	s	Master secret key of $\mathcal{AS}$
$PK = sP$	Master public key of $\mathcal{AS}$	ID_i	Identity of $\mathcal{AP}_i$
ID_j	Real identity of $\mathcal{MN}_j$	PID_j	One time pseudo identity of $\mathcal{MN}_j$
s_i, R_i	The key pair of $\mathcal{AP}_i$	$PK_i = s_iP$	Public key of $\mathcal{AP}_i$
s_j, R_j	The key pair of $\mathcal{MN}_j$	$PK_j = s_jP$	Public key of $\mathcal{MN}_j$
t_j	Time stamp	$\mathcal{A}$	The Adversary

9.2.1 Roadmap of the Chapter

Rest of the chapter is organized as follows: Section 9.3 reviews Li et al.'s handover authentication protocol for wireless networks. Section 9.4 analyzed the access point impersonation attack on Li et al.'s protocol. Section 9.5 describes our improved handover authentication protocol for wireless networks. Section 9.6 performs the security analysis of the proposed protocol in the random oracle model and ProVerif tool. Section 9.7 incorporates the performance analysis and a comparative analysis of our protocol and other related protocols. Finally, chapter's summary is solicited in Section 9.8.

9.3 Review of Li et al.'s Protocol

In this section, we review Li et al.'s privacy-aware handover authentication protocol [9]. Li et al.'s protocol is consisting of following three phases:

9.3.1 System Setup Phase

In this phase, given a security parameter k, $\mathcal{AS}$ initializes all system parameters in the following ways:

- Selects a large prime number q and field size p where $p = 2q + 1$.
- Generates a elliptic curve E/F_p, then a base point P and cyclic group G under addition of order q are specified over E/F_p.
- Chooses a master secret key $s \in Z_q^*$ and computes master public key $PK = sP$.

- Selects two one way hash functions $H_1 : \{0,1\}^* \times G \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \times G \rightarrow Z_q^*$, and a key generation function $\mathcal{F} : G \rightarrow \{0,1\}^k$.
- Publishes the system parameters $\{p,\ q,\ E/F_p,\ P,\ G,\ PK = sP,\ H_1,\ H_2,\ \mathcal{F}\}$ and keeps s secret.

For each $\mathcal{AP}_i$, $\mathcal{AS}$ computes the private and public keys as follows:

- Assigns a unique identifier ID_i to $\mathcal{AP}_i$.
- Selects a random number $r_i \in Z_q^*$ and computes $R_i = r_iP$.
- Computes $h_i = H_1(ID_i, R_i)$ and $s_i = r_i + sh_i$ and set an expiry time $T_{i_{exp}}$ for ID_i.
- It is assumed that $\mathcal{AP}_i$ and $\mathcal{AS}$ are having a pre-shared secret key. $\mathcal{AS}$ using the pre-shared key encrypts and sends the tuple (s_i, R_i) to $\mathcal{AP}_i$.

Upon receiving the encrypted (s_i, R_i), $\mathcal{AP}_i$ decrypts (s_i, R_i) and keep the tuple (s_i, R_i) as his private key, $\mathcal{AP}_i$ further computes his public key $PK_i = s_iP = R_i + H_1(ID_i, R_i)PK$.

9.3.2 Handover Preparation Phase

It has been assumed that initially a complete authentication has been performed between $\mathcal{AS}$ and the mobile node $\mathcal{MN}_j$, which ended up after sharing a secret key among $\mathcal{AS}$ and $\mathcal{MN}_j$. $\mathcal{AS}$ generates a set of dynamic identifiers $PID_1,\ PID_2,\ \cdots,\ PID_n$ and a one time set of public and private key pairs for $\mathcal{MN}_j$. $\mathcal{AS}$ performs the following steps for $\mathcal{MN}_j$:

- Selects a random number $r_j \in Z_q^*$ and computes $R_j = r_jP$.
- Computes $h_j = H_1(PID_j, R_j)$ and $s_j = r_j + sh_j$.

Finally, $\mathcal{AS}$ using pre-shared key encrypts and sends (s_j, R_j) to $\mathcal{MN}_j$. Upon receiving encrypted (s_j, R_j), $\mathcal{MN}_j$ first decrypts the message (s_j, R_j) and obtains his private key pair (s_j, R_j), then computes his public key $PK_j = s_jP = R_j + H_1(PID_j, R_j)PK$.

9.3.3 Handover Authentication Phase

The handover authentication is performed when $\mathcal{MN}_j$ moves out from the coverage of one access point to a new access point. Each access point periodically broadcasts a beacon message containing its identity ID_i and R_i along with other network related information.

$\mathcal{MN}_j$ after receiving beacon message enters into handover authentication with $\mathcal{AP}_i$ having identity ID_i. Following steps are performed for handover authentication:

Step HA1 $\mathcal{MN}_j \rightarrow \mathcal{AP}_i : \{PID_j, R_j, A, t_j, X, Y\}$

$\mathcal{MN}_j$ selects random $a \in Z_q^*$ and computes $A = aP$. Further, $\mathcal{MN}_j$ selects another random number $x \in Z_q^*$ then computes $X = xP$. Let $m = \{PID_j, R_j, A, t_j\}$, where t_j is the freshly generated time stamp. $\mathcal{MN}_j$ using m and his private key s_j generates signature $\delta = \{X, Y, R_j\}$ where $Y = x + s_j H_2(PID_j, X, m)$. Finally, $\mathcal{MN}_j$ sends the login message $\{PID_j, R_j, A, t_j, X, Y\}$ to $\mathcal{AP}_i$.

Step HA2 $\mathcal{AP}_i \rightarrow \mathcal{MN}_j : \{ID_i, B, MAC_{ij}\}$

Upon receiving $\{PID_j, R_j, A, t_j, X, Y\}$, $\mathcal{AP}_i$ verifies the freshness of t_j, aborts the session, if t_j is not fresh. Otherwise, $\mathcal{AP}_i$ extracts $\mathcal{MN}_j$'s signature $\delta = \{X, Y, R_j\}$ and verifies it by following equation: $YP \stackrel{?}{=} X + (H_2(PID_j, X, m)(H_1(PID_j, R_j)PK) + R_j)$, if unsuccessful the session is terminated by $\mathcal{AP}_i$. Otherwise, $\mathcal{AP}_i$ selects a random number $b \in Z_q^*$ and computes $B = bP$. $\mathcal{AP}_i$ further computes $K_{AM} = (s_i + b)(H_1(PID_j, R_j)PK + R_j + A)$ and the session key $k_{am} = \mathcal{F}(K_{AM}, PID_j, ID_i)$. Finally, $\mathcal{AP}_i$ computes the message authentication code by applying a secure message authentication function λ as follows: $MAC_{ij} = \lambda(PID_j, ID_i, B, A, K_{AM})$ and sends $\{ID_i, B, MAC_{ij}\}$ to $\mathcal{MN}_j$ in reply message.

Step HA3 After receiving reply message, $\mathcal{MN}_j$ computes $K_{MA} = (s_j + a)(H_1(ID_i, R_i)PK + R_i + B)$ and session key $k_{ma} = \mathcal{F}(K_{MA}, PID_j, ID_i)$. $\mathcal{MN}_j$ verifies the following equation: $MAC_{ij} \stackrel{?}{=} \lambda(PID_j, ID_i, B, A, K_{MA})$, if verification is successful $\mathcal{MN}_j$ treats $\mathcal{AP}_i$ as legal application provider, and a secure channel is established between $\mathcal{MN}_j$ and $\mathcal{AP}_i$.

Li et al.'s handover authentication phase is also illustrated in Fig. 9.2.

9.4 Impersonation Attack on Li et al.'s Protocol

This section proves that Li et.al.'s handover authentication protocol is vulnerable to access point impersonation attack. An adversary $\mathcal{A}$ can easily impersonate as a legal access point $\mathcal{AP}_i$ to deceive a mobile node $\mathcal{MN}_j$ under the proposed adversarial model. Initially, $\mathcal{A}$ intercepts $\mathcal{AP}_i$'s beacon message which contains ID_i and R_i, then the following steps are performed between $\mathcal{A}$ and $\mathcal{MN}_j$ for successful impersonation attack.

Step 1 When $\mathcal{MN}_j$ moves and comes in the coverage range of $\mathcal{A}$, it sends authentication

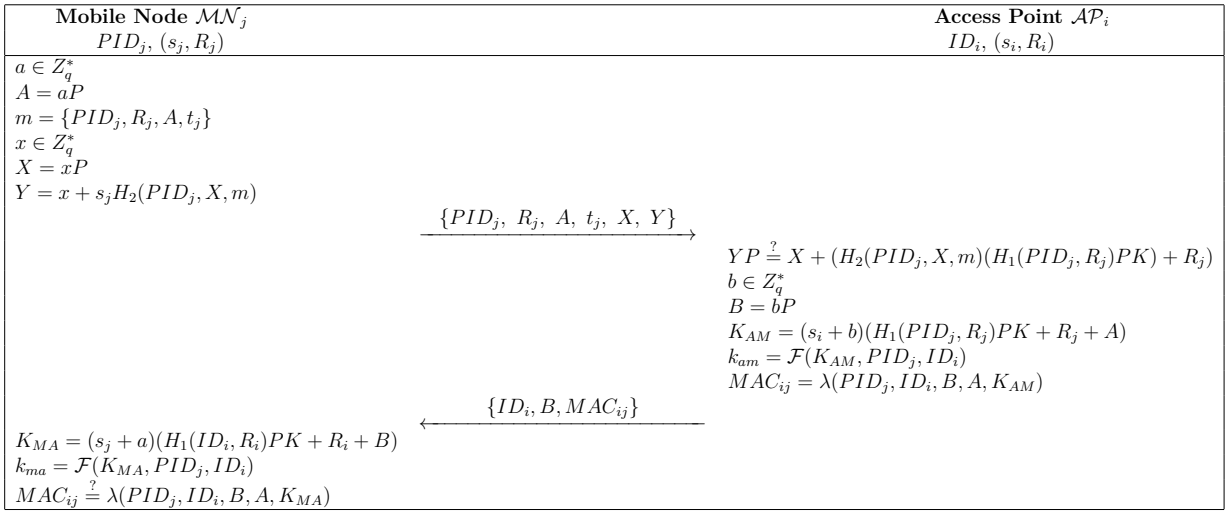


Figure 9.2: Li et al.'s handover authentication protocol

request message containing $\{PID_j,\ R_j,\ A,\ t_j,\ X,\ Y\}$.

Step 2 $\mathcal{A}$ intercepts the message and computes the following:

$$B = P - [H_1(ID_i, R_i)PK + R_i] \tag{9.1}$$

$$K_{AM} = H_1(PID_j, R_j)PK + R_j + A \tag{9.2}$$

$$k_{am} = \mathcal{F}(K_{AM}, PID_j, ID_i) \tag{9.3}$$

$$MAC_{ij} = \lambda(PID_j, ID_i, B, A, K_{AM}) \tag{9.4}$$

Step 3 $\mathcal{A}$ sends $\{ID_i,\ B,\ MAC_{ij}\}$ to $\mathcal{MN}_j$.

Step 4 $\mathcal{MN}_j$ computes the following:

$$K_{MA} = (s_j + a)(H_1(ID_i, R_i)PK + R_i + B) \tag{9.5}$$

$$= H_1(PID_j, R_j)PK + R_j + A \tag{9.6}$$

$$k_{ma} = \mathcal{F}(K_{MA}, PID_j, ID_i) \tag{9.7}$$

$$MAC_{ji} = \lambda(PID_j, ID_i, B, A, K_{MA}) \tag{9.8}$$

Step 5 $\mathcal{MN}_j$ checks whether $MAC_{ij} \stackrel{?}{=} MAC_{ji}$, if it does not hold, $\mathcal{MN}_j$ aborts the session, otherwise $\mathcal{MN}_j$ accepts $\mathcal{A}$ as a legal AP, the session key computes between

$\mathcal{MN}_j$ and $\mathcal{A}$ is $k_{am} = \mathcal{F}(K_{MA}, PID_j, ID_i)$.

The impersonation attack on Li et al.'s protocol is further explained in Fig. 9.3.

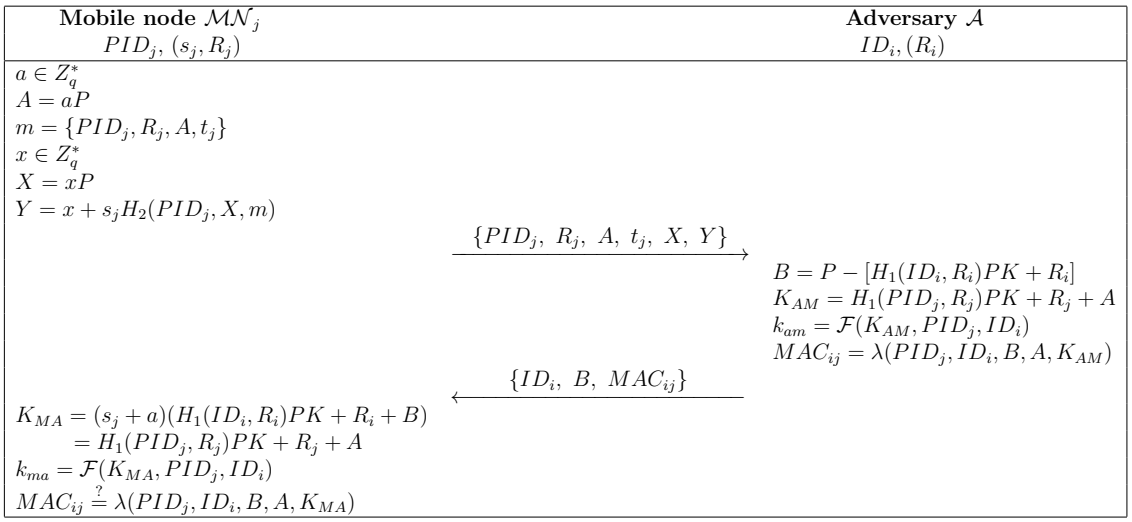


Figure 9.3: Impersonation Attack on Li et al.'s Handover Authentication Protocol

Proposition 1. At end of impersonation attack, the mobile node $\mathcal{MN}_j$ accepts adversary $\mathcal{A}$ as legal access point $\mathcal{AP}_i$.

Proof. $\mathcal{A}$ periodically broadcasts the beacon message $\{ID_i, R_i\}$, while roaming $\mathcal{MN}_j$ enters into the range of $\mathcal{A}$. $\mathcal{MN}_j$ after receiving beacon message, sends request message $\{PID_j,\ R_j,\ A,\ t_j,\ X,\ Y\}$. $\mathcal{A}$ intercepts the request message and computes B, K_{AM}, k_{am} and MAC_{ij}. Finally, $\mathcal{A}$ sends $\{ID_i,\ B,\ MAC_{ij}\}$ to $\mathcal{MN}_j$. $\mathcal{MN}_j$ recognizes the legitimacy of $\mathcal{AP}_i$ if equation (9.8) holds, which can hold if K_{AM} computes by $\mathcal{A}$ in equation (9.2) is equal to K_{MA} computed by $\mathcal{MN}_j$ in equation (9.5), as PID_j, ID_i, B, A are public so easily accessible to adversary. We show K_{AM} and K_{MA} are equal as follows:

$$\begin{aligned} K_{MA} &= (s_j + a)(H_1(ID_i, R_i)PK + R_i + B) && \text{By Eq. 9.5} \\ &= H_1(PID_j, R_j)PK + R_j + A && \text{By Eq. 9.1} \\ &= K_{AM} && \text{By Eq. 9.2} \end{aligned}$$

Hence, $\mathcal{MN}_j$ accepted $\mathcal{A}$ as the legal access point $\mathcal{AP}_i$, and the session key computed by both sides is same. □

9.5 Proposed Handover Authentication Protocol

In this section, we improve the handover authentication protocol proposed by Li et. al. The improved protocol not only robust against known attacks, but also more lightweight than Li et.al.'s protocol. The proposed protocol can be described by following three phases:

9.5.1 System Setup Phase

$\mathcal{AS}$ sets up all the public and private system parameters. Given a security parameter k, $\mathcal{AS}$ performs following steps:

- Selects p, q, where q ($> 2^{160}$) is a large prime number and p is field size, where $p = 2q+1$.
- Selects E/F_p an elliptic curve over F_p, a base point P over E/F_p and an additive cyclic group G generated by P.
- Chooses a master secret key $s \in Z_q^*$ and compute the master public key $PK = sP$.
- Chooses two one way hash functions $H_1 : \{0,1\}^* \times G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G \rightarrow Z_q^*$ and a key generation function $\mathcal{F} : G \rightarrow \{0,1\}^k$.
- Publishes the system parameters $\{p, q, E/F_p, P, G, PK = sP, H_1, H_2, \mathcal{F}\}$ and keeps s secret.

Afterwords, $\mathcal{AS}$ computes private and public keys of all access points. Initially, each $\mathcal{AP}_i$ is assigned a unique identifier ID_i and an expiry time $T_{i_{exp}}$ for ID_i. The $\mathcal{AS}$ generates identity based keys for each $\mathcal{AP}_i$. $\mathcal{AS}$ performs following steps for each $\mathcal{AP}_i$:

- Selects a number $r_i \in Z_q^*$ and computes $R_i = r_i P$.
- Computes $h_i = H_1(ID_i, R_i)$ and $s_i = r_i + sh_i$.

Finally, $\mathcal{AS}$ encrypts and sends the tuple (s_i, R_i) to $\mathcal{AP}_i$. It is assumed both $\mathcal{AP}_i$ and $\mathcal{AS}$ are having a pre-shared secret key. Upon receiving encrypted (s_i, R_i), $\mathcal{AP}_i$ decrypts (s_i, R_i) and keep the tuple (s_i, R_i) as his private key, $\mathcal{AP}_i$ further computes his public key $PK_i = s_i P = h_i PK + R_i$, where $h_i = H_1(ID_i, R_i)$.

9.5.2 Handover Preparation Phase

It has been assumed that initially a complete authentication has been performed between $\mathcal{AS}$ and the mobile node $\mathcal{MN}_j$, which ended up after sharing a secret key among $\mathcal{AS}$ and

$\mathcal{MN}_j$. In this phase, $\mathcal{AS}$ selects a set of pseudo identities $\{PID_1, PID_2, \cdots, PID_n\}$ and for each pseudo identity PID_j, $\mathcal{AS}$ generates a one time set of public and private key pairs. $\mathcal{AS}$ performs following steps for $\mathcal{MN}_j$ with an identifier PID_j.

- Selects a random number $r_j \in Z_q^*$ and computes $R_j = r_j P$.
- Computes $h_j = H_1(PID_j, R_j)$ and $s_j = r_j + sh_j$.

Finally, $\mathcal{AS}$ sends (s_j, R_j) to $\mathcal{MN}_j$ through some secure channel. Upon receiving (s_j, R_j), $\mathcal{MN}_j$ first decrypts the message (s_j, R_j) and obtains his private key pair (s_j, R_j), then computes his public key $PK_j = s_j P = h_j PK + R_j$, where $h_j = H_1(PID_j, R_j)$.

9.5.3 Handover Authentication Phase

The handover authentication phase is carried out when a mobile node moves out from the coverage of one access point to a new access point. Let $\mathcal{AP}_i$ is an access point with identity ID_i. $\mathcal{AP}_i$ periodically broadcasts a beacon message containing its identity ID_i and R_i along with other network related information. Let $\mathcal{MN}_j$ with real identity ID_i and pseudo identity PID_j enters in the coverage range of an access point $\mathcal{AP}_i$. $\mathcal{MN}_j$ after receiving beacon message enters into handover authentication with $\mathcal{AP}_i$. Following steps are performed between $\mathcal{MN}_j$ and $\mathcal{AP}_i$ for handover authentication:

Step PHA1 $\mathcal{MN}_j \rightarrow \mathcal{AP}_i : \{PID_j, R_j, A, t_j, Y\}$

$\mathcal{MN}_j$ selects a random number $a_j \in Z_q^*$ and computes $A_j = a_j P$, $\mathcal{MN}_j$ then computes signature on $m_j = \{PID_j, R_j, A_j, t_j\}$ using his private key s_j as $Y_j = a_j + s_j z_j$, where $z_j = H_2(PID_j, A_j, m_j)$. Finally, $\mathcal{MN}_j$ sends the request message $\{PID_j, R_j, A_j, t_j, Y_j\}$ to $\mathcal{AP}_i$, where t_j is the current time stamp recorded at $\mathcal{MN}_j$.

Step PHA2 $\mathcal{AP}_i \rightarrow \mathcal{MN}_j : \{ID_i, B_i, MAC_{ij}\}$

Upon receiving the request message $\{PID_j, R_j, A_j, t_j, Y_j\}$, $\mathcal{AP}_i$, first verifies the freshness of t_j, aborts the session if t_j is not fresh. Otherwise, $\mathcal{AP}_i$ checks $Y_j P \stackrel{?}{=} A_j + z_j(h_j PK + R_j)$, where $z_j = H_2(PID_j, A_j, m_j)$, $h_j = H_1(PID_j, R_j)$. If it is unsuccessful, $\mathcal{AP}_i$ terminates the session. Otherwise, $\mathcal{AP}_i$ selects a random number $b_i \in Z_q^*$ and computes $B_i = b_i P$. $\mathcal{AP}_i$ further computes $K_{ij} = (s_i + b_i l_i)(h_j PK + R_j + l_j A_j)$, where $l_i = H_2(ID_i, R_i, B_i)$, $l_j = H_2(PID_j, R_j, A_j)$ and the session key $k_{ij} = \mathcal{F}(K_{ij}, PID_j, ID_i)$. Finally, $\mathcal{AP}_i$ computes the message authentication code by applying a secure message authentication function λ as follows: $MAC_{ij} = \lambda(PID_j, ID_i, B_i, A_j, K_{ij})$. Finally, $\mathcal{AP}_i$ sends $\{ID_i, B_i, MAC_{ij}\}$ to $\mathcal{MN}_j$ in a reply message.

Step PHA3 After receiving the reply message $\{ID_i,\ B_i,\ MAC_{ij}\}$, $\mathcal{MN}_j$ computes $K_{ji} = (s_j + l_j a_j)(h_i PK + R_i + l_i B_i)$, where $l_j = H_2(PID_j,\ R_j,\ A_j)$, $l_i = H_2(ID_i,\ R_i,\ B_i)$ and session key $k_{ji} = \mathcal{F}(K_{ji},\ PID_j,\ ID_i)$. $\mathcal{MN}_j$ computes $MAC_{ji} = \lambda(PID_j,\ ID_i,\ B_i,\ A_j,\ K_{ji})$ and verifies whether the following equation: $MAC_{ji} \stackrel{?}{=} MAC_{ij}$, if verification is successful $\mathcal{MN}_j$ treat $\mathcal{AP}_i$ as legal application provider, and a secure channel is established between $\mathcal{MN}_j$ and $\mathcal{AP}_i$ with session key k_{ji}, otherwise $\mathcal{MN}_j$ aborts the session.

Proposed handover authentication phase is further elaborated in the Fig. 9.4.

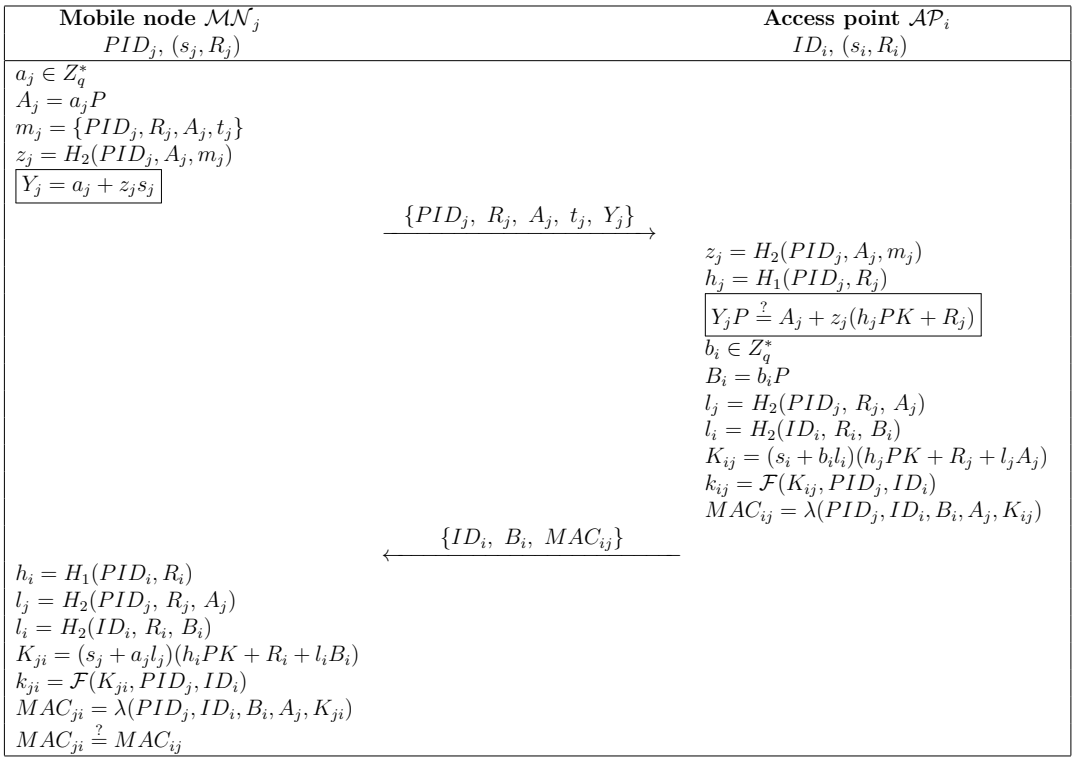


Figure 9.4: Proposed Handover Authentication Scheme protocol

9.6 Security Analysis

In this section, we perform provable security analysis in the random oracle model and formal security validation using ProVerif tool of our proposed protocol in consistent with the design goals as mentioned in subsection 9.1.3. The analysis is performed to verify that proposed protocol can avoid the strong global adversary as described in subsection 9.1.2.

9.6.1 Formal security analysis in the random oracle model

The proposed authentication process involves two participants, a mobile node $\mathcal{MN}_j$ and an access point $\mathcal{AP}_i$. There may be several instances of the participants, for $\mathcal{P} \in \{\mathcal{MN}_j, \mathcal{AP}_i\}$. We denote $\mathcal{P}^x$ as the x^{th} instance of either $\mathcal{MN}_j$ or $\mathcal{AP}_i$. We consider the same adversarial model introduced in [159, 160, 162], which is also mentioned in section 9.1.2. According to the capabilities, adversary can accomplish the following queries:

- $H_1/H_2/F/\lambda$**:** These are one way hash oracles, each of the said query maintains a respective list.
- **Extract(**ID_j**):** This query returns the identity based private key s_j of $\mathcal{P}^j$.
- **Send(**$\mathcal{P}^x, msg$**):** This query imitates the active attack, a global adversary $\mathcal{A}$ can make this query, where $\mathcal{A}$ is authorized to modify eavesdropped message and to create a fresh message, then send it to the protocol participant $\mathcal{P}^x$. The output for this query will be the reply message from $\mathcal{P}^x$. This query terminates same as the steps of proposed handover authentication protocol.
- **Execute(**$\mathcal{P}^x, \mathcal{P}^y$**):** This query outputs the messages exchanged between $\mathcal{P}^x$ and $\mathcal{P}^y$.
- **Reveal(**$\mathcal{P}^x$**):** The attacker makes this query to get the session key exchanged between $\mathcal{P}^x$ and $\mathcal{P}^y$, if accepted, it's output is the session key, otherwise, it returns a random string.
- **Corrupt(**$\mathcal{P}^x$**):** Through this query $\mathcal{A}$ can access the private key of $\mathcal{MN}_j$ or $\mathcal{AP}_i$.
- **Test(**$\mathcal{P}^x$**):** By simulating this query, $\mathcal{A}$ can obtain the session key. The results will be $\perp$, if no session key is generated by $\mathcal{P}^x$. Otherwise, it results into flipping of a coin ω. $Test$ returns the session key if $\omega = 1$. Contrarily, if $\omega = 0$ it returns a uniformly distributed random string with equal length as of the actual session key.

Now, we define some of the definitions pertinent to the security of proposed protocol.

Definition 6 (Acceptance)**.** An instance $\mathcal{P}^x$ is accepted, if the involved participant has considered it legal. If accepted $\mathcal{P}^x$ will have all the messages sent and received as session identifier.

Definition 7 (Partnering)**.** Two instance $\mathcal{P}^x \in \mathcal{MN}_j$ and $\mathcal{P}^y \in \mathcal{AP}_i$ are termed as partner subject to satisfaction of following conditions, (i) both $\mathcal{P}^x$ and $\mathcal{P}^y$ are accepted and (ii) both are participating in same session.

Definition 8 (Fresh)**.** An instance $\mathcal{P}^x \in \{\mathcal{MN}_j, \mathcal{AP}_i\}$ is said to be fresh, if it posses a session key and no reveal query has been performed on $\mathcal{P}^x$.

Definition 9 (HAP-Security)**.** The advantage for $\mathcal{A}$ to break the security of a handover authentication protocol (HAP) is the probability to possibly guess the result of coin flipping ω by $Test(\mathcal{P}^x)$, where $\mathcal{P}^x$ is $fresh$ as well as accepted. Let $\mathcal{A}$ outputs ω', the advantage is as follows:

$$Adv^{HAP}(\mathcal{A}) = |Pr[\omega = \omega'] - \frac{1}{2}| \tag{9.9}$$

The proposed authentication protocol is termed as HAP-secure if $Adv^{HAP}(\mathcal{A}) \leq \epsilon$, for some negligible function ϵ.

Definition 10 (Negligible function)**.** A function ϵ is said to be negligible if, for every $c > 0$, there exists k_0 such that $\epsilon \leq \frac{1}{k^c}$ for every $k \geq k_0$.

Definition 11 (Elliptic Curve Discrete Logarithm Problem (ECDLP))**.** Given two random point $R, S \in E/F_p$, find a scalar v such that $R = vS$. The probability that an algorithm $\mathcal{D}$ can compute v in polynomial time t is as follows: $Adv_{\mathcal{D}}^{ECDLP}(t) = Pr[\mathcal{D}(S, R) = v : v \in Z_q^*]$. The ECDLP assumption implies that $Adv_{\mathcal{D}}^{ECDLP}(t) \leq \epsilon$.

Definition 12 (Elliptic Curve Computational Diffie-Hellman (ECCDH) problem)**.** Given three point S, αS and βS over an elliptic curve E/F_p, where α, $\beta \in Z_q^*$. The probability that an algorithm $\mathcal{D}$ can compute $\alpha\beta S$ in polynomial time t is as follows: $Adv_{\mathcal{D}}^{ECCDH}(t) = Pr[\mathcal{D}(S, \alpha S, \beta S) = \alpha\beta S : \alpha, \beta \in Z_q^*]$. The ECCDH assumption implies that $Adv_{\mathcal{D}}^{ECCDH}(t) \leq \epsilon$.

Theorem 5. The proposed handover authentication protocol achieves mutual authentication between $\mathcal{AP}_i$ and $\mathcal{MN}_j$ provided the $ECDLP$ assumption holds and H_1, H_2 are modeled as random oracles. Contrarily if an adversary $\mathcal{A}$ can violate authentication between $\mathcal{AP}_i$ and $\mathcal{MN}_j$ with probability ϵ, then there exists a polynomial time algorithm $\mathcal{D}$, who can solve $ECDLP$ with at least probability ϵ', where number of H_1 queries are bounded as q_{h1} and corrupt queries are by q_c.

$$\epsilon' \geq (\frac{1}{q_{h1}})(1 - \frac{q_c}{q_{h1}})\epsilon \tag{9.10}$$

Proof. If $\mathcal{A}$ can violate the mutual authentication between $\mathcal{AP}_i$ and $\mathcal{MN}_j$, then a polynomial time algorithm $\mathcal{D}$ can be constructed to solve the ECDLP. The algorithm $\mathcal{D}$ maintains three lists L_{h1}^l, L_{h2}^l and L_{Ex}^l as follows:

- The list L_{h1}^l: It contains the tuples of the form (in_{1j}, ou_{1j}), where in_{1j} is the input tuple to the hash function H_1 and ou_{1j} is the corresponding output.
- The list L_{h2}^l: It contains the tuples of the form (in_{2j}, ou_{2j}), where in_{2j} is the input tuple to the hash function H_2 and ou_{2j} is the corresponding output.

- Extract list L_{Ex}^{list}: It contains the tuples of the form (PID_j, R_j, s_j).

The proof is consisting of challenge response interactive game played between adversary $\mathcal{A}$ and $\mathcal{D}$. The hash function H_1, H_2 are assumed to be random oracles. To violate proposed protocol's security, $\mathcal{D}$ and $\mathcal{A}$ interacts as follows:

- **Setup:** $\mathcal{D}$ executes system setup algorithm, and generates $\{p, q, E/F_p, P, G, PK = sP, H_1, H_2, \mathcal{F}\}$, where $P, PK = sP$ is is an instance of ECDLP. Then $\mathcal{D}$ returns the system parameters to $\mathcal{A}$. Note that $\mathcal{D}$ does not know master secret key $s \in Z_q^*$. $\mathcal{A}$ performs following queries and gets the respective outputs from $\mathcal{D}$ as follows:
- **Queries:** $\mathcal{A}$ can issue H_1, H_2, **Extract** and **Send** queries. Then, $\mathcal{D}$ responds as follows:
  - H_1: Assume that $\mathcal{A}$ asks H_1 query with the input (PID_j, R_j), $\mathcal{D}$ responds with a tuple (PID_j, R_j, h_j) if it exists in L_{h1}^l. Otherwise, $\mathcal{D}$ selects a random number h_j, outputs it and saves the new tuple (PID_j, R_j, h_j) in L_{h1}^l.
  - H_2: Assume that $\mathcal{A}$ asks a H_2 query with the input (PID_j, m_j), $\mathcal{D}$ responds with a tuple (PID_j, m_j, A_j) if it exists in L_{h2}^l. Otherwise, $\mathcal{D}$ selects a random number A_j, outputs it and saves the new tuple (PID_j, m_j, A_j) in L_{h2}^l.
  - **Extract(ID_x) queries:** When $\mathcal{D}$ received a *Extract*(ID_x) query, $\mathcal{D}$ selects two numbers $h_x, r_x \in Z_q^*$, sets $s_x = r_x$, $H_1(ID_x, R_x) = h_x$ and $R_x = s_xP - h_xPK$. Note that $s_x = r_x$ satisfies the equation $R_x + H_1(ID_x, R_x)PK = s_xP$. Now, $\mathcal{D}$ does as follows:
    - If $ID_x \in \{PID_j, ID_i\}$, $\mathcal{D}$ returns $s_x = \perp$ to $\mathcal{A}$ and adds the tuples (ID_x, R_x, h_x) and $(ID_x, R_x, \perp)$ in L_{h1}^l and L_{Ex}^l, respectively.
    - Else, $\mathcal{C}$ returns $s_x = r_x$ to $\mathcal{A}$ and adds (ID_x, R_x, h_x) and (ID_x, R_x, s_x) in L_{h1}^l and L_{Ex}^l, respectively.
  - **Corrupt**(ID_x): When this query is asked, $\mathcal{D}$ did as follows:
    - If $ID_x \in \{PID_j, ID_i\}$, access (ID_x, R_x, h_x) and $(ID_x, R_x, \perp)$ from L_{h1}^l and L_{Ex}^l and return them to $\mathcal{A}$.
    - Else, access (ID_x, R_x, h_x) and (ID_x, R_x, s_x) from L_{h1}^l and L_{Ex}^l and return them to $\mathcal{A}$.
  - **Send queries:** When $\mathcal{A}$ makes a *Send*(ID_x) for m_x, $\mathcal{D}$ access the lists L_{Ex}^l and L_{h1}^l, then responds as follows:

* If $ID_x = PID_j$, $\mathcal{D}$ does as follows:
  - Chooses $Y_x \in Z_q^*$ and compute $A_x = Y_x P - z_x(R_x + h_x PK)$.
  - Outputs (PID_x, R_x, A_x, t_x, Y_x).
* Else, $\mathcal{D}$ does as follows:
  - Chooses a number $a_x \in Z_p^*$ and compute $A_x = a_x P$.
  - Computes $Y_x = a_x + z_x s_x$.
  - Outputs (PID_x, R_x, A_x, t_x, Y_x).

- **Output:** By applying forking lemma [165], $\mathcal{A}$ can output two different authentication requests messages $\{PID_j, R_j, A_j, t_j, Y_j\}$ and $\{PID_j, R_j, A_j, t_j, Y_j'\}$, with different hash values $z_j' \neq z_j$. For valid request messages, we can write:

$$Y_j P = A_j + z_j(h_j PK + R_j) \tag{9.11}$$
$$Y_j' P = A_j + z_j'(h_j PK + R_j) \tag{9.12}$$

Using the equations (9.11) and (9.12), we have:

$$(Y_j' - Y_j)P = (z_j' - z_j)(h_j PK + R_j) \tag{9.13}$$

As $PK = sP$ and $R_j = r_j P$, and thus the equation 9.13 can be written as

$$sP = \left[\frac{(Y_j' - Y_j)}{h_j(z_j' - z_j)} - \frac{r_j}{h_j}\right] P \tag{9.14}$$

Finally, we have

$$s = \left[\frac{(Y_j' - Y_j)}{h_j(z_j' - z_j)} - \frac{r_j}{h_j}\right] \tag{9.15}$$

Hence, $\mathcal{D}$ can break $ECDLP$ as $s = \left[\frac{((Y_j' - Y_j)}{h_j(z_j' - z_j)} - \frac{r_j}{h_j}\right]$.

- **Success probability:** $\mathcal{D}$ aborts the simulation for following two events:
  $E1 : A$ returns forgery for ID_x other than the chosen $ID_x = ID_y$.

$E2 : A$ makes corrupt query on ID_y.
We have $Pr[E1] = 1 - \frac{1}{q_{h1}}$ and $Pr[E2] = q_c(\frac{1}{q_{h1}})$. Therefore, the probability that $\mathcal{D}$ will not abort is: $Pr[\neg E1 \wedge \neg E2] = (\frac{1}{q_{h1}})(1 - \frac{q_c}{q_{h1}})$. Hence, $\mathcal{D}$ can solve $ECDLP$ with success probability: $\epsilon' \geq (\frac{1}{q_{h1}})(1 - \frac{q_c}{q_{h1}})\epsilon$, where ϵ is the probability for $\mathcal{A}$ to win the game.

□

Theorem 6. The proposed handover authentication protocol can achieve authentication between $\mathcal{AP}_i$ and $\mathcal{MN}_j$ provided that ECCDH problem is hard to break by any polynomial time algorithm.

Proof. If $\mathcal{A}$ can violate the mutual authentication between $\mathcal{AP}_i$ and $\mathcal{MN}_j$, then a polynomial time algorithm $\mathcal{D}$ can be constructed to solve ECCDH problem. $\mathcal{D}$ maintains the following lists L^l_{h1}, L^l_{h2}, L^l_{Ex}, L^l_f and L^l_λ as follows:

- The list L^l_{h1}: It contains the tuples of the form (in_{1j}, ou_{1j}), where in_{1j} is the input tuple to the hash function H_1 and ou_{1j} is the corresponding output.
- The list L^l_{h2}: It contains the tuples of the form (in_{2j}, ou_{2j}), where in_{2j} is the input tuple to the hash function H_2 and ou_{2j} is the corresponding output.
- The list L^{list}_{Ex}: It contains the tuples of the form (PID_j, R_j, s_j).
- The list L^l_f: It contains the tuples of the form (in_{fj}, ou_{fj}), where in_{fj} is the input tuple to the hash function $\mathcal{F}$ and ou_{fj} is the corresponding output.
- The list L^l_λ: It contains the tuples of the form $(in_{\lambda_j}, ou_{\lambda_j})$, where in_{λ_j} is the input tuple to the hash function λ and ou_{λ_j} is the corresponding output.

The proof is consisting of challenge response interactive game played between adversary $\mathcal{A}$ and $\mathcal{D}$. The one way function λ is assumed to be the random oracle. To violate the security of proposed handover authentication protocol, $\mathcal{D}$ and $\mathcal{A}$ interacts as follows:

- **Setup:** $\mathcal{D}$ keeps keeps (P, vP, uP) as the instance of ECDHP and outputs system parameters $\{p, q, E/F_p, P, G, PK = sP, H_1, H_2, \mathcal{F}, \lambda\}$. $\mathcal{A}$ performs following queries and gets the respective outputs:
- **Queries:** $\mathcal{A}$ simulates following queries to violate security of proposed protocol.
  - λ: Assume that $\mathcal{A}$ asks λ query with the input $(PID_j, ID_i, B_i, A_j, K_{ij})$, $\mathcal{D}$ responds with MAC_{ij}, if it exists in L^l_λ. Otherwise, $\mathcal{D}$ selects a random number

MAC_{ij} and out puts and saves the new tuple (PID_j, ID_i, B_i, A_j, K_{ij}, MAC_{ij}) in L_λ^l.

- **Extract(ID_x) queries:** When $\mathcal{D}$ received a *Extract*(ID_x) query, $\mathcal{D}$ does as follows:
  - If ($ID_x = PID_j$), $\mathcal{D}$ searches a tuple (PID_j, R_j, h_j) into L_{h1}^l and computes $R_j = uP - h_iPK$. Therefore, $R_j + h_jPK = uP - h_jPK + h_jPK = uP$. Thus, u acts as the private key of PID_j. Then, $\mathcal{D}$ sets $s_j = \perp$ as PID_j's private key and returns $s_j = \perp$ to $\mathcal{A}$ and adds the tuples (PID_j, R_j, $\perp$) and (PID_j, R_j, h_j) to L_{Ex}^l and L_{h1}^l, respectively.
  - If ($ID_x = ID_i$), $\mathcal{D}$ searches a tuple (ID_i, R_i, h_i) into L_{h1}^l and computes $R_i = vP - h_iPK$. Therefore, $R_i + h_iPK = vP - h_iPK + h_iPK = vP$. Thus, v acts as the private key of ID_i. Then, $\mathcal{D}$ sets $s_i = \perp$ as ID_i's private key and returns $s_i = \perp$ to $\mathcal{A}$ and adds the tuples (ID_i, R_i, $\perp$) and (ID_i, R_i, h_i) to L_{Ex}^l and L_{h1}^{list}, respectively.
  - Else, $\mathcal{D}$ selects two numbers h_x, $r_x \in Z_q^*$, sets $s_x = r_x$, $H_1(ID_x, R_x) = h_x$ and $R_x = r_xP - h_xPK$. Note that $s_x = r_x$ satisfies the equation $R_x + h_xPK = s_xP$. $\mathcal{D}$ returns s_x to $\mathcal{A}$ and adds the tuples (ID_x, R_x, d_x) and (ID_x, R_x, h_x) to L_{Ex}^l and L_{h1}^l, respectively.
- **Corrupt(ID_x) queries:** When $\mathcal{D}$ received a *Corrupt*(ID_x) query from $\mathcal{A}$, $\mathcal{D}$ does as follows:
  - $\mathcal{C}$ returns $\perp$ if $ID_x \in \{PID_j, ID_i\}$ holds.
  - Else, $\mathcal{D}$ searches the lists L_{Ex}^l and returns the private key s_x if there is a tuple (ID_x, R_x, s_x). Otherwise, $\mathcal{D}$ executes $H_1(ID_x)$ and *Extract*(ID_x) queries for the tuples (ID_x, R_x, h_x) and (ID_x, R_x, s_x), then outputs s_x as the private key. $\mathcal{D}$ adds the tuples (ID_x, R_x, h_x) and (ID_x, R_x, s_x) to L_{Ex}^l and L_{h1}^l, respectively.
- **Send queries:** When $\mathcal{A}$ makes a *Send* query, $\mathcal{D}$ accesses a tuple ($PID_j, ID_i, B_i, A_j, K_{ij}$) from the list L_λ^l and returns K_{ij} to $\mathcal{A}$.
- **Reveal queries:** When $\mathcal{A}$ simulates a *Reveal* query on ID_x, $\mathcal{D}$ abandons the protocol execution if $ID_x \in \{PID_j, ID_i\}$. $\mathcal{D}$ outputs random $r \in Z_q^*$ if the matching session is not accepted. Otherwise $\mathcal{D}$ outputs the existent session key.
- **Test queries:** If this query is asked in corresponding session, $\mathcal{D}$ selects random

bit ω, if $\omega = 1$, $\mathcal{D}$ returns it to $\mathcal{A}$. Otherwise, $\mathcal{D}$ returns some random value.

- **Output:** $\mathcal{A}$ can forge $\mathcal{AP}_i$, if he become able to generate valid response message ID_i, B_i, MAC_{ij}.
  - The probability $\mathcal{A}$ can guess $MAC_{ij} = \lambda(PID_j, ID_i, B_i, A_j, K_{ij})$ without simulating λ and knowledge of K_{ij} or K_{ji} is $\frac{1}{2^k}$.
  - For $\mathcal{AP}_i$, $R_i = vP - h_jPK$ and for $\mathcal{MN}_j$, $R_j = uP - h_jPK$, hence $uvP = K_{ij} - b_il_i(uP) - a_jl_j(vP) - (l_il_j)(a_jb_iP)$. Hence, $\mathcal{A}$ can correctly guesses MAC_{ij} without knowing K_{ij}, if he can solve the $ECCDH$ problem.
- **Success Probability:** Now, we will analyze $\mathcal{A}$'s forging capabilities.
  - The probability $\mathcal{A}$ can guess $MAC_{ij} = \lambda(PID_j, ID_i, B_i, A_j, K_{ij})$ without simulating λ and knowledge of K_{ij} or K_{ji} is $\frac{1}{2^k}$.
  - For $\mathcal{AP}_i$, $R_i = vP - h_jPK$ and for $\mathcal{MN}_j$, $R_j = uP - h_jPK$, hence $uvP = K_{ij} - b_il_i(uP) - a_jl_j(vP) - (l_il_j)(a_jb_iP)$. Hence, $\mathcal{A}$ can correctly guesses MAC_{ij} without knowing K_{ij}, if he can solve the $ECCDH$ problem.

  The probability that $\mathcal{A}$ can break the $ECCDH$ problem is $\epsilon' \geq \epsilon - \frac{1}{2^k}$.

□

Theorem 7. The proposed handover authentication protocol ensures anonymity and privacy of the mobile node.

Proof. In proposed scheme $\mathcal{AS}$ assigns a number of one time pseudo identities PID_1, PID_2, $\cdots$, PID_n. During authentication $\mathcal{MN}_j$ uses his unlink-able pseudo identity PID_j, the real identity ID_j of $\mathcal{MN}_j$ is not even revealed to $\mathcal{AP}_i$. Therefore, advantage carried by an adversary $\mathcal{A}$ to break $\mathcal{MN}_j$'s anonymity is negligible. Hence proposed protocol protects $\mathcal{MN}_j$'s anonymity and privacy.

□

9.6.2 Simple Proof of Security Requirements

This subsection accommodates the simple security requirements proof as mentioned in subsection 9.1.3. The security of protocol is dependent on elliptic curve discrete logarithm problem (ECDLP) and computational Diffie-Hellman assumptions. This subsection first

proves the correctness of our proposed protocol then describes detail of each design goal proof.

- **Correctness:** In proposed protocol the session keys computed by $\mathcal{MN}_j$ and $\mathcal{AP}_i$ are same. The computation of session key involves three parameters out of which two parameters PID_j, ID_i are public, while $\mathcal{MN}_j$ computes K_{ij} and $\mathcal{AP}_i$ computes K_{ij}. The session key computed on both side is same if and only if $K_{ij} = K_{ji}$, the proof is as follows:

$$\begin{aligned} K_{ij} &= (s_i + b_i l_i)(h_j PK + R_j + l_j A_j) \\ &= (s_i + b_i l_i)(h_j sP + r_j P + l_j a_j P) \\ &= (s_i + b_i l_i)(s_j P + l_j a_j P) \\ &= (s_i + b_i l_i)(s_j + l_j a_j)P \\ &= (s_j + l_j a_j)(s_i + b_i l_i)P \\ &= (s_j + l_j a_j)(s_i P + b_i l_i P) \\ &= (s_j + l_j a_j)(h_i sP + r_i P + l_i b_i P) \\ &= (s_j + l_j a_j)(h_i PK + R_i + l_i B_i) \\ &= K_{ji} \end{aligned} \tag{9.16}$$

  Thus, we have $k_{ji} = \mathcal{F}(K_{ji}, PID_j, ID_i) = \mathcal{F}(K_{ij}, PID_j, ID_i) = k_{ij}$. Hence, the session keys computed on both sides are same.

- **Mutual Authentication & Key agreement:** $\mathcal{MN}_j$ sends $\{PID_j, R_j, A_j, t_j, Y_j\}$ to $\mathcal{AP}_i$. Upon receiving the message $\mathcal{AP}_i$ verifies authenticity of $\mathcal{MN}_j$ by verifying signatures $Y_j P \stackrel{?}{=} A_j + z_j(h_j PK + R_j)$, where $z_j = H_2(PID_j, A_j, m_j)$ and $h_j = H_1(PID_j, R_j)$, only valid $\mathcal{MN}_j$ can pass this test, as computation of Y_j requires the secret key s_j and session specific parameter a_j. Furthermore, $\mathcal{AP}_i$ computes and sends MAC_{ij} to $\mathcal{MN}_j$, which verifies $MAC_{ij} \stackrel{?}{=} MAC_{ji}$, where $MAC_{ji} = \lambda(PID_j, ID_i, B_i, A_j, K_{ji})$, if it holds, $\mathcal{AP}_i$ is treated as legal access point. The computation of MAC_{ij} requires the access of $\{PID_j, ID_i, B_i, A_j, K_{ij}\}$, out of these parameters only K_{ij} is private and can only be computed by legal $\mathcal{AP}_i$, because K_{ij} is computed using the secret key s_i of $\mathcal{AP}_i$ along with session specific b_i and l_i, where $l_i = H_2(ID_i, R_i, B_i)$. Hence both $\mathcal{MN}_j$ and $\mathcal{AP}_i$ are mutually authenticated. Furthermore, the session key generated k_{ij} contains session specific secret parameters b_i and a_j from both $\mathcal{AP}_i$ and $\mathcal{MN}_j$. So the *forward secrecy* of the session key is provided in proposed protocol.

Table 9.2: Comparison of Security Parameters

Scheme:	Proposed	[9]	[17]	[163]	[162]	[160]
Anonymity and privacy	Yes	Yes	Yes	Yes	Yes	Yes
Mutual Authentication	Yes	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes	Yes	Yes
Resists $\mathcal{MN}$ impersonation	Yes	Yes	Yes	Yes	Yes	Yes
Resists $\mathcal{AP}$ impersonation	Yes	No	Yes	Yes	Yes	Yes
Resists Replay attack	Yes	Yes	Yes	Yes	Yes	Yes
Resists key compromise attack	Yes	Yes	Yes	Yes	Yes	No
Provable Security	Yes	Yes	Yes	Yes	Yes	Yes

9.6.3 Automated Security Verification through ProVerif

To analyze our protocol, we model the steps as mentioned in section 9.5. Then we check the secrecy of the session key and the reachability property as shown in Fig. 9.5. Finally we got the results as follows:

```
1 RESULT inj event(endMnode(id)) ==> inj event(beginMnode(id)) is true.
2 RESULT inj event(endAPoint(id_1449)) ==> inj event(beginAPoint(id_1449)) is
    true.
3 RESULT not attacker(kma[]) is true.
```

The results (1) and (2) shows that mobile node process as well as access point process started and terminated successfully, while (3) shows that attacker is unable to find kma. Hence, authentication and secrecy is preserved.

9.7 Security and Performance Comparisons

This section briefly compares the performance and security comparisons of proposed protocol with related existing protocols [9,160,162,163]. Table 9.2 illustrates the security features. The proposed scheme, and schemes proposed in [17,162,163] are possessing all security features, while scheme proposed in [160] is vulnerable to key compromise attack and scheme in [9] is vulnerable to access point impersonation attack.

In order to understand the performance comparisons, we define some notations as follows:

- t_{bp}: Time to compute a bilinear pairing operation.
- t_{bsm}: Time to compute scalar point multiplication based on pairing.
- t_{mtp}: Time to compute a map to point function.
- t_{epm}: Time to compute elliptic curve scalar point multiplication.

```
(***************** Channels ****************)
free ch1:channel [private].
free ch2:channel [private].
free ch3:channel.
(************ Names & Variables ************)
const P: bitstring.
const p: bitstring.
const q: bitstring.
const Ti: bitstring.
free IDi: bitstring.
free PK: bitstring.
free PIDj: bitstring.
(** Constructors*destructors*Equations **)
fun concat(bitstring,bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun add(bitstring,bitstring):bitstring.
fun syme(bitstring,bitstring):bitstring.
fun inverse(bitstring):bitstring.
fun sig(bitstring):bitstring.
fun h1(bitstring):bitstring.
fun h2(bitstring):bitstring.
fun F(bitstring,bitstring,bitstring):bitstring.
fun lambda(bitstring,bitstring,bitstring,
     bitstring,bitstring):bitstring.
reduc forall m : bitstring, key : bitstring;
     symd(syme(m, key), key) = m.
(******************events******************)
event beginAPoint(bitstring).
event endAPoint(bitstring).
event beginMnode(bitstring).
event endMnode(bitstring).
```

(a) Declarations

```
(*********Authentication Server**********)
let pAServer=
new s: bitstring;
let PK = mult(s, P) in
new ri : bitstring;
let Ri = mult(ri, P) in
let hi = h1(concat(IDi, Ri)) in
let si = add(ri,mult(s, hi)) in
out(ch1, (si, Ri));
new rj : bitstring;
let Rj = mult(rj, P) in
let hj = h1(concat(PIDj, Rj)) in
let sj = add(rj, mult(s, hj)) in
out(ch2, (sj, Rj)).
(***************Access Point***************)
let pAPoint=
in(ch1,(Xsi:bitstring,XRi:bitstring));
let Pki = mult(Xsi,P) in
event beginAPoint(IDi);
out(ch3,(XRi,IDi));
in(ch3,(XPIDj:bitstring,XRj:bitstring,XA':
     bitstring,Xtj:bitstring,XY:bitstring));
let m' = concat(XPIDj,concat(XRj,concat(A,Xtj))
     ) in
let YP= mult(XY,P) in
let YP'=add(XA,mult(h2(concat(XPIDj,concat(XA,m
     '))),add(mult((h1(concat(XPIDj,XRj))),PK),
     XRj))) in
if(YP = YP')
then
new b:bitstring;
let B=mult(b,P) in
let KAM=mult(add(Xsi,mult(b,h1(concat(B,concat(
     IDi,XRi))))),add(mult(h1(concat(XPIDj,XRj)
     ),PK),add(XRj,XA)))in
let kam=F(KAM,XPIDj,IDi) in
let MACij=lambda(XPIDj,IDi,B,A,KAM) in
out(ch3,(IDi,B,MACij));
event endAPoint(IDi).
(***************Mobile Node***************)
let pMnode=
in(ch2,(Xsj:bitstring,XRj:bitstring));
let Pkj = mult(Xsj,P) in
in(ch3,(XRi:bitstring,XXIDi:bitstring));
event beginMnode(PIDj);
new a: bitstring;
let A = mult(a,P) in
new tj:bitstring;
let m = concat(PIDj,concat(XRj,concat(A,tj)))
     in
let Y = add(a,mult(Xsj,h2(concat(PIDj,concat(A,
     m))))) in
out(ch3,(PIDj,XRj,A,tj,Y));
in(ch3,(XIDi:bitstring,XB:bitstring,XMACij:
     bitstring));
let KMA =mult(add(Xsj,a),add(add(mult(h1(concat
     (IDi,XRi)),PK),XRj),mult(XB,h1(concat(XB,
     concat(IDi,XRi)))))) in
let kma = F(KMA,PIDj,XIDi) in
let MACij = lambda(PIDj,XIDi,XB,A,KMA) in
if(MACij = XMACij)
then
event endMnode(PIDj).
```

(b) Processes

```
(********Process Replication***********)
process ((pClientU)||(!pServerS))
(************** *queries* **************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj event(end_User(id)) ==>
     inj event(begin_User(id)) .
query id:bitstring; inj event(end_Server(id))
     ==>inj event(begin_Server(id)) .
```

(c) Main

Figure 9.5: ProVerif Validation

Table 9.3: Performance Analysis

Protocol	Computation Cost of $\mathcal{MN}_j$	Computation Cost of $\mathcal{AP}_i$	Total execution time
He et al. [160]	$t_{bp} + t_{bsm} + t_{mtp} \approx 29.46$	$3t_{bp} + t_{mtp} \approx 63.16$	$\approx 92.62ms$
Tsai et al. [162]	$t_{bp} + 2t_{bsm} + 2t_{mtp} \approx 35.84$	$3t_{bp} + 2t_{mtp} \approx 66.2$	$\approx 102.04ms$
Islam and Khan [163]	$3t_{epm} \approx 6.63$	$5t_{epm} \approx 11.05$	$\approx 17.68ms$
He et al. [17]	$t_{bp} + 2t_{bsm} + 5t_{mtp} \approx 47.92$	$4t_{bp} + t_{bsm} + 5t_{mtp} \approx 102.60$	$\approx 150.52ms$
Li et al. [9]	$3t_{epm} \approx 6.63$	$4t_{epm} \approx 8.84$	$\approx 15.47ms$
Proposed	$2t_{epm} \approx 4.42$	$4t_{epm} \approx 8.84$	$\approx 13.20ms$

Table 9.3 demonstrate the computation cost comparisons of proposed protocol with related existing protocols. The experimental computation time mentioned in [166] are as follows: t_{bp} takes approximately $20.04ms$, t_{bsm} takes 6.38, time for t_{epm} is $2.21ms$, while t_{mtp} takes $3.04ms$. The computation time for one way hash function is considered to be negligible.

During handover authentication phase of He et al.'s protocol [160] $\mathcal{MN}_j$ performs $t_{bp} + t_{bsm} + t_{mtp}$ operations and $\mathcal{AP}_i$ performs $3t_{bp} + 1t_{mtp}$ operations, the total handover authentication time is approximately $92.62ms$. In Tsai et al.'s protocol [162] number of operations for $\mathcal{MN}_j$ are $t_{bp} + 2t_{bsm} + 2t_{mtp}$ and for $\mathcal{AP}_i$ number of operations are $3t_{bp} + 2t_{mtp}$, the time consumed during handover authentication is approximately $102.04ms$. In Islam and Khan's handover authentication protocol [163], $\mathcal{MN}_j$ takes $3t_{epm}$ operations and $\mathcal{AP}_i$ takes $5t_{epm}$ operations, total time taken for handover authentication is approximately $17.68ms$. In He et al.'s protocol [17], $\mathcal{MN}_j$ takes $t_{bp} + 2t_{bsm} + 5t_{mtp}$ operations and $\mathcal{AP}_i$ takes $4t_{bp} + t_{bsm} + 5t_{mtp}$ operations, total time taken for handover authentication is approximately $150.52ms$. In Li et al.'s protocol's [9] $\mathcal{MN}_j$ executes $3t_{epm}$ operations and $\mathcal{AP}_i$ executes $4t_{epm}$ operations, the total handover execution time is approximately $15.47ms$. During handover authentication phase of the proposed protocol, $\mathcal{MN}_j$ performs only $2t_{epm}$ operations, and $\mathcal{AP}_i$ takes $4t_{epm}$ operation, the total running time for handover authentication time is approximately $13.20ms$. The proposed protocol has reduced one t_{epm} operation performed by $\mathcal{MN}_j$ as compared to Li et.al.'s protocol, while it has reduced one t_{epm} operation on both $\mathcal{MN}_j$ and $\mathcal{AP}_i$ as compared to Islam and Khan's protocol, the proposed protocol has over casted Tsai et al.'s and He et al.'s protocol.

9.8 Chapter Summary

In this chapter, we analyzed Li et al.'s privacy-aware handover authentication protocol for wireless networks, and found that it is vulnerable to access point impersonation attack. We then put forwarded an improved protocol to overcome the security weakness of Li et al.'s protocol. We analyzed that the improved protocol is provably secure in the random oracle

model against the hardness assumptions of the elliptic curve discrete logarithm problem and elliptic curve computational Diffie-Hellman problem. In addition, we formally validated the security of improved protocol using widespread automated tool ProVerif. The proposed protocol ensures user anonymity and robustness against all known attacks while reducing overall computation as compared with other related protocols.

Chapter 10

A multi-server Authentication Scheme using ECC

Big data refers to the huge amount of data with complicated and diverse structure to be stored and analyzed for retrieving results. This kind of result retrieval is known as big data analysis, which is performed by disclosing concealed pattern and correlations present in the colossal data. Big data analysis is playing a vital role in present day businesses and contemporary science, because it helps organizations and companies to attain competitive benefits through deeper and wealthier insights into precious gigantic data. There are numerous sources for such gigantic data, social networking interaction is one of them. Huge social networking data storage, manipulation and transfer becomes difficult to manage and can be compromised by various security attacks therefore efficient authentication mechanism should be developed to make it more secure and reliable. Moreover, social networking services are inherently multi-server environments. Therefore, authentication schemes must be specifically designed for multi-server architecture in order to maintain compatibility.

Tsai et al. [80] in 2008, presented one of the first efficient authentication scheme for multi-server environment. This scheme comprises only hash functions and random numbers in order to achieve sufficient security at lower computation cost and after that numerous similar schemes are designed for multi-server architecture [81–83]. Yoon et al. [167] introduced authentication scheme based on biometrics in 2013, this scheme is designed for multi-server architecture. He et al. [133] however stated in 2014 that Yoon et al.'s scheme can be easily compromised by the smart card stolen and impersonation attacks. He et al. then introduced an enhanced scheme to mitigate concerns present in Yoon et al.'s scheme.

In 2014, Xue et al.'s [168] declared that the Li et al.'s key exchange authentication scheme [98]

presented in 2012 is vulnerable to denial of service, offline password guessing, replay, stolen verifier and forgery attacks. Therefore, Xue et al.'s presented an enhanced scheme to overcome shortcomings of Li et al.'s scheme. In 2015, Lu et al. [169] exposed the vulnerability of the Xue et al.'s scheme against offline password, masquerade and insider attacks. Lu et al. remove the shortcoming of Xue et al.'s scheme and presented an enhanced scheme for multi-server architecture. The schemes discussed so far, offer two factor authentication using smart cards and password. Due to the demerits of the existing two factor authentication schemes, there is a need of biometrics based three factor authentication scheme. A number of such authentication schemes are readily available [151–153, 170–172]. However, most of the three factor authentication schemes are designed specifically for single server architecture making it incompatible for multi-server architecture. Chuang et al. in 2014 [50] introduced authentication scheme utilizing biometrics and smart card for multi-server architecture and declared it to be secure against the known attacks. Soon, Mishra et al. [94] identified that Chuang et al.'s scheme is not invincible to server spoofing, smart card stolen and impersonation attacks. Further, Mishra et al. proposed key agreement authentication scheme using smart card and biometrics. Mishra et al. declared it to be secure against all security threats. Later on, Lu et al. [10, 11] recognized that Mishra et al.'s scheme is vulnerable to server spoofing and impersonation attacks and fails to provide forward secrecy. In response to Mishra et al.'s scheme Lu et al. introduced two independent authentication schemes [10, 11] based on three factor biometrics for multi-server architecture and declared that their schemes are invincible against the known attacks. But this chapter provide an evidence that Lu et al.'s both schemes can be compromised by Well-known attacks. The Lu et al.'s first scheme is insecure against user anonymity violation and impersonation attacks, whereas Lu et al.'s second scheme is insecure against user impersonation attack. This chapter exhibits that by knowing the public identity of any the other user, the unfair user of the system can impersonate him easily.

Rest of the chapter is structured as follows: Section 10.1 presents review of two Lu et al.'s authentication schemes based on three factor for multi-server environments, followed by their cryptanalysis performed in Section 10.2. The proposed scheme is discussed in Section 10.3. The formal and informal security analysis is performed in section 10.4 followed by automated security validation in section 10.5. The performance evaluation is shown in Section 10.6. Finally, chapter's summary is solicited in Section 10.7.

Table 10.1: Notation Guide

Notations	Description
RC, $\mathcal{S}_y$, $\mathcal{U}_x$, $\mathcal{A}$	Registration center, Server, User, Attacker
SID_y, ID_{ux}, PW_{ux}, BIO_{ux}	Identities of $\mathcal{S}_y$, $\mathcal{U}_x$, $\mathcal{U}_x$'s password, and biometrics
x_{ux}, Pub_{sy}, Pri_{sy}	$\mathcal{U}_x$'s secret key, public and private key pair of $\mathcal{S}_y$
y_{rc}, PSK_{rs}	RC's secret key, secret key between $\mathcal{S}_y$ and RC
SC_{ux}, $h(.)$, $H(.)$, $\|$, $\oplus$	$\mathcal{U}_i$'s smart card, Hash, BioHash functions, Concatenation, XOR operators

10.1 Review of Lu et al.'s Schemes

In this section, we briefly review Lu et al.'s multi-server biometrics based authentication schemes [10, 11] in subsection 10.1.1 and 10.1.2, respectively.

10.1.1 Review of Lu et al.'s Scheme-1

Lu et al.'s biometrics based authentication scheme for multi-server environments [11] is illustrated in Fig. 10.1 and is elaborated in following three phases:

10.1.1.1 Registration Phase

$\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux} and imprints his biometrics BIO_{ux}. Further, $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux}\|H(BIO_{ux}))\}$ to RC on a private channel. Upon reception, RC computes $X_{ux} = h(ID_{ux}\|y_{rc})$, $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, then stores $X_{ux}, h(PSK_{rs})$ and V_{ux} in the smart card SC_{ux}. RC sends smart card (SC_{ux}) to $\mathcal{U}_x$. Upon reception of smart card, $\mathcal{U}_x$ computes $Y_{ux} = h(PSK_{rs}) \oplus x_{ux}$. Finally, smart card contains $X_{ux}, Y_{ux}, V_{ux}, h(.)$.

10.1.1.2 Login and Authentication Phase

$\mathcal{U}_x$ enters his smart card in specialized reader and inputs his biometrics BIO_{ux} , password PW_{ux} and identity ID_{ux}. Following steps are performed between the smart card (SC_{ux}) and the server $\mathcal{S}_y$:

Step L1A1: SC_{ux} checks $V_{ux} \stackrel{?}{=} h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, if it is not true, session is aborted by SC_{ux}. Otherwise SC_{ux} computes $K = h(Y_{ux} \oplus x_{ux})\|SID_{sy})$ and $M_1 = K \oplus ID_{ux}$. Then SC_{ux} generates a nonce $M_2 = n_{ux} \oplus K$, $M_3 = K \oplus h(PW_{ux}\|H(BIO_{ux}))$ and $Z_{ux} = h(X_{ux}\|n_{ux}\|h(PW_{ux}\|H(BIO_{ux})\|T_1))$, where T_1 is the fresh timestamp.

Step L1A2: Smart card SC_{ux} sends $\{M_1, M_2, M_3, Z_{ux}, T_1\}$ to $\mathcal{S}_y$.

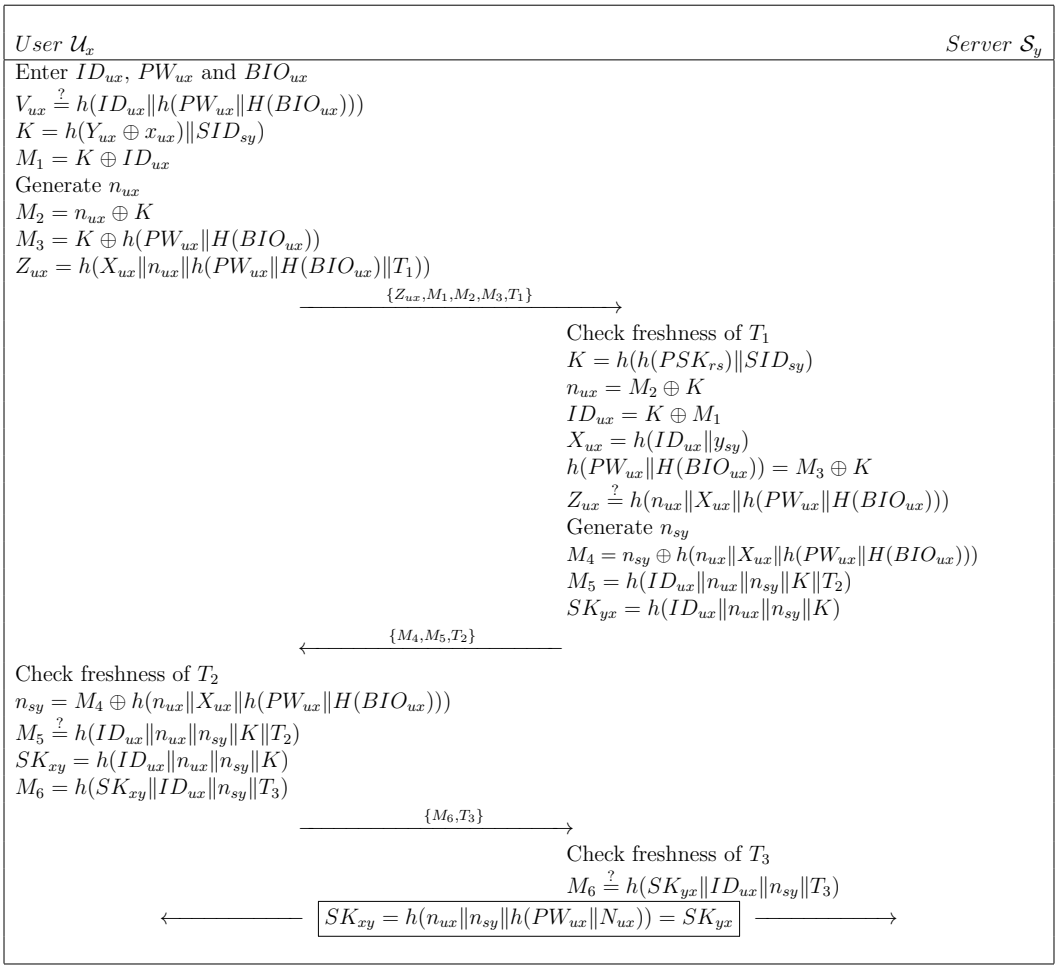


Figure 10.1: Lu et al.'s Scheme-1

Step L1A3: $\mathcal{S}_y$ upon receiving login message checks the freshness of T_1, aborts the session if T_1 is not fresh. Otherwise, computes $K = h(h(PSK_{rs})\|SID_{sy})$, $n_{ux} = M_2 \oplus K$, $ID_{ux} = K \oplus M_1$, $X_{ux} = h(ID_{ux}\|y_{sy})$ and $h(PW_{ux}\|H(BIO_{ux})) = M_3 \oplus K$.

Step L1A4: $\mathcal{S}_y$ verifies $Z_{ux} \stackrel{?}{=} h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, if it is not true $\mathcal{S}_y$ aborts the session. Otherwise, $\mathcal{S}_y$ selects a random number n_{sy} and computes $M_4 = n_{sy} \oplus h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, $M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$ and the session key $SK_{yx} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$. Further, $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$, where T_2 is current timestamp.

Step L1A5: Upon reception, $\mathcal{U}_x$ checks the freshness of T_2, if T_2 is fresh $\mathcal{U}_x$ computes $n_{sy} = M_4 \oplus h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and checks validity of $M_5 \stackrel{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$. If it is not valid $\mathcal{U}_x$ aborts the session. Otherwise, $\mathcal{U}_x$ computes the session key $SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$ and $M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3)$. Finally, $\mathcal{U}_x$ sends M_6, T_3 to $\mathcal{S}_y$, where T_3 is current timestamp.

Step L1A6: $\mathcal{S}_y$ upon receiving the message checks $M_6 \stackrel{?}{=} h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$ if it holds, $\mathcal{S}_y$ considers $\mathcal{U}_x$ as authenticated. The session key shared among both is:

$$SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K) \tag{10.1}$$

10.1.1.3 Password Change Phase

To change password, $\mathcal{U}_x$ enters his smart card in the reader, then inputs PW_{ux}, ID_{ux} and imprints BIO_{ux}. The smart card verifies $V_{ux} \stackrel{?}{=} h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, if it is true, $\mathcal{U}_x$ is asked to enter his new password PW_{ux}^{new}. Then the smart card computes $V_{ux}^{new} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and replaces V_{ux} by V_{ux}^{new}.

10.1.2 Review of Lu et al.'s Scheme-2

In this section, we briefly review Lu et al.'s biometrics based authentication scheme. Lu et al. employed public key techniques to achieve user anonymity and forward secrecy. Their scheme involves three participants: a user $\mathcal{U}_i$, a server $\mathcal{S}_y$ and the registration center RC. The scheme is illustrated in Fig. 10.2. We also elaborate Lu et al.'s scheme by following three phases:

10.1.2.1 Registration Phase

Initially, $\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux}, a random number N_{ux} along with his master private key x_{ux}. Then $\mathcal{U}_x$ scans his biometrics BIO_{ux}. Further, $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux}, N_{ux})\}$ to RC on a private channel. RC computes $R_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$ and personalizes the smart card SC_{ux} by $\{R_{ux}, h(PSK_{rs})\}$, where PSK_{rs} is the shared secret key between RC and $\mathcal{S}_y$. RC using private channel sends SC_{ux} to $\mathcal{U}_x$. Upon receiving smart card, $\mathcal{U}_x$ computes $X_{ux} = h(PSK_{rs}) \oplus x_{ux}$, $B_{ux} = N_{ux} \oplus H(BIO_{ux})$. Then $\mathcal{U}_x$ deletes $h(PSK_{rs})$ from the smart card (SC_{ux}), stores X_{ux} and B_{ux} in the smart card (SC_{ux}). Finally, the smart card (SC_{ux}) contains $\{R_{ux}, X_{ux}, B_{ux}, h()\}$.

10.1.2.2 Login and Authentication Phase

During login phase $\mathcal{U}_x$ inserts his SC_{ux} into card reader, imprints his biometrics (BIO_{ux}) and submits ID_{ux} and PW_{ux}. The steps performed by SC_{ux} and $\mathcal{S}_y$ are as follows:

Step L2A1: SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and $R'_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$.

Step L2A2: SC_{ux} verifies $R_{ux} \stackrel{?}{=} h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$, if not true, SC_{ux} aborts the session.

Step L2A3: SC_{ux} generates a random number n_{ux} and computes $M_1 = E_{Pub_{sy}}(ID_{ux}, n_{ux}, -h(PW_{ux}\|N_{ux}))$ and $M_2 = h((X_{ux} \oplus x_{ux})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$.

Step L2A4: Further, SC_{ux} sends login message $\{M_1, M_2\}$ to $\mathcal{S}_y$.

Step L2A5: For the received login message, $\mathcal{S}_y$ using his private key decrypts M_1 to get $(ID_{ux}, n_{ux}, h(PW_{ux}\|N_{ux}))$.

Step L2A6: $\mathcal{S}_y$ checks whether $M_2 \stackrel{?}{=} h(h(PSK_{rs})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$, if not true $\mathcal{S}_y$ aborts the session. Otherwise, $\mathcal{S}_y$ selects a random number n_{sy} and computes $M_3 = n_{sy} \oplus h(n_{ux}\|ID_{ux}\|h(PW_{ux}\|N_{ux}))$, the session key $SK_{yx} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$ and $M_4 = h(ID_{ux}\|n_{ux}\|SK_{yx}\|h(PW_{ux}\|N_{ux}))$. Further $\mathcal{S}_y$ sends $\{M_3, M_4\}$ to $\mathcal{U}_x$.

Step L2A7: For the received login message, $\mathcal{U}_x$ computes $n_{sy} = M_3 \oplus h(n_{ux}\|ID_{ux}\|h(PW_{ux}\|-N_{ux}))$ and session key $SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$. $\mathcal{U}_x$ then checks $M_4 \stackrel{?}{=} h(ID_{ux}\|-$
$n_{ux}\|SK_{xy}\|h(PW_{ux}\|N_{ux}))$. If it holds, $\mathcal{U}_x$ ponders $\mathcal{S}_y$ as authenticated.

Step L2A8: Finally, $\mathcal{U}_x$ computes and sends $M_5 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$ to $\mathcal{S}_j$.

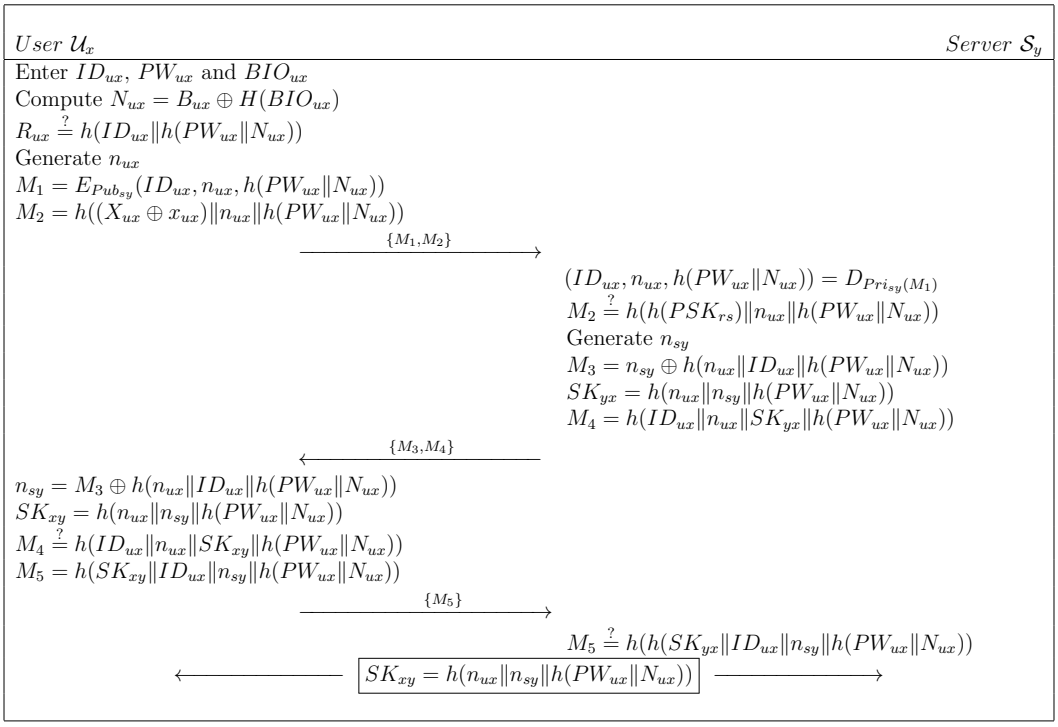


Figure 10.2: Lu et al.'s Scheme-2

Step L2A9: $\mathcal{S}_y$ checks $M_5 \stackrel{?}{=} h(h(SK_{yx}\|ID_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$, if it holds, $\mathcal{U}_x$ ponders $\mathcal{S}$ as authenticated.

The computed shared key between $\mathcal{U}_x$ and $\mathcal{S}_y$ is:

$$SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux})) \tag{10.2}$$

10.1.2.3 Password Change Phase

$\mathcal{U}_x$ inserts his smart card (SC_{ux}) in specialized reader. $\mathcal{U}_x$ then inputs ID_{ux}, PW_{ux} and BIO_{ux}. SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and checks $R_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$, if it holds SC_{ux} asks for new password. $\mathcal{U}_x$ inputs new password PW_{ux}^{new}. SC_{ux} computes $R_{ux}^{new} = h(ID_{ux}\|h(PW_{ux}^{new}\|N_{ux}))$. Finally, SC_{ux} replaces R_{ux} by R_{ux}^{new}.

10.2 Cryptanalysis of Lu et al.'s Schemes

This section performs cryptanalysis of Lu et al.'s schemes. We show that Lu et al's scheme-1 is vulnerable to: (1) user anonymity violation attack; (2) user impersonation attack and

having correctness problems. Likewise, we show that Lu et al.'s scheme-2 is vulnerable to user impersonation attacks.

10.2.1 Weaknesses of Lu et al.'s scheme-1

Following subsections show that Lu et al.'s scheme-1 is vulnerable to user anonymity violation and impersonation attacks:

10.2.1.1 User Anonymity Violation Attack

Here, we prove that Lu et al.'s scheme-1 is vulnerable to user anonymity violation attack. For successful user impersonation attack, initially an attacker $\mathcal{A}$ selects his identity ID_{ua}, password PW_{ua}, biometrics BIO_{ua} and his own secret key x_{ua}. Then $\mathcal{A}$ registers to the system and obtains a smart card containing $X_{ua} = h(ID_{ua}\|y_{rc})$, $V_{ua} = h(ID_{ua}\|h(PW_{ua}\|H(BIO_{ua})))$ and $Y_{ua} = h(PSK_{rs}) \oplus x_{ua}$. $\mathcal{A}$ performs following steps for successful anonymity violation attack:

Step UAV1: $\mathcal{A}$ extracts $h(PSK_{rs})$ as follows:

$$h(PSK_{rs}) = x_{ua} \oplus Y_{ua} \tag{10.3}$$

Step UAV2: When $\mathcal{U}_x$ initiates the authentication requests by sending $Z_{ux}, M_1, M_2, M_3, T_1$ to $\mathcal{S}_y$. $\mathcal{A}$ intercepts the message and computes:

$$K = h(h(PSK_{rs}\|SID_{xy})) \tag{10.4}$$

$$n_{ux} = M_2 \oplus K \tag{10.5}$$

$$ID_{ux} = K \oplus M_1 \tag{10.6}$$

In eq. 10.6, ID_{ux} is the real identity of user $\mathcal{U}_x$. Hence, $\mathcal{A}$ has successfully break the anonymity of $\mathcal{U}_x$.

10.2.1.2 User Impersonation

Here, we prove that Lu et al.'s scheme-1 is vulnerable to impersonation attack. We show that an adversary $\mathcal{A}$ can impersonate any other registered user of the system if he becomes able

to steal his smart card. Initially, $\mathcal{A}$ extracts $X_{ux} = h(ID_{ux}\|y_{rc})$ out of a stolen smart card. Then he performs following steps to impersonate himself as $\mathcal{U}_x$:

Step IA1: $\mathcal{A}$ computes:

$$K = h(h(PSK_{rs}\|SID_{xy})) \tag{10.7}$$
$$M_1 = K \oplus ID_{ux} \tag{10.8}$$

Step IA2: $\mathcal{A}$ generates two random numbers n_{ua} and P_{ua}. Then generates timestamp T_1 and computes:

$$M_2 = n_{ua} \oplus K \tag{10.9}$$
$$M_3 = K \oplus P_{ua} \tag{10.10}$$
$$Z_{ua} = h(X_{ux}\|n_{ua}\|P_{ua}\|T_1) \tag{10.11}$$

Step IA3: $\mathcal{A}$ sends $Z_{ua}, M_1, M_2, M_3, T_1$ to $\mathcal{S}_y$.

Step IA4: $\mathcal{S}_y$ upon receiving login message, checks the freshness of T_1, as T_1 is freshly generated so $\mathcal{S}_y$ computes:

$$K = h(h(PSK_{rs})\|SID_{sy}) \tag{10.12}$$
$$n_{ua} = M_2 \oplus K \tag{10.13}$$
$$ID_{ux} = K \oplus M_1 \tag{10.14}$$
$$X_{ux} = h(ID_{ux}\|y_{sy}) \tag{10.15}$$
$$P_{ua} = M_3 \oplus K \tag{10.16}$$

Step IA5: $\mathcal{S}_y$ verifies $Z_{ux} \stackrel{?}{=} h(n_{ua}\|X_{ux}\|P_{ua})$ and finds it true. $\mathcal{S}_y$ then selects a random number n_{sy} and computes:

$$M_4 = n_{sy} \oplus h(n_{ua}\|X_{ux}\|P_{ua})) \tag{10.17}$$
$$M_5 = h(ID_{ux}\|n_{ua}\|n_{sy}\|K\|T_2) \tag{10.18}$$
$$SK_{yx} = h(n_{ua}\|n_{sy}\|P_{ua})) \tag{10.19}$$

Step IA6: Further, $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$, where T_2 is current timestamp.

Step IA7: Upon reception $\mathcal{A}$ computes:

$$n_{sy} = M_4 \oplus h(n_{ua} \| X_{ux} \| Pua) \tag{10.20}$$

$$SK_{xy} = h(ID_{ux} \| n_{ua} \| n_{sy} \| K) \tag{10.21}$$

$$M_6 = h(SK_{xy} \| ID_{ux} \| n_{sy} \| T_3) \tag{10.22}$$

Step IA8: Finally, $\mathcal{A}$ sends $M_6,\ T_3$ to $\mathcal{S}_y$, where T_3 is current timestamp. $\mathcal{S}_y$ upon receiving the message checks $M_6 \stackrel{?}{=} h(SK_{yx} \| ID_{ux} \| n_{sy} \| T_3)$ and finds it true.

Hence, $\mathcal{A}$ has successfully deceives $\mathcal{S}_y$ by impersonating himself as $\mathcal{U}_x$. The session key shared among both is:

$$SK_{xy} = h(ID_{ux} \| n_{ua} \| n_{sy} \| K) \tag{10.23}$$

10.2.2 Weaknesses of Lu et al.'s Scheme-2

This section elaborates the weaknesses of Lu et al.'s scheme against user impersonation attack. We show that a dishonest legal user $\mathcal{A}$ can easily masquerade himself as an other honest user $\mathcal{U}_x$, considering the common adversarial model as mentioned in subsection 2.2.6.

10.2.2.1 User Impersonation Attack

Here, we show that Lu et al.'s scheme cannot resist a forgery attack by a legal user to impersonate himself as another user of the system. Let $\mathcal{A}$ be a legal user having smart card SC_{ua} and wants to impersonate himself as another user $\mathcal{U}_x$. Following steps will be performed by $\mathcal{A}$ for a successful forgery attack to $\mathcal{S}_y$.

Step IA 1: $\mathcal{A}$ extracts the information stored in SC_{ua} and computes:

$$h(PSK_{rs}) = X_{ua} \oplus x_{ua} \tag{10.24}$$

Step IA 2: $\mathcal{A}$ generates two random numbers n_{ua} and P_{ua} and computes:

$$M_{\bar{1}} = E_{Pub_{sy}}(ID_{ux}, n_{ua}, P_{ua}) \tag{10.25}$$

$$M_{\bar{2}} = h((X_{ua} \oplus x_{ua}) \| n_{ua} \| P_{ua}) \tag{10.26}$$

Step IA 3: $\mathcal{A}$ sends $M_{\bar{1}}$ and $M_{\bar{2}}$ as login message to $\mathcal{S}_j$.

Step IA 4: For the received login message, $\mathcal{S}_y$ decrypts $M_{\bar{1}}$ to obtain:

$$(ID_{ux}, n_{ua}, P_{ua}) = D_{Pri_{sy}}(M_{\bar{1}}) \tag{10.27}$$

Step IA 5: $\mathcal{S}_y$ further verifies $M_{\bar{2}} \stackrel{?}{=} h(h(PSK_{rs})\|n_{ua}\|P_{ua})$ and finds it to be true.

Step IA 6: $\mathcal{S}_y$ further selects n_{sy} and computes:

$$M_3 = n_{sy} \oplus h(n_{ua}\|ID_{ux}\|P_{ua}) \tag{10.28}$$
$$SK_{yx} = h(n_{ux}\|n_{sy}\|P_{ua}) \tag{10.29}$$
$$M_4 = h(ID_{ux}\|n_{ua}\|SK_{yx}\|P_{ua}) \tag{10.30}$$

Step IA 7: $\mathcal{S}_y$ sends M_3 and M_4 to $\mathcal{U}_x$ as response message.

Step IA 8: $\mathcal{A}$ intercepts the message and computes:

$$n_{sy} = M_3 \oplus h(n_{ua}\|ID_{ux}\|P_{ua}) \tag{10.31}$$
$$SK_{xy} = h(n_{ua}\|n_{sy}\|P_{ua}) \tag{10.32}$$
$$M_5 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|P_{ua}) \tag{10.33}$$

Step IA 9: $\mathcal{A}$ sends M_5 to $\mathcal{S}_y$.

Step IA 10: $\mathcal{S}_y$ checks $M_5 \stackrel{?}{=} h(h(SK_{yx}\|ID_{ux}\|n_{sy}\|P_{ua})$ and finds it to be true.

Hence, $\mathcal{A}$ successfully deceived $\mathcal{S}_y$ by impersonating himself as $\mathcal{U}_x$. The shared key between $\mathcal{A}$ and $\mathcal{S}_y$ is:

$$SK_{yx} = h(n_{ua}\|n_{sy}\|P_{ua}) \tag{10.34}$$

10.3 Proposed Scheme

In this section, we propose an improved and secure biometrics based three factor authentication scheme to overcome the weaknesses of Lu et al.'s schemes. The proposed scheme is depicted in figure 10.3 and is explained in following four subsections:

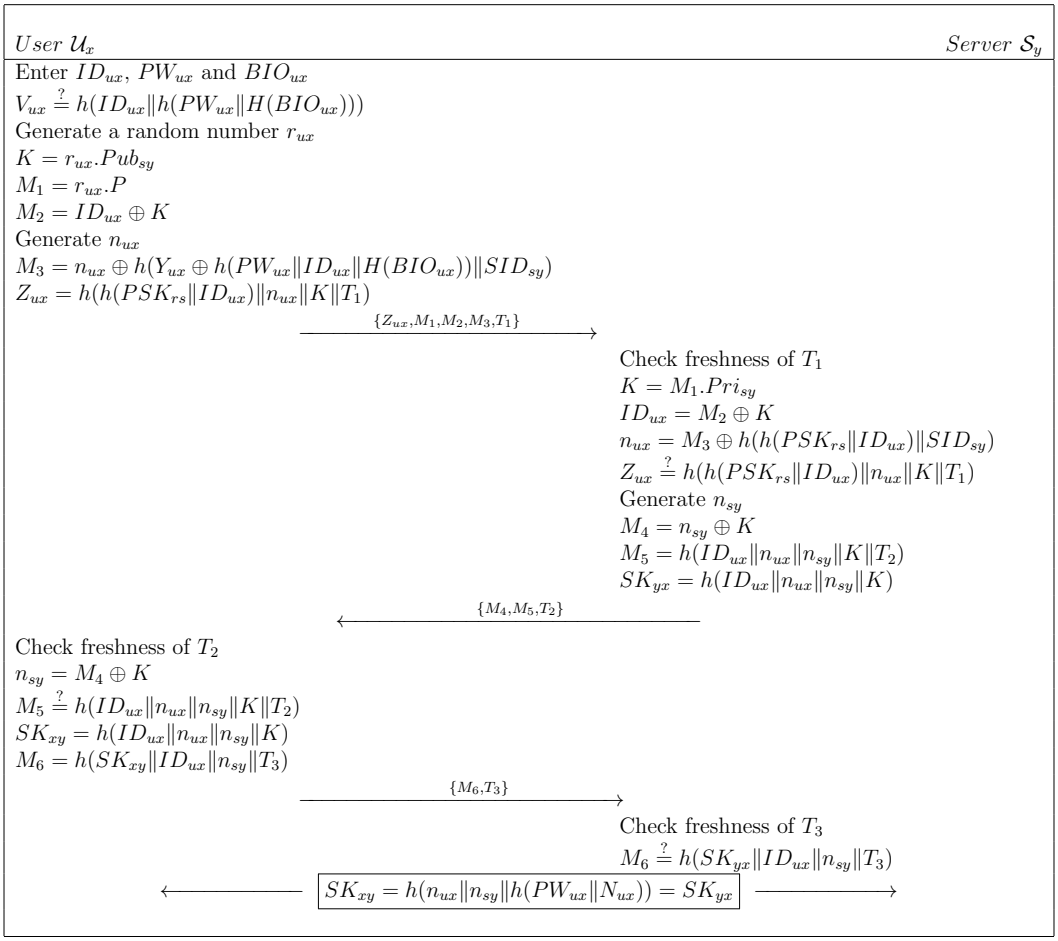


Figure 10.3: Proposed Scheme

10.3.1 Initialization

In this phase, system parameters are selected by registration server. Initially, registration server RC selects an elliptic curve $E_p(a,b) \mod p$, a base point P over $E_p(a,b)$, a one way hash function $h(.)$, biometrics hashing $H(.)$ and a shared key with all servers PSK_{rs}. Finally, RC publishes system public parameters $E_p(a,b), h(.), H(.)$.

10.3.2 Registration Phase

In this phase, both the users and servers registers with the registration server. Following two subsections describes the process of registration:

10.3.2.1 Server Registration

To register with the system, a server $\mathcal{S}_y$ selects his identity SID_{sy} and his private key Pri_{sy}. Then $\mathcal{S}_y$ computes his public key $Pub_{sy} = Pri_{sy}.P$ and sends his identity SID_{sy} and his public key Pub_{sy} to RC. Upon reception, RC shares the secret key PSK_{rs} with $\mathcal{S}_y$ and publishes $\mathcal{S}_y$'s public key Pub_{sy}.

10.3.2.2 User Registration

User registration involves following three steps:

Step PR 1: $\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux} and scans his biometrics BIO_{ux}. Further, $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux} \| H(BIO_{ux}))\}$ to RC on a private channel.

Step PR 2: Upon reception of request, RC computes $V_{ux} = h(ID_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$, $h(PSK_{rs} \| ID_{ux})$. Then stores $h(PSK_{rs} \| ID_{ux})$ and V_{ux} in the smart card SC_{ux}. RC sends smart card (SC_{ux}) to $\mathcal{U}_x$.

Step PR 3: For the received $SC_{ux}, \mathcal{U}_x$ computes $Y_{ux} = h(PSK_{rs} \| ID_{ux}) \oplus h(PW_{ux} \| ID_{ux} \| H(BIO_{ux}))$. Finally, smart card contains $Y_{ux}, V_{ux}, h(.)$.

10.3.3 Login and Authentication Phase

Login phase starts when any user $\mathcal{U}_x$ enters his SC_{ux} into card reader, embosses his biometrics (BIO_{ux}) and enters ID_{ux} and PW_{ux}. The subsequent steps accomplished by SC_{ux} and $\mathcal{S}_y$

are as under:

Step LA1: SC_{ux} calculates $h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and confirms $V_{ux} \stackrel{?}{=} h(ID_{ux}\|h(-PW_{ux}\|H(BIO_{ux})))$, if condition does not hold, SC_{ux} terminates the session.

Step LA2: SC_{ux} produces a random number r_{ux} and calculates $K = r_{ux}.Pub_{sy}$, $M_1 = r_{ux}.P$ and $M_2 = ID_{ux} \oplus K$.

Step LA3: Moreover, SC_{ux} produces another random number n_{ux} and calculates $M_3 = n_{ux} \oplus h(Y_{ux} \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))\|SID_{sy})$ and $Z_{ux} = h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$

Step LA4: Thereafter, SC_{ux} transmits login message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$ to $\mathcal{S}_y$.

Step LA5: On getting login message, $\mathcal{S}_y$ verifies freshness of T_1.

Step LA6: $\mathcal{S}_y$ calculates $K = M_1.Pri_{sy}$ with his private key and also calculates $ID_{ux} = M_2 \oplus K$ and $n_{ux} = M_3 \oplus h(h(PSK_{rs}\|ID_{ux})\|SID_{sy})$.

Step LA7: $\mathcal{S}_y$ verifies $Z_{ux} \stackrel{?}{=} h(h(PSK_{rs}\|ID_{us})\|n_{ux}\|K\|T_1)$, if does not hold, $\mathcal{S}_y$ terminates the session. Otherwise, $\mathcal{S}_y$ generates a random number n_{sy} and calculates $M_4 = n_{sy} \oplus K$, $M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$ and the session key $SK_{yx} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$. Further, $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$.

Step LA8: On receiving login message, $\mathcal{U}_x$ verifies freshness of T_2 and computes $n_{sy} = M_4 \oplus K$. Then confirms $M_5 \stackrel{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$, if it holds, $\mathcal{U}_x$ cogitates $\mathcal{S}_y$ as authenticated. Then session key is computed as $SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$.

Step LA9: After that, $\mathcal{U}_x$ calculates $M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3)$ and and transmits $\{M_6, T_3\}$ to $\mathcal{S}_y$.

Step LA10: $\mathcal{S}_y$ checks the freshness of T_3 and also verifies $M_6 \stackrel{?}{=} h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$ if it holds, $\mathcal{U}_x$ cogitates $\mathcal{S}_y$ as authenticated.

The derived shared key between $\mathcal{U}_x$ and $\mathcal{S}_y$ is:

$$SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux})) = SK_{yx} \tag{10.35}$$

10.3.4 Password Change Phase

$\mathcal{U}_x$ inserts his smart card (SC_{ux}) in specialized reader. $\mathcal{U}_x$ then inputs ID_{ux}, PW_{ux} and BIO_{ux}. SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and checks $R_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$, if it holds SC_{ux} asks for new password. $\mathcal{U}_x$ inputs new password PW_{ux}^{new}. SC_{ux} computes $R_{ux}^{new} =$

$h(ID_{ux}\|h(PW_{ux}^{new}\|N_{ux}))$ and $X_{ux}^{new} = X_{ux} \oplus h(PW_{ux}\|ID_{ux}\|N_{ux}) \oplus h(PW_{ux}^{new}\|ID_{ux}\|N_{ux}^{new})$ Finally, SC_{ux} replaces R_{ux} and X_{ux} by R_{ux}^{new} and X_{ux}^{new}.

10.4 Security Analysis

The formal security analysis followed by security discussion is performed in this section. Further, protocol verification through automated tool ProVerif is also substantiated here.

10.4.1 Formal Security

To demonstrate that proposed scheme is provably secure, we adopted the same analysis as mentioned in [8, 94]. Following oracles are defined for analysis purpose:

- ***Reveal:*** This oracle unconditionally outputs a string S from the one way hash function $R = h(S)$.
- ***Extract:*** This oracle unconditionally outputs the scalar multiplier k out of a given elliptic curve points $O = kP$ and P.

Theorem 8. *The proposed biometrics based multi-server authentication scheme is provably secure for an attacker $\mathcal{A}$ to stanch $\mathcal{U}_x$'s identity (ID_{ux}), the parameter K, the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$ considering one way hash function as random oracle and under the hardness assumption of $ECDLP$.*

Proof. Let $\mathcal{A}$ be an adversary having capabilities to compute $\mathcal{U}_x$'s ID_{ux}, the secret session parameter K, the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$. $\mathcal{A}$ simulates both oracles *Reveal* and *Extract* to run the algorithmic experiment $EXPE1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}$ against our proposed three factor biometrics based authentication scheme for multi-server environments ($TFBAMS$). The success probability for the mentioned experiment is defined as $Succe_1 = |Prb[EXPE1_{\mathcal{A},TFBAMS}^{HASH,ECDLP} = 1] - 1|$. $\mathcal{A}$'s advantage is solicited as $Advt1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}(t, q_{rev}, q_{ext}) = max_{\mathcal{A}}(Succe_1)$, where $\mathcal{A}$ is allowed to make at maximum q_{rev} *Reveal* and q_{ext} *Extract* queries. Referring to the experiment $\mathcal{A}$ can compute ID_{ux}, K, SK_{xy} and PSK_{rs}, if he can (i) invert the hash function and (ii) solve the ECDLP. However, referring to Definition 1, it is computationally infeasible to invert a secure one way hash function, similarly by Definition 2, it is computationally infeasible to solve ECDLP. Hence, we have $Advt1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}(t, q_{rev}, q_{ext}) \leq \epsilon$. Therefore, proposed three factor biometrics based authentication scheme for multi-server environments is secure against an adversary $\mathcal{A}$ to

computes $\mathcal{U}_x$'s ID_{ux}, the secret session parameter K, the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$. □

Algorithm 3 $EXPE^{HASH,ECDLP}_{\mathcal{A},TFBAMS}$

Eavesdrop the login message $Z_{ux}, M_1, M_2, M_3, T_1$, Where $M_1 = r_{ux}.P$, $M_2 = ID_{ux} \oplus K$, $M_3 = n_{ux} \oplus h(h(PSK_{rs}\|ID_ux)\|SID_{sy})$ and $Z_{ux} = h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$
Call Extract oracle on M_1 and P to obtain $r'_{ux} \leftarrow Extract(M_1, P)$
Compute $K' = r_{ux} \oplus Pub_{sy}$ and $ID'_{ux} = K' \oplus M_2$
Call Reveal on Z_{ux} to get $h(PSK_{rs}\|ID_{ux})'\|n'_{ux}\|K''\|T'_1) \leftarrow Reveal(Z_{ux})$
if $(K'' = K')$ **then**
    Call Reveal on $h(PSK_{rs}\|ID_{ux})'$ and get $(PSK'_{rs}\|ID''_{ux}) \leftarrow Reveal(h(PSK_{rs}\|ID_{ux})')$
    if $(ID'_{ux} = ID''_{ux})$ **then**
        Accept ID'_{ux} and PSK'_{rs} along with session specific parameters n'_{ux} and K'
        Eavesdrop challenge message M_4, M_5, T_2, where $M_4 = n_{sy} \oplus K$ and $M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$
        Compute $n'_{sy} = M'_4 oplus K'$ and $SK'_{xy} = h(ID'_{ux}\|n'_{ux}\|n'_{sy}\|K)$
        Eavesdrop response message M_6, T_3
        Compute $M'_6 = h(SK'_{xy}\|ID'_{ux}\|n'_{sy}\|T_3)$
        if $(M'_6 = M_6)$ **then**
            Accept SK'_{xy}
        else
            return Fail
        end if
    else
        return Fail
    end if
else
    return Fail
end if

Theorem 9. *The proposed biometrics based multi-server authentication scheme is provably secure for an attacker $\mathcal{A}$ to stanch $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ considering one way hash function as random oracle for the stolen smart card attack.*

Proof. Let $\mathcal{A}$ be an adversary having capabilities to stanch $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ out of a stolen smart card. $\mathcal{A}$ simulates *Reveal* oracle to run the algorithmic experiment $EXPE2^{HASH}_{\mathcal{A},TFBAMS}$ against our proposed three factor biometrics based authentication scheme for multi-server environments $(TFBAMS)$. The success probability for the mentioned experiment is defined as $Succe_2 = |Prb[EXPE2^{HASH}_{\mathcal{A},TFBAMS} = 1] - 1|$. $\mathcal{A}$'s advantage is solicited as $Advt2^{HASH}_{\mathcal{A},TFBAMS}(t, q_{rev} = max_{\mathcal{A}}(Succe_2)$, where $\mathcal{A}$ is allowed to make at maximum q_{rev} *Reveal* queries. Referring to the experiment $\mathcal{A}$ can compute $H(BIO_{ux})$, ID_{ux}, PW_{ux} and PSK_{rs}, if he can invert the hash function. However, referring to Definition 1, it is computationally infeasible to invert a secure one way hash function. Hence, we have $Advt2^{HASH}_{\mathcal{A},TFBAMS}(t, q_{rev}) \leq \epsilon$. Therefore, proposed three factor biometrics based authentication scheme for multi-server environments is secure against an adversary $\mathcal{A}$ to computes $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ out of a stolen smart card. □

Algorithm 4 $EXPE^{HASH}_{\mathcal{A},TFBAMS}$

Extract the parameters Y_{ux}, V_{ux} from stolen smart card using the methods mentioned in [28, 29] Where $Y_{ux} = h(PSK_{rs}\|ID_{ux}) \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))$ and $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
Call Reveal oracle on V_{ux} and obtain $(ID'_{ux}\|h(PW_{ux}\|H(BIO_{ux}))') \leftarrow Reveal(V_{ux})$
Call Reveal on $h(PW_{ux}\|H(BIO_{ux}))'$ to get $(PW'_{ux}\|H(BIO_{ux})') \leftarrow Reveal(h(PW_{ux}\|H(BIO_{ux}))')$
Compute $W = h(PW'_{ux}\|ID'_{ux}\|H(BIO_{ux})')$ and $T = Y_{ux} \oplus W = \overline{h(PSK_{rs}\|ID_{ux})}$
Call Reveal on T and obtain $(PSK'_{rs}\|ID''_{ux}) \leftarrow Reveal(T)$
if $(ID''_{ux} = ID'_{ux})$ **then**
    Accept PSK_{rs}, PW'_{ux} and $H(BIO_{ux})'$
else
    return Fail
end if

Table 10.2: Comparison of Security Parameters

Scheme:	Proposed	[10]	[11]	[94]	[50]
Anonymity and privacy	Yes	Yes	No	Yes	Yes
Mutual authentication and key agreement	Yes	Yes	Yes	Yes	Yes
Resists impersonation attack	Yes	No	No	No	No
Resists smart card theft attack	Yes	Yes	Yes	Yes	No
Resists replay attack	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	No	Yes
Resists insider and stolen verifier attacks	Yes	Yes	Yes	Yes	Yes
Resists password guessing attack	Yes	Yes	Yes	Yes	Yes
No clock synchronization	Yes	Yes	Yes	Yes	Yes

10.4.2 Further Security Discussion

In this subsection, we informally describes the security functionalities provided by the proposed scheme.

10.4.2.1 Anonymity and Privacy

In our proposed biometrics based scheme the user $\mathcal{U}_x$'s identity ID_{ux} is not sent over public network rather M_1 and M_2 are sent to $\mathcal{S}_y$. These two parameters are freshly generated for each session. The anonymity can only be broken if an adversary can compute K, but it can be seen that K can be computed only by the use of $\mathcal{S}_y$'s private key. Hence, proposed scheme preserves anonymity and untraceability.

10.4.2.2 Mutual Authentication

$\mathcal{S}_y$ authenticates $\mathcal{U}_x$ by checking $Z_{ux} \stackrel{?}{=} h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$. Computation of Z_{ux} involves $h(PSK_{rs}\|ID_{ux})$ which requires the smart card as well as password PW_{ux} and the biometrics BIO_{ux} of $\mathcal{U}_x$. Therefore, to deceive $\mathcal{S}_y$, the adversary needs $\mathcal{U}_x$'s password, biometrics as well as his smart card. Likewise, $\mathcal{U}_x$ authenticates $\mathcal{S}_y$ by checking

$M_5 \stackrel{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$, which requires the computation of $\mathcal{U}_x$'s identity ID_{ux}, the session parameter n_{ux} and K. ID_{ux} and K can be computed only by using $\mathcal{S}_y$'s private key as mentioned in subsection 10.4.2.1, while n_{ux} can be computed by using $h(h(PSK_{rs}\|ID_{ux})\|SID_{sy})$, which requires the shared secret key between $\mathcal{S}_y$ and RC. So, in order to deceive $\mathcal{U}_x$, the adversary needs $\mathcal{S}_y$'s private key Pri_{sy} as well as the shared key $h(PSK_{rs})$ between $\mathcal{S}_y$ and RC. Hence, only legal user can pass authentication test from server and vice versa. Therefore, proposed scheme provides proper mutual authentication.

10.4.2.3 User and Server Impersonation Attacks

Only legal user can generate legal authentication request message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$ and response message $\{M_6, T_3\}$, similarly only legal server can respond with challenge message $\{M_4, M_5, T_2\}$ as proved in subsection 10.4.2.2.

10.4.2.4 Smart Card Theft/Stolen Attack

Let us assume, the adversary by using some means is able to acquire $\mathcal{U}_x$'s smart card. The adversary further extracts the parameters $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, $Y_{ux} = h(PSK_{rs}\|ID_{ux}) \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))$ and $h(.)$. Then to compute the secret parameter $h(PSK_{rs}\|ID_{ux})$, the adversary needs PW_{ux} and BIO_{ux}. Hence, the stolen smart card will not benefit the adversary for forgery.

10.4.2.5 Replay Attack

If some adversary after intercepting the login request message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$, replays it later on. The server $\mathcal{S}_y$ after receiving the message will check the freshness of timestamp T_1, as the timestamp is old dated, $\mathcal{S}_y$ will simply discard the message. Therefore, replay attack is not viable on the proposed scheme.

10.4.2.6 Perfect Forward Secrecy

The computed session key between $\mathcal{S}_y$ and $\mathcal{U}_x$ contains share (n_{sy}, n_{ux}) from both the participants, respectively. So,S even if the long term private key of $\mathcal{S}_y$ or $\mathcal{U}_x$'s password is revealed to the attacker it will not benefit to compute previous session keys. Therefore, the proposed scheme possesses perfect forward secrecy.

10.4.2.7 Insider and Stolen Verifier Attacks

For the proposed scheme, $\mathcal{S}_y$ does not store any parameter related to $\mathcal{U}_x$'s password (PW_{ux}) or his biometrics (BIO_{ux}). As there is no verifier table, so no stolen verifier attack is possible. Likewise, $\mathcal{U}_x$ does not send his password (PW_{ux}) or his biometrics BIO_{ux} in plain text. Hence, no insider will have any advantage to expose his password or biometrics.

10.4.2.8 Password Guessing Attack

For the proposed scheme, the information relating to $\mathcal{U}_x$'s password is protected by his identity ID_{ux}, biohashed biometrics $H(BIO_{ux})$. Further, it is enclosed by exclusive OR with $h(PSK_{rs}\|ID_{ux})$. Moreover, there is no parameter stored in smart card to check the validity of guessed password by adversary. Hence, no offline password guessing attack is feasible on proposed scheme. Likewise, the system incorporates built in maximum number of login requests, which ensures no online password guessing attack. In proposed scheme, the information relating to $\mathcal{U}_i$'s password is protected by N_{ux} and oneway hash function. Further, there is no parameter to verify correctness of user's password. Hence, password guessing attack is not feasible on proposed scheme.

10.5 Verification through ProVerif

To demonstrate the security of proposed scheme, we have implemented the login and authentication steps of the protocol as illustrated in Fig. 10.3 and explained in subsection 10.3.3. We have shown declaration part in Fig. 10.4(a). Process part is illustrated in Fig. 10.4(b). We have defined two processes: server process *(ServerSy)* and user process *(UserUx)*. Main part is shown in Fig. 10.4(c). The results are as follows:

1. RESULT inj-event(end˙Serversy(id)) ==> inj-event(begin˙Serversy(id)) is true.
2. RESULT inj-event(end˙Userux(id˙1114)) ==> inj-event(begin˙Userux(id˙1114)) is true.
3. RESULT not attacker(SKxy[]) is true.

The results (1) and (2) validates that both user and server processes started and terminated normally, which confirms the correctness and reachability properties. While (3) verifies that the session key *(SKxy[])* is not exposed to adversary. Hence, the proposed protocol possesses reachability as well as secrecy and authentication properties.

```
(************** Channels *************)
free Ch_Pub:channel.
(********* Names & Variables *********)
free IDux:bitstring.
free PWux:bitstring.
free Yux:bitstring.
free BIOux:bitstring [private].
free Vux:bitstring [private].
const P: bitstring.
free Pubsy:bitstring.
free PSKrs:bitstring [private].
free SIDsy:bitstring [private].
(**Constructors*destructors*Equations **)
fun h(bitstring):bitstring.
fun H(bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun concat(bitstring,bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
equation forall a:bitstring,b:bitstring;
  xor(xor(a,b),b)=a.
```

(a) Declarations

```
(***************Events ***************)
event begin_Userux(bitstring).
event end_Userux(bitstring).
event begin_Serversy(bitstring).
event end_Serversy(bitstring).
(********Process Replication***********)
process (    (!ServerSy) |   (!UserUx) )
(************* *queries* *************)
free SKxy:bitstring [private].
query attacker(SKxy).
query id:bitstring; inj event(end_Userux(id
    )) ==> inj event(begin_Userux(id)) .
query id:bitstring; inj event(end_Serversy(
    id)) ==> inj event(begin_Serversy(id))
    .
```

(c) Main

```
(**************processes *************)
(***************User ux****************)
let UserUx =
(*Login and Authentication Phase*)
let Vux'=h(concat(IDux,h(concat(PWux,H(
    BIOux))))) in
new rux:bitstring;
let K=mult(rux,Pubsy) in
let M1=mult(rux,P) in
let M2=xor(IDux,K) in
new nux:bitstring;
new T1:bitstring;
let M3=xor(nux,h(xor(Yux , concat(h(concat(
    PWux,(IDux,H(BIOux)))),SIDsy)))) in
let Zux=h(concat(h(concat(PSKrs,IDux)),(nux
    ,K,T1))) in
out(Ch_Pub,(Zux,M1,M2,M3,T1));
in(Ch_Pub,(xM4:bitstring,xM5:bitstring,xT2:
    bitstring));
new T2:bitstring;
let nsy=xor(xM4,K) in
let M5=h(concat(IDux,(nux,nsy,K,T2))) in
if(M5=xM5) then
let SKxy=h(concat(IDux,(nux,nsy,K))) in
new T3:bitstring;
let M6=h(concat(SKxy,(IDux,nsy,T3))) in
out(Ch_Pub,(M6,T3))
else 0.
(************** Server Sy***************)
let ServerSy=
new Prisy:bitstring;
let Pubsy=mult(Prisy,P) in
in(Ch_Pub,(xZux:bitstring,xM1:bitstring,xM2
    :bitstring,xM3:bitstring,xT1:bitstring)
    );
new T1:bitstring;
let K=mult(xM1,Prisy) in
let IDux'=xor(xM2,K) in
let nux=xor(xM3,h(concat(h(concat(PSKrs,
    IDux')),SIDsy))) in
let Zux=h(concat(h(concat(PSKrs,IDux')),(
    nux,K,T1))) in
if (Zux=xZux) then
new nsy:bitstring;
new T2:bitstring;
let M4=xor(nsy,K) in
let M5=h(concat(IDux',(nux,nsy,K,T2))) in
let SKxy=h(concat(IDux',(nux,nsy,K))) in
out(Ch_Pub,(M4,M5,T2));
in(Ch_Pub,(xM6:bitstring,xT3:bitstring));
new T3:bitstring;
let M6=h(concat(SKxy,(IDux',nsy,T3))) in
if(M6=xM6) then 0.
```

(b) Processes

Figure 10.4: ProVerif Validation

10.6 Performance Comparisons

This section presents performance assessment of the proposed scheme against two Lu et al.'s pertinent schemes and two other related schemes. Following notations are used as per Kilinc and Yanik [69] experiments:

- T_{Oh} refers to accumulated execution time of one-way hash operation, that consumes $0.0023ms$.
- T_{Re} refers to accumulated execution time of RSA encryption, that consumes $3.8500ms$.
- T_{Rd} refers to accumulated execution time of RSA decryption, that consumes $0.1925ms$.
- T_{Epm} refers to elliptic curve point multiplication and it takes $2.229ms$.

Table 10.3: Computation Cost Comparison

Scheme	User Side	Server Side	Total Execution time
Chuang et al. [50]	$8T_{Oh}$	$8T_{Oh}$	$16T_{Oh} \approx 0.0368$
Mishra et al. [94]	$10T_{Oh}$	$7T_{Oh}$	$17T_{Oh} \approx 0.0391$
Lu et al. [11]	$9T_{Oh}$	$8T_{Oh}$	$17T_{Oh} \approx 0.0391ms$
Lu et al. [10]	$8T_{Oh} + 3T_{Re}$	$8T_{Oh} + 3T_{Rd}$	$16T_{Oh} + 3T_{Re} + 3T_{Rd} \approx 12.1643ms$
Proposed Scheme	$9T_{Oh} + 2T_{Epm}$	$7T_{Oh} + 1T_{Epm}$	$16T_{Oh} + 3T_{Epm} \approx 6.7148ms$

The comparison presented in Table 10.3 reveals that the proposed scheme is computationally inexpensive than both Lu et al.'s schemes. Moreover, proposed scheme provides invincibility against the known threats. Therefore, it can be declared that the proposed scheme is not only robust and efficient against known attacks but it is also lightweight in terms of its computation cost.

10.7 Chapter Summary

In this chapter, we have cryptanalyzed two most recent biometrics based multi-factor authentication schemes proposed by Lu et al. We have proved both of their schemes to be vulnerable to impersonation attacks, additionally we have also showed that one of their scheme is also vulnerable to anonymity violation attack. Then we proposed an improved biometrics based multi-factor authentication scheme. The proposed scheme is proved to be robust against all known attacks. We have substantiated the security of proposed scheme using famous automated security validation tool ProVerif.

Chapter 11

An ID-based multi-server Authentication Scheme for Mobile Cloud Computing using Bilinear Mapping

Mobile cloud computing (MCC) refers to corporal structure where computation, manipulation and storage of data and information takes place, away from mobile devices. This corporal structure or infrastructure itself is designated as cloud [173]. MCC is emerging as an eminent facility for mobile world to experience efficient and cost effective utilization of the remote resources for computation as well as data storage. Although MCC is proved to be useful and got huge publicity but its utilization trend is below its expected potential because ABI research observed that only 19 % of the total mobile users has subscribed the MCC services in 2014. International Data Corporation has also revealed the reason about less utilization of MCC is due to the fact that top level management in most of the organizations has avoided to take up the MCC services due to security and privacy concerns [174, 175].

Since MCC facilitates mobile users to access remote resources with the help of their mobile devices over wireless communication medium such as WLAN or 3G/4G networks. Mobile user can request a specific service through specific mobile application or web browser available or installed on his/her mobile device. The mobile application or web browser will then initiate the mutual authentication amid user and cloud service provider. This authentication once completed will let the user to enjoy the services and resources offered over the cloud. The authentication scheme should be lightweight and secure to bring computational efficiency

at resource constrained mobile device and protect it from adversary attacks. Moreover, authentication scheme should maintain privacy of the user to prevent celebrated identity tracing and identity impersonation attacks [176, 177].

Distributed cloud computing invoke major concerns about key management, because mobile users are interested to utilize diverse MCC services from various service providers that in turn require separate user accounts for every service provider along with separate password or secret keys to perform authentication. Therefore, mobile users will certainly appreciate such solution that only demand single password or secret key in order to access various services available on distinct clouds. OpenID and Passport and many other schemes that are categorized under Single Sign-On (SSO) schemes can be considered as probable solution for key management concerns in MCC. These schemes require single password or secret key to access cloud services from distinct service provides. Majority of SSO schemes entail third party for authenticating each user but such schemes fails to perform reliably and efficiently, if specific third party itself is crashed or become inundate by too many service requests at a particular time interval. Moreover, protocol for secure message transmission is also necessary to maintain the integrity and confidentiality of the exchanged messages among the participants [12, 178]. Unfortunately, most of the transmission protocols that are developed so far, require intensive communication cost. Therefore, consequently they are considered to be infeasible for resource constrained mobile devices.

Customary authentication schemes induce substantial computations due to common public key cryptosystems as these public key cryptosystems like Discrete Logaritheorem Problem (DLP) and RSA demand larger size key and in turn devour computation resources rapidly specially in resource constrained devices such as mobiles. Elliptic curve cryptography (ECC) relatively can be considered preferable for mobile devices as it offers equal strength at the cost of trivial key size [179–181]. Bilinear pairing is then introduced in elliptic curve, in order to establish an ID-based encryption and decryption procedures. Since then researchers have focused on developing ID-based cryptosystems because they have resolved the major problem related to public key cryptosystems in terms of high computation cost that is incurred during authentication and management of public keys [182]. Generally, ID-based cryptosystems derive the public key of a particular user by using his/her unique identity that in turn mitigate lots of computation, verification and storage overhead required for maintaining, computing and verifying public keys of other users in customary public key cryptosystems. Many attempts have been made to implement ID-based cryptosystems in distributed cloud and grid computing networks. The first attempt in grid environment is made in 2004 by Lim and Robshaw [183] and then in 2005, they have elaborated the concept of dynamic key

infrastructure in grid [184]. Li et al. [185] proposed an ID-based authentication scheme in 2009, for providing a secure and reliable authentication solution within cloud computing architecture. But [186, 187] find out that Li et al. scheme fails to provide invincibility against untraceability and user anonymity.

Majority of customary ECC or bilinear pairing based authentication schemes are specifically developed for client server architecture therefore it is difficult to implement them in distributed service architecture [188, 189]. The imperative problem is to maintain and manage several secret keys from various service providers. Although this problem can be solved if all service providers are encouraged to share a common master secret key. But this could leads towards another problem if an adversary is able to successfully compromise any of the service providers, he/she can easily impersonate as any other service provider to dodge all users. Moreover, an adversary after accessing master secret key can in turn compromise the session keys maintained between any other service provider and user in the absence of perfect forward secrecy in the implemented authentication scheme. So, adversary can intercept each and every information exchanged between the two. Thus this method can be concluded as inappropriate for distributed mobile cloud architecture. A viable solution is suggested by Tsai and Lo [12]. In their method they suggested a SSO for numerous cloud service providers. It is to be noted that all the service providers are not assumed as trusted in distributed mobile cloud environments. Fig. 11.1 depicts a typical desirable authentication scenario for distributed mobile cloud scenario. Where the users and service providers initially get registered with registration center, which in turns assign ID-based key pairs. Then each user is allowed to get mutually authenticated with his desired service provider without intervention of the registration center.

11.0.1 Motivation and Contributions

Very recently, Tsai and Lo [12] mentioned that most of the existing authentication schemes are designed for single server environments. Hence, are not suitable for distributed mobile cloud environments where a SSO can provide services from various service providers. Therefore, Tsai and Lo claimed to propose a novel authentication scheme for distributed mobile cloud computing environment, secure against the known attacks. In their scheme, Tsai and Lo make use of a single ID based private key to access multiple servers.

However, in this chapter we show that the scheme proposed by Tsai and Lo [12] is vulnerable to service provider forgery attack. Then we propose an improved scheme to safeguard against known attacks. The proposed scheme is having following merits:

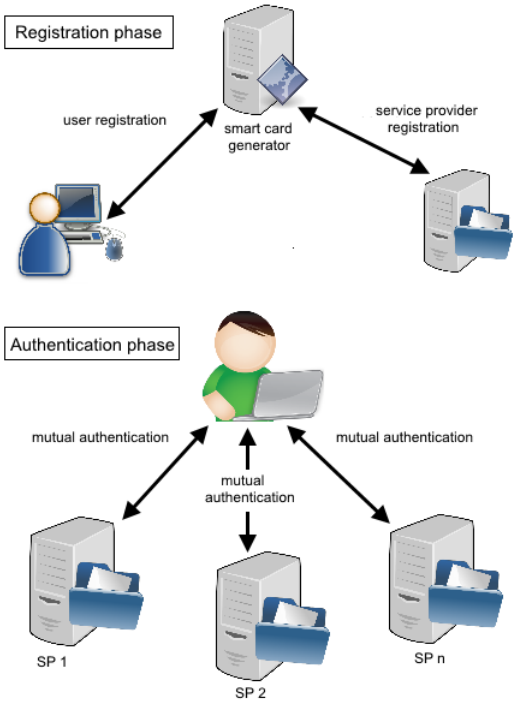


Figure 11.1: Authentication scenario for distributed MCC

- The proposed scheme is thoroughly investigated and proved to be secure in random oracle model under the hardness assumptions of $k-CAA$, CDH and $DCDH$ problems.
- The proposed scheme is secure under the protocol validation model of popular automated tool ProVerif.
- The proposed scheme achieves same online computation cost as compared with original Tsai and Lo's scheme.
- The proposed scheme provides user anonymity and untraceability.

11.0.2 Roadmap of the chapter

Rest of the chapter is systematized as follows. Section 11.1, reviews Tsai and Lo's novel authentication scheme for distributed mobile cloud computing environment. Section 11.2, cryptanalyzes Tsai and Lo's scheme and proves it to be vulnerable to server forgery attack. Section 11.3, demonstrates the improved proposed scheme. Section 11.4, proves the security of proposed scheme in random oracle model and under performs the security analysis of the proposed protocol in the random oracle model and the protocol validation model of popular automated tool ProVerif. Section 11.6, incorporates the security and performance analysis of

Table 11.1: Notation Guide

Notations	Description		
U_i, S_j	User i, Service provider j	G_1, G_2	Cyclic multiplicative, Additive group of q
e, P	A bilinear pairing group, Generator of G_1	$H(.), SCG$	Oneway hash function, Smart card generator
ID_i, S_i	Identity and private key of user i	ID_j, S_j	Identity and private key of service provider j
$H(ID_i), H(ID_j)$	Public keys of user i and service provider j	$s, P_{pub} = sP$	Private and public key pair of SCG
K_{ij}, $\|$	Session key of S_j and U_i, Concatenation		

the proposes scheme with Tsai and Lo's scheme. Finally, chapter's summary is solicited in Section 11.7.

11.1 Review of Tsai and Lo's Scheme

This section briefly reviews Tsai and Lo's privacy aware authentication scheme for mobile cloud environments. The scheme consists of following three phases:

11.1.1 System Setup Phase

SCG selects two cyclic groups G_1 over addition and G_2 over multiplication of same prime order q. Let P be the generator of G_1. SCG then chooses his private key s and computes his public key $P_{pub} = sP$. Further, SCG computes $e(P, P)$ and selects the pairing function $e : G_1 \times G_1 \rightarrow G_2$ along with five secure hash functions $H_1 : Z_p \rightarrow Z_p$, $H_2 : G_2 \rightarrow Z_p$, $H_3 : Z_p \rightarrow Z_p$, $H_4 : Z_p \rightarrow Z_p$ and $h : Z_p \rightarrow G_1$. Finally SCG publishes the public parameters $\{e, H_1, H_2, H_3, H_4, h, P, P_{pub}, e(P, P)\}$ and keeps his private key s secret.

11.1.2 Registration Phase

For registration, every participant P_k (user U_i or service provider S_j) selects his identity ID_k and sends it to SCG. Upon reception, SCG computes his private key as follows:

$$S_k = \frac{1}{s + H_1(ID_k)} P \tag{11.1}$$

Where ID_k can either be the identity of user or the service provider. ECG sends the private key S_k to each participant P_k using some secure channel. The user U_i on reception of his private key S_i computes $E_i = S_i \oplus h(PW_i \| f_i)$, where PW_i and f_i are the password and

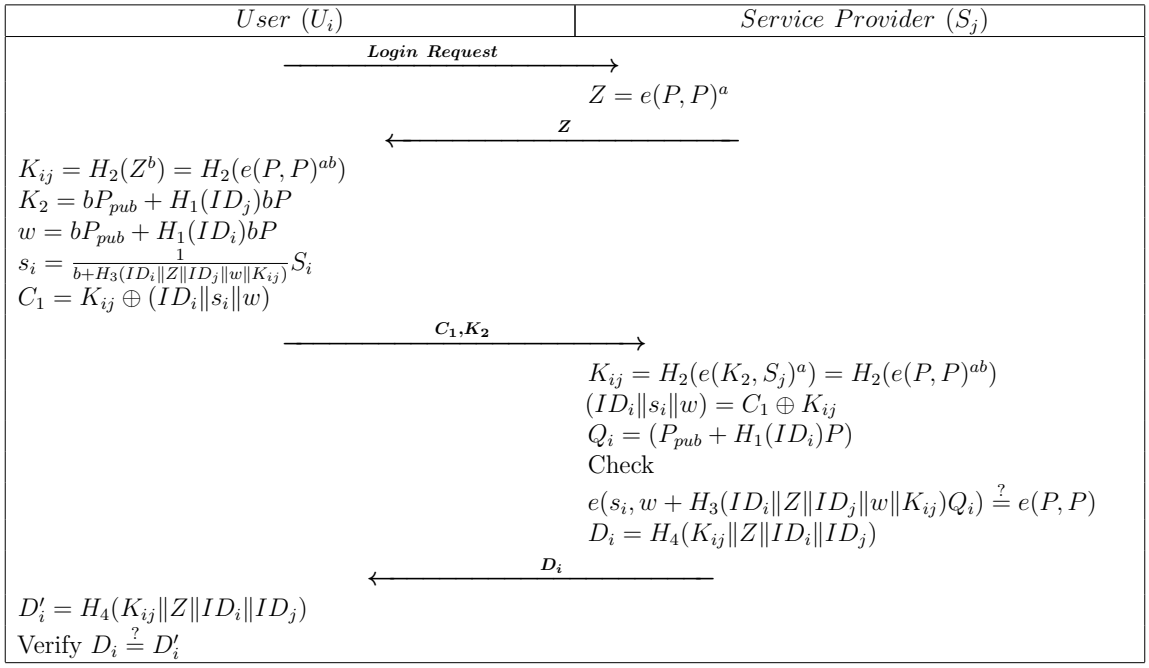


Figure 11.2: Tsai and Lo's Scheme

biometrics/fingerprints of the user U_i. Further, U_i stores E_i in his smart card. While S_j stores his private key S_j in some secure memory accessible only to S_j.

11.1.3 Authentication

Authentication phase is initiated by a user U_i, when he wants to acquire services by some service provider S_j. U_i enters his smart card in reader and inputs his password PW_i and biometrics/fingerprints f_i. The smart card then computes $S_i = E_i \oplus h(PW_i\|f_i)$. Following steps are performed between U_i and S_j, which are also illustrated in Fig. 11.2.

Step TA1: $U_i \rightarrow S_j : \{login\ request\}$

U_i sends login request to service provider S_j.

Step TA2: $S_j \rightarrow U_i : \{Z\}$

S_j selects some random number a and computes:

$$Z = e(P,P)^a \tag{11.2}$$

Further, S_j sends Z to U_i.

Step TA3: $U_i \rightarrow S_j : \{C_1, K_2\}$

U_i upon reception of Z, chooses some random number b and computes:

$$K_{ij} = H_2(Z^b) = H_2(e(P,P)^{ab}) \tag{11.3}$$

$$K_2 = bP_{pub} + H_1(ID_j)bP \tag{11.4}$$

$$w = bP_{pub} + H_1(ID_i)bP \tag{11.5}$$

$$s_i = \frac{1}{b + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})} S_i \tag{11.6}$$

$$C_1 = K_{ij} \oplus (ID_i \| s_i \| w) \tag{11.7}$$

U_i sends (C_1, K_2) to S_j.

Step TA4: $S_j \rightarrow U_i : \{D_i\}$

$$K_{ij} = H_2(e(K_2, S_j)^a) = H_2(e(P,P)^{ab}) \tag{11.8}$$

$$(ID_i \| s_i \| w) = C_1 \oplus K_{ij} \tag{11.9}$$

$$Q_i = (P_{pub} + H_1(ID_i)P) \tag{11.10}$$

S_j then computes $e(s_i, w + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})Q_i)$ and checks:

$$e(s_i, w + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})Q_i) \stackrel{?}{=} e(P,P) \tag{11.11}$$

If Eq. 11.11 holds true, S_j perceives U_i as authenticated and computes:

$$D_i = H_4(K_{ij} \| Z \| ID_i \| ID_j) \tag{11.12}$$

S_j sends D_i to U_i.

Step TA5: For the received message D_i, U_i computes:

$$D_i' = H_4(K_{ij} \| Z \| ID_i \| ID_j) \tag{11.13}$$

Finally, U_i checks:

$$D_i \stackrel{?}{=} D_i' \tag{11.14}$$

If Eq. 11.14 holds true, U_i assumes S_j authenticated. The session key computed by

both U_i ans S_j is as follows:

$$K_{ij} = H_2(Z^b) = H_2(e(P,P)^{ab}) \tag{11.15}$$

11.2 Cryptanalysis of Tsai and Lo's Scheme

This section shows that Tsai and Lo's authentication scheme for distributed mobile cloud environments is vulnerable to server forgery attack. We show that an adversary just after acquiring the identities of a user and the service provider can forge himself as a legitimate service provider. We first describe the common adversarial model, then show that under the mentioned adversarial model Tsai and Lo's scheme is vulnerable to server forgery attack.

11.2.1 Adversarial Model

We have adopted the common adversarial model as mentioned [25–27]. Where following assumption are made according to the capabilities of adversary ($\mathcal{A}$):

1. $\mathcal{A}$ is assumed to fully control the communication channel, precisely $\mathcal{A}$ can intercept, add, block, replay, modify or can send forged message to any participant.
2. $\mathcal{A}$ can be some insider having knowledge of system's public parameters or can be an outsider.
3. $\mathcal{A}$ is having the knowledge of public identities of the registered users and service providers.

11.2.2 Server Forgery Attack

Here, we show that an adversary $\mathcal{A}$ can easily forge himself as the legitimate service provider S_j under the common adversarial model as illustrated in subsection 11.2.1. Following steps are performed between $\mathcal{A}$ and U_i for a successful forgery attack:

Step SFA1: U_i sends login request to service provider S_j. $\mathcal{A}$ intercepts the message and selects some random number a and computes:

$$Z = e(P_{pub} + H_1(ID_j)P, P)^a \tag{11.16}$$

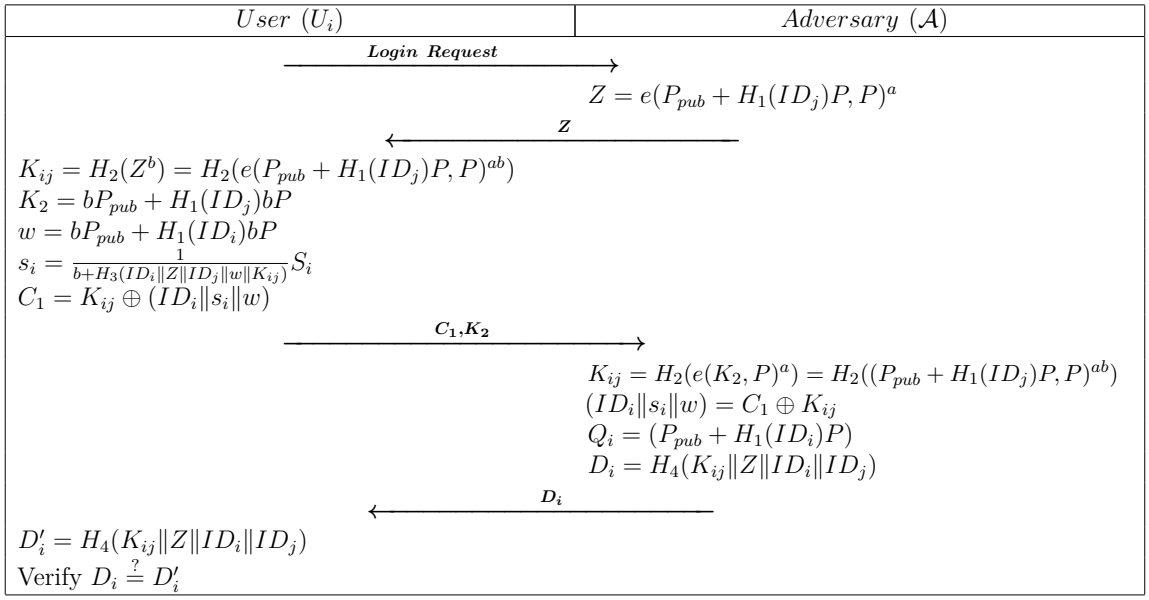


Figure 11.3: Forgery Attack on Tsai and Lo's Scheme

Further, $\mathcal{A}$ sends Z to U_i.

Step SFA2: U_i upon reception of Z, chooses some random number b and computes:

$$K_{ij} = H_2(Z^b) = H_2(e(P_{pub} + H_1(ID_j)P, P)^{ab}) \tag{11.17}$$

$$K_2 = bP_{pub} + H_1(ID_j)bP \tag{11.18}$$

$$w = bP_{pub} + H_1(ID_i)bP \tag{11.19}$$

$$s_i = \frac{1}{b + H_3(ID_i\|Z\|ID_j\|w\|K_{ij})}S_i \tag{11.20}$$

$$C_1 = K_{ij} \oplus (ID_i\|s_i\|w) \tag{11.21}$$

U_i then sends (C_1, K_2) to S_j. $\mathcal{A}$ intercepts the message and computes:

$$K_{ij} = H_2(e(K_2, P)^a) = H_2((P_{pub} + H_1(ID_j)P, P)^{ab}) \tag{11.22}$$

$$(ID_i\|s_i\|w) = C_1 \oplus K_{ij} \tag{11.23}$$

$$Q_i = (P_{pub} + H_1(ID_i)P) \tag{11.24}$$

$$D_i = H_4(K_{ij}\|Z\|ID_i\|ID_j) \tag{11.25}$$

$\mathcal{A}$ sends D_i to U_i.

Step SFA3: For the received message D_i, U_i computes:

$$D_i' = H_4(K_{ij} \| Z \| ID_i \| ID_j) \tag{11.26}$$

Finally, U_i checks:

$$D_i \stackrel{?}{=} D_i' \tag{11.27}$$

If Eq. 11.27 holds true, U_i assumes $\mathcal{A}$ as authenticated service provider S_j. The session key computed by both U_i and $\mathcal{A}$ is as follows:

$$K_{ij} = H_2(e(K_2, P)^a) = H_2((P_{pub} + H_1(ID_j)P, P)^{ab}) \tag{11.28}$$

Proposition 2. *At end of the forgery attack, the user U_i accepts the adversary $\mathcal{A}$ as the legitimate service provider S_j.*

Proof. During authentication, U_i authenticates S_j on the basis of D_i and the session key K_{ij}. For successful forgery attack, following two condition must be satisfied:

1. K_{ij} computed by U_i is same as computed by $\mathcal{A}$.
2. D_i' computed by U_i and D_i computed by $\mathcal{A}$ are same.

$\mathcal{A}$ sends $Z = e(P_{pub} + H_1(ID_j)P, P)^a$ in Eq. 11.16. Then U_i computes K_{ij} as in Eq. 11.18, similarly $\mathcal{A}$ computes K_{ij} in Eq. 11.23. We now show that K_{ij} computed in Eq. 11.18 is same as computed in Eq. 11.23.

$$\begin{aligned} K_{ij} &= H_2(Z^b) && \text{By Eq. 11.18} \\ &= H_2((e(P_{pub} + H_1(ID_j)P, P)^a)^b) \\ &= H_2(e(bP_{pub} + H_1(ID_j)bP, P)^a) \\ &= H_2(e(K_2, P)^a) \\ &= K_{ij} && \text{By Eq. 11.23} \end{aligned}$$

Similarly, $\mathcal{A}$ computes D_i in Eq. 11.25 while U_i computes D_i' in Eq. 11.26. Now we prove that $D_i = D_i'$.

$$\begin{aligned} D_i &= H_4(K_{ij} \| Z \| ID_i \| ID_j) && \text{By Eq. 11.25} \\ D_i' &= H_4(K_{ij} \| Z \| ID_i \| ID_j) && \text{By Eq. 11.26} \end{aligned}$$

As it is already proved that K_{ij} computed on both sides is same. Likewise Z is also same. So we have:

$$D_i = D_i'$$

Hence, U_i accepts adversary $\mathcal{A}$ as the legitimate service provider S_j. □

11.3 Proposed Scheme

This section describes the proposed authentication scheme based on Tsai and Lo's scheme. Similar to Tsai and Lo, proposed scheme can be described by following three phases: (1) System setup; (2) Registration; and (3) Authentication. We have modified only authentication phase, while system setup and registration phases are taken from Tsai and Lo's scheme in its present form. The proposed scheme as illustrated in Fig. 11.4 is explained in following subsection.

11.3.1 Authentication

Authentication phase is initiated by a user U_i, when he wants to acquire services by some service provider S_j. U_i enters his smart card in reader and inputs his password PW_i and biometrics/fingerprints f_i. The smart card then computes $S_i = E_i \oplus h(PW_i \| f_i)$. Following steps are performed between U_i and S_j:

Step PA1: $U_i \rightarrow SP_j : \{login\ request\}$
U_i sends login request to service provider SP_j.

Step PA2: $SP_j \rightarrow U_i : \{Z\}$
SP_j selects some random number a and computes:

$$Z = e(P, P)^a \tag{11.29}$$

Further SP_j sends Z to U_i.

Step PA3: $U_i \rightarrow SP_j : \{C_1, K_2\}$
U_i upon reception of Z, chooses some random number b and computes:

$$K_{ij} = H_2(Z^b) \tag{11.30}$$

$$K_2 = bP_{pub} + H_1(ID_j)bP \tag{11.31}$$

$$w = bP_{pub} + H_1(ID_i)bP \tag{11.32}$$

$$s_i = \frac{1}{b + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})} S_i \tag{11.33}$$

$$C_1 = K_{ij} \oplus (ID_i \| s_i \| w) \tag{11.34}$$

U_i sends (C_1, K_2) to SP_j.

Step PA4: $SP_j \rightarrow U_i : \{D_i, K_1\}$

$$K_{ij} = H_2(e(K_2, SP_j)^a) = H_2(e(P, P)^{ab}) \tag{11.35}$$

$$(ID_i \| s_i \| w) = C_1 \oplus K_{ij} \tag{11.36}$$

$$Q_i = (P_{pub} + H_1(ID_i)P) \tag{11.37}$$

SP_j then computes $e(s_i, w + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})Q_i)$ and checks:

$$e(s_i, w + H_3(ID_i \| Z \| ID_j \| w \| K_{ij})Q_i) \stackrel{?}{=} e(P, P) \tag{11.38}$$

If Eq. 11.11 holds true, SP_j perceives U_i as authenticated and computes:

$$K_1 = aP_{pub} + H_1(ID_i)aP \tag{11.39}$$

$$D_i = H_4(K_{ij} \| Z \| ID_i \| ID_j) \tag{11.40}$$

SP_j sends $D_i,\ K_1$ to U_i.

Step PA5: For the received message D_i, U_i computes:

$$E_i = H_2(e(K_1, S_i)^b) \tag{11.41}$$

$$D_i' = H_4(E_i \| Z \| ID_i \| ID_j) \tag{11.42}$$

Finally U_i checks:

$$D_i \stackrel{?}{=} D_i' \tag{11.43}$$

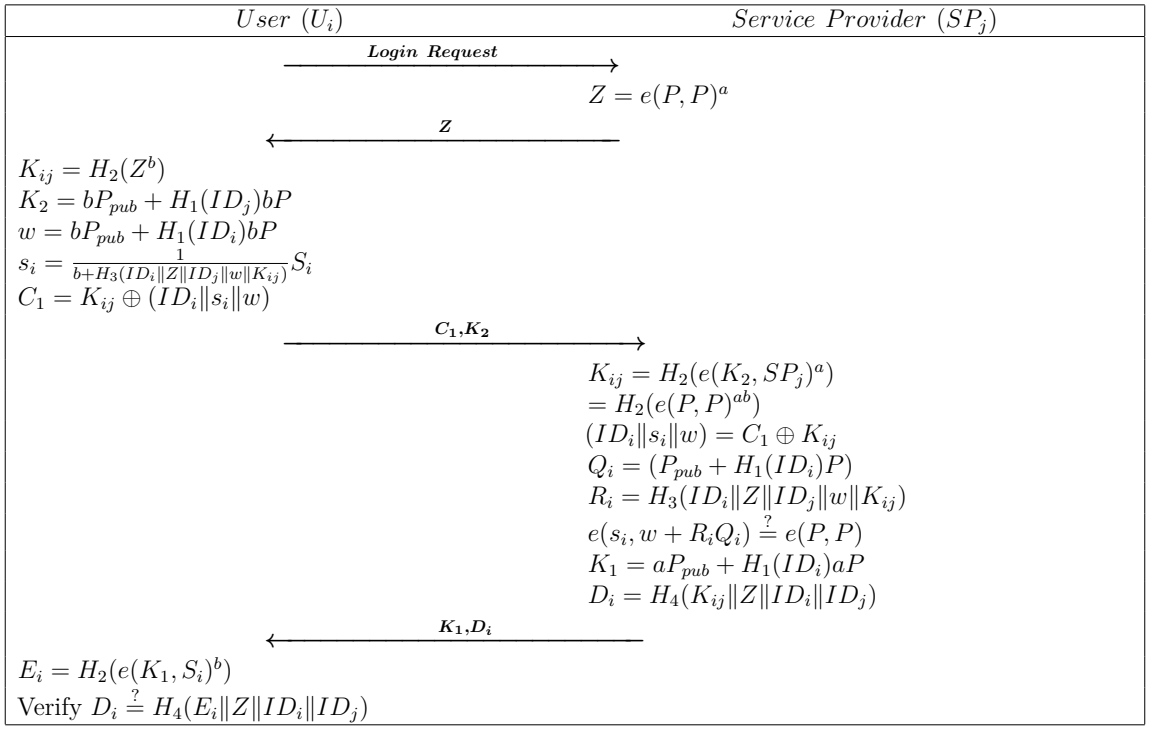


Figure 11.4: Proposed Scheme

If Eq. 11.43 holds true, U_i assumes SP_j authenticated. The session key computed by both U_i and SP_j is as follows:

$$K_{ij} = H_2(Z^b) = H_2(e(P,P)^{ab}) \tag{11.44}$$

11.3.2 Correctness

Here, we show that the session key generated on both sides is same, which confirm the correctness of proposed scheme.

$$K_{ij} = H_2(e(K_2, SP_j)^a) \tag{11.45}$$

$$= H_2(e(bP_{pub} + (H_1(ID_j)bP\frac{1}{s + H_1(ID_j)}.P))^a) \tag{11.46}$$

$$= H_2(e(P,P)^{(b(s+H_1(ID_j))(\frac{1}{s+H_1(ID_j)})^a)}) \tag{11.47}$$

$$= H_2(e(P,P)^{ab}) \tag{11.48}$$

$$= H_2(Z^b) \tag{11.49}$$

11.4 Security Analysis

This section formally describes the security of proposed scheme under random oracle model.

A. Security Model

The protocol model $\mathcal{P}$ consists of two participants: a user U and a service provider S. While $\mathcal{P}$ is in execution, there are a number of instances of each participant U and S. Each participant instance is associated with an identifier i putative as an oracle involved in $\mathcal{P}$'s divergent execution. Let U^i and S^j are the i^{th} and j^{th} instances of U and S respectively. With out differences, we denote X^k as an instance for both U^i and S^j. An oracle results to three states: (1) accept; (2) reject; and (3) $\perp$. The oracle leads to accept if it got right answer. The incorrect answer results to reject state, while $\perp$ is a result when no answer is received. Following are the adversary's capabilities:

1) **Extract**(ID_i): This query enables A to obtain U^i's private key related to its identity ID_i.

2) **Send**(M, X^k): By this query, the adversary A can send an arbitrary message M and obtains the computation result by the oracle.

3) **H**(i, m): This is hash oracle which outputs an arbitrary value r. Employment of this query builds a record (m, r). According to first parameter it generates four different lists L_{HL1}, L_{HL2}, L_{HL3} and L_{HL4}. All the four lists L_{HLi} are initially empty.

4) **Reveal**(X^k): Using this query A can obtain the session key K_{ij} from an oracle.

5) **Corrupt**(X^k): This query enables the adversary to obtain private key of the participating entity X^k.

6) **Test**(X^k): This query works for getting the session key. $Test(X^k)$ outputs $\perp$, if no session key is generated by X^k. Otherwise, it's employment results into flipping of a coin ω. If $\omega = 1$, existent session key is returned otherwise a random string is returned.

The employed definitions of partnering and freshness are described as follows:

1) Partnering: Two participants U^i and S^j are said to be partner if following conditions are met:

i) $U^i \in U$ and $SP_j \in S$.

ii) The shared session key K_{ij} is same on both sides.

iii) Only U^i and S^j has joined the distinct session.

2) Freshness: A session key constructed by an oracle and its partner is fresh if the following conditions hold.

i) Both U^i and S^j has shared a session key $K_{ij} \neq NULL$, while no *Reveal* query has been invoked by any of the partner.

ii) $Send(X^k, M)$ is called after the *Corrupt* query is called.

We denote $Succe(A)$ as the event, where A guesses ω selected in $Test$ query. A advantage is defined as: $Adv_{A,\mathcal{P}}(k) = |2.Pr\left[Succe(A)\right] - 1|$.

B. Security Analysis

The security analysis is very similar to Tsai and Lo's scheme except the answer to some queries. Following three theorems proves the security of proposed scheme, further theorem 4 is solicited to incorporate the anonymity and untraceability. Before initiating the proof process, following definitions are introduced. Let G_1 be a cyclic additive group of prime order q.

1) Definition 1 (k-CAA Problem): Given an integer k and s, $P \in G_1$, sP, $\{x_1, x_2, \ldots, x_k \in Z_q^*\}$, $\{(1/(x_1 + s))P, (1/(x_2 + s))P, \ldots, (1/(x_1 + s))P\}$ it is computationally infeasible to compute $1/(x_0 + s))P$ for $x \notin \{x_1, x_2, \ldots, x_k\}$.

2) Definition 2 (DCDH problem): Given $a, b \in Z_q^*$, $P \in G_1$, aP, bP it is computationally infeasible to compute $ab^{-1}P$.

3) Definition 3 (CDH problem): Given $a, b \in Z_q^*$, $P \in G_1$, aP, bP it is computationally infeasible to compute abP.

Let $Enc_k(M)/Dec_k(M)$ illustrates an exclusive or operation for encryption and decryption of a message M using key k. For analysis purposes, the mentioned $Hash$, $Send$, $Reveal$, $Corrupt$, $Execute$ and $Test$ queries are simulated as per the real attacks.

Theorem 10. *The proposed authentication schemes achieves user to service provider (U_i to SP_j) authentication provided H_1, H_2, H_3 and H_4 are modeled as random oracles and under the hardness assumption of $k - CAA$ problem. Contrarily, if an adversary A can violate U_i to SP_j authentication scheme, then there exists a polynomial time*

algoritheorem C, which can solve $k-CAA$ problem.

Proof. Let A can sabotage U_i to SP_j mutual authentication. Initially, C learns an instance $\{P, sP, \{x_1, x_2, \ldots, x_k \in Z_q^*\}, (1/(x_1+s))P, \{(1/(x_2+s))P, \ldots, (1/((x_k+s)))P\}\}$ of k-CAA problem. Goal of A is to compute $(1/(x_0+s))P$. A runs set up system algoritheorem to compute public parameters $\{x_1, x_2, \ldots, x_k \in Z_q^*\}$, $\{(1/(x_1+s))P, (1/(x_2+s))P, \ldots, (1/(x_1+s))P\}$. Following queries are simulated for interaction between A and C:

H_1 Query: When this query is asked for ID_i, C checks the maintained list L_{HL1}. C returns the result R_1 if found in the list, otherwise C computes $R_1 = H_1(ID_i)$. Stores the pair (ID_i, R_1) in L_{HL1} and returns R_1 to A.

H_2 Query: When this query is asked on $e(K_1, S_i b)$, C checks the maintained list L_{HL2}. C returns the result R_2 if found in the list, otherwise C computes $R_2 = H_2(Z^b)$. Stores the pair (Z^b, R_2) in L_{HL2} and returns R_2 to A.

H_3 Query: When A asks this query on $(ID_i, Z, ID_j, \omega, K_{ij})$. C checks the existence of tuple $(ID_i, Z, ID_j, \omega, K_{ij})$ in L_{HL3}. If the tuple exists then C sends $R_3 = H_3(ID_i \| Z \| ID_j \| \omega \| K_{ij})$ to A. Otherwise, C computes $R_3 = H_3(ID_i \| Z \| ID_j \| \omega \| K_{ij})$ and stores $(ID_i, Z, ID_j, \omega, K_{ij}, R_3)$ in L_{HL3} then returns R_3 to A.

H_4 Query: Upon reception of this query on (K_{ij}, Z, ID_i, ID_j). C checks the record in list L_{HL4}, if the record exists in L_{HL4}, C returns $R_4 = H_4(K_{ij} \| Z \| ID_i \| ID_j)$ to A. Otherwise, C computes $R_4 = H_4(K_{ij} \| Z \| ID_i \| ID_j)$ and stores $(K_{ij}, Z, ID_i, ID_j, R_4)$ in L_{HL4}. Finally, C returns R_4 to A.

Extract: When this query is asked on ID_i, C checks $H_1(ID_i) \in \{x_1, x_2, \ldots, x_k \in Z_q^*\}$. The query is terminated by failure message, if $H_1(ID_i) \notin \{x_1, x_2, \ldots, x_k \in Z_q^*\}$, the occurrence of this event is denoted as E_1. Further C checks $ID_i \in L_{HL1}$ and sends corresponding S_i to A. Otherwise C computes and sends $S_i = (1/(s + H_1(ID_i))P)$ to A.

Send Query: Send queries replicates the active attacks on communication and are categorized as follows:

1) **Send**$(U_i, INIT)$: C generates login request against this query.
2) **Send**$(SP_j, Login)$: When this query is asked, C selects a random number a and computes the pair $Z = e(P, P)^a$, $K_1 = aP_{pub} + H_1(ID_i)aP$. C then returns Z to A.

3) **Send**(U_i, Z):

Against this query, C checks $H_1(ID_i) \in \{x_1, x_2, \ldots, x_k \in Z_q^*\}$. The query is terminated by failure message, if $H_1(ID_i) \notin \{x_1, x_2, \ldots, x_k \in Z_q^*\}$, the occurrence of this event id denoted as E_2. C then selects a random number b and computes $K_{ij} = H_2(Z^b),\ K_2 = bP_{pub} + H_1(ID_j)bP,\ \omega = (bP_{pub} + H_1(ID_j)bP),\ s_i = (1/(b + H_3(ID_i\|Z\|ID_j\|\omega\|K_{ij})))SP_j,\ C_1 = E_{K_ij}(ID_i\|s_i\|\omega)$.

4) **Send**$(SP_j, (K_2, C_1))$: When A asks this query, C computes $K_{ij} = H_2(e(K_2, SP_j)^a)$, $(ID_i\|s_i\|\omega) = K_{ij} \oplus C_1,\ Q_i = P_{pub} + H_1(ID_i)P$ and checks whether $e(s_i, \omega + H_3(ID_i\|Z\|ID_j\|\omega\|K_{ij})Q_i)$ is equal to $e(P, P)$. C sends failure if it does not holds. Otherwise C computes and returns $D_i = H_4(K_{ij}\|Z\|ID_i\|ID_j)$, $K_1 = aP_{pub} + H_1(ID_i)aP$ to A.

5) **Send**(U_i, D_i): When this query is invoked C computes $E_i = H_2(e(K_1, S_i)^b)$, $D_i' = H_4(E_i\|Z\|ID_i\|ID_j)$ and checks the equality of D_i' with received D_i. C authenticates A if equality exists. Otherwise request is rejected by C.

Analysis: The adversary Adv can violate the U_i to SP_j authentication with out having U_i's private key, if he can generate forged signatures (ω', s_i') based on authentication message $(ID_i\|Z\|ID_j\|\omega\|K_{ij})$. In order to qualify the forged signature (ω', s_i') must pass the test mentioned in Eq. (11.38). Here an event E_3 is solicited to represent if $H_1(ID_i) \in \{x_1, x_2, \ldots, x_k \in Z_q^*\}$. Contrarily, if E_3 does not occur A can solve $k - CAA$ problem as C generates $(H_1(ID_i)) = x_i \notin \{x_1, x_2, \ldots, x_k \in Z_q^*\}$, $(1/(x_i+s))P \notin \{(1/(x_1+s))P, (1/(x_2+s))P, \ldots, (1/(x_k+s))P\}$.We also denote ϵ' as the advantage carried out by A and ϵ the advantage to break the proposed authentication scheme. The adversary can break the proposed scheme if he ables to generate valid results of $Extract$, $Send$ and $Hash$ queries which can happen only if none of the events $E_1,\ E_2,\ E_3$ occurred. The probability to break $k - CAA$ problem is as follows:

$$Pr[\neg E_1 \wedge \neg E_2 \wedge \neg E_3] = \left(\frac{q_{exq}}{q_{h1q}}\right)^{q_{exq}+q_{snq}} \left(\frac{q_{h1q} - q_{exq}}{q_{h1q}}\right) \tag{11.50}$$

Where C is allowed to make q_{h1q}, q_{exq} and q_{snq} queries relating to H_1, $Extract$ and $Send(Z, U_i)$ respectively. A's advantage is as follows:

$$\epsilon' \geq \left(\epsilon - \frac{1}{2^k}\right) \left(\frac{q_{exq}}{q_{h1q}}\right)^{q_{exq}+q_{snq}} \left(\frac{q_{h1q} - q_{exq}}{q_{h1q}}\right) \tag{11.51}$$

□

Theorem 11. *The proposed authentication schemes achieves service provider to user (SP_j to U_i) authentication provided H_1, H_2, H_3 and H_4 are modeled as random oracles and under the hardness assumption of DCDH problem. Contrarily, if an adversary A can violate SP_j to U_i authentication, then there exists a polynomial time algoritheorem C, which can solve DCDH problem.*

Proof. Initially C runs system setup algoritheorem and compute all public parameters $\{G_1, G_2, e, H_1, H_2, H_3, H_4, h, P, P_{pub}, Enc(.), Dec(.)\}$. C then interacts with A as follows:

H_1 Query: When this query is asked for ID_j, C checks the maintained list L_{HL1}. C returns the result R_1 if found in the list, otherwise C computes $R_1 = H_1(ID_i)$. Stores the pair (ID_i, R_1) in L_{HL1} and returns R_1 to A.

H_2 hash query: If A invokes an H_2 query on $e(Z, bP)$, B checks whether $e(Z, bP)$ exists in L_{H2}. If the later is found in L_{H2}, B returns h_2 to A; otherwise, B computes $h_2 = H_2(Z, bP)$ and then stores a new tuple $(e(Z, bP), h_2)$ in L_{H2}. Next B returns h_2 to A.

H_2 Query: When this query is asked on $e(Z^b)$, C checks the maintained list L_{HL2}. C returns the result R_2 if found in the list, otherwise C computes $R_2 = H_2(Z^b)$. Stores the pair $((e(Z^b, S_i b), R_2)$ in L_{HL2} and returns R_2 to A.

H_3 Query: When A asks this query on $(ID_i, Z, ID_j, \omega, K_{ij})$. C checks the existence of tuple $(ID_i, Z, ID_j, \omega, K_{ij})$ in L_{HL3}. If the tuple exists then C sends $R_3 = H_3(ID_i \| Z \| ID_j \| \omega \| K_{ij})$ to A. Otherwise, C computes $R_3 = H_3(ID_i \| Z \| ID_j \| \omega \| K_{ij})$ and stores $(ID_i, Z, ID_j, \omega, K_{ij}, R_3)$ in L_{HL3} then returns R_3 to A.

H_4 Query: Upon reception of this query on (K_{ij}, Z, ID_i, ID_j). C checks the record in list L_{HL4}, if the record exists in L_{HL4}, C returns $R_4 = H_4(K_{ij} \| Z \| ID_i \| ID_j)$ to A. Otherwise, C computes $R_4 = H_4(K_{ij} \| Z \| ID_i \| ID_j)$ and stores $(K_{ij}, Z, ID_i, ID_j, R_4)$ in L_{HL4}. Finally, C returns R_4 to A.

Send Query: Send queries replicates the active attacks on communication and are categorized as follows:

1) **Send**$(U_i, INIT)$: C generates login request against this query.
2) **Send**$(SP_j, Login)$: When this query is asked, C selects a random number a and computes the pair $Z = e(P, P)^a$. C then returns Z to A.
3) **Send**(U_i, Z):

Against this query, C selects a random number b and computes $K_{ij} = H_2(Z^b)$, $K_2 = bP_{pub}+H_1(ID_j)bP$, $\omega = (bP_{pub}+H_1(ID_j)bP)$, $s_i = (1/(b+H_3(ID_i\|Z\|ID_j\|\omega\|K_{ij})))$ SP_j, $C_1 = E_{K_ij}(ID_i\|s_i\|\omega)$. C then returns (K_2, C_1).

4) **Send**$(SP_j, (K_2, C_1))$: When A asks this query, C computes $K_{ij} = H_2(e(K_2, SP_j)^a)$, $(ID_i\|s_i\|\omega) = K_{ij} \oplus C_1$, $Q_i = P_{pub} + H_1(ID_i)P$ and checks whether $e(s_i, \omega + H_3(ID_i\|Z\|ID_j\|\omega\|K_{ij})Q_i)$ is equal to $e(P, P)$. C sends failure message if it does not holds. Otherwise C computes and returns $D_i = H_4(K_{ij}\|Z\|ID_i\|ID_j)$, $K_1 = aP_{pub} + H_1(ID_i)aP$ to A.

5) **Send**$(U_i, (D_i, K_1))$: When this query is invoked C computes $E_i = H_2(e(K_1, S_i)^b)$, $D_i' = H_4(E_i\|Z\|ID_i\|ID_j)$ and checks the equality od D_i' with received D_i. C authenticates A if equality exists. Otherwise request is rejected by C.

Analysis: We denote N_u as the number of user authentication instances, l the ECC bit length and k the size of H_4 digest. A can violated SP_j to U authentication, if he can generate forged D_i. Following are the three conditions required to forge D_i.

1) A guesses D_i without knowing K_{ij} and calling H_4. The probability for such guessing is less than $(1/2^k)$.

2) A need not to guess D_i, if the values K_1 and K_2 are same in two sessions. In such case A has to chalk out the identity ID_i, the probability for such case is less than $(N_u/2^{l^2})$.

3) If A intends to violate SP_j to U_i authentication by obtaining session key K_{ij} for some arbitrary $b, x, a \in Z_q^*$ such that $P_{pub} + H_1(ID_j) = xP$, and $K_2 = bP_{pub} + bH_1(ID_j) = bxP$. Then he has to break $DCDH$ problem. In such case the probability for guessing D_i correctly is ϵ'.

Precisely, in order to break SP_j to U_i authentication A has to solve $DCDH$ problem with advantage $\epsilon' \geq \left(1/2^k\right) - \left(N_u/2^{l^2}\right)$.

□

Theorem 12. *If A can guess the tossed coin b in $Test$ query then there exists a polynomial time algoritheorem C which can solve CDH problem.*

Proof. We denote E_sk the event that A accesses K_{ij} (the session key). Similarly, the event $Test(U_i)$ is declared for successful $Test$ query to U_i's oracle, E_{U2S} as the event for successful violation of U_i to SP_j authentication by A and $Test(SP_j)$ as the successful $Test$ query to SP_j's oracle. Following probability equation holds:

Table 11.2: Security Analysis

Scheme→ Security Properties↓	Our	[12]	[189]	[190]
Resistance to Replay attack	✓	✓	✓	✗
Resistance to Forgery attack	✓	✗	✓	✗
Resistance to Man-in-middle attack	✓	✓	✓	✓
User anonymity	✓	✓	✓	✗
User untraceability	✓	✓	✗	✗
No time synchronization	✓	✓	✓	✗
Provides multi-server authentication	✓	✓	✗	✗
Provable security	✓	✓	✓	✗

$$Pr[E_{sk} \wedge Test(U_i)] + Pr[E_{sk} \wedge Test(SP_j) \wedge E^{U2S}] + Pr[E_{sk} \wedge Test(SP_j) \wedge \neg E^{U2S}] \geq \frac{\epsilon}{2} \tag{11.52}$$

Let Pr_{U2S} be the probability for A to break U_i to SP_j authentication, i.e., $Pr_{U2S} = Pr[E_{sk} \wedge Test(SP_j) \wedge E^{U2S}]$. Then, we have

$$Pr[E_{sk} \wedge Test(U_i)] + Pr[E_{sk} \wedge Test(SP_j) \wedge \neg E^{U2S}] \geq \frac{\epsilon}{2} - Pr_{U2S} \tag{11.53}$$

Obviously, $Pr[E_{sk} \wedge Test(SP_j) \wedge \neg E^{U2S}] = 0$, and we have

$$Pr[E_{sk} \wedge Test(U_i)] \geq \frac{\epsilon}{2} - Pr_{U2S} \tag{11.54}$$

Referring Theorem 1 and 2 Pr_{U2S} is negligible while ϵ is non-negligible. Therefore, $(\epsilon/2) - Pr_{U2S}$ is also non-negligible. Hence to correctly guess b, A has to solve CDH problem. □

Theorem 13. *If A can violate user anonymity and untraceability then there exists a polynomial bound algoritheorem C which can solve $DCDH$ problem.*

Proof. A can violate user anonymity and untraceability, if he can decrypt C_1. For decryption A has to learn the session key $K_{ij} = H_2(e(P,P)^{ab})$. Referring to Theorem 2 the probability to obtain K_{ij} is identical as of solving $DCDH$ problem. Hence proposed scheme fulfills anonymity and untraceability.

□

```
(***************** Channels *****************)
free PCh:channel.
(********* Constants & Variables **********)
const P:bitstring.
free IDi:bitstring.
free Si:bitstring [private].
free IDj:bitstring.
free Sj:bitstring [private].
free s:bitstring [private].
(************** Constructors **************)
fun OWH(bitstring):bitstring.
fun OWH2(bitstring):bitstring.
fun OWH3(bitstring):bitstring.
fun OWH4(bitstring):bitstring.
fun INVRS(bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun SUM(bitstring,bitstring):bitstring.
fun MULT(bitstring,bitstring):bitstring.
fun ECPM(bitstring,bitstring):bitstring.
fun CONCAT(bitstring,bitstring):bitstring.
fun BLP0(bitstring,bitstring):bitstring.
fun BLP(bitstring,bitstring,bitstring):
    bitstring.
(***************** Events *****************)
event begin_Useri(bitstring).
event end_Useri(bitstring).
event begin_ServiceProviderj(bitstring).
event end_ServiceProviderj(bitstring).
```

(a) Declarations

```
(***************** Main *****************)
process ( (!SCG) | (!Useri) | (!
    ServiceProviderj) )
free Kij:bitstring [private].
query attacker(Kij).
query id:bitstring; inj event(end_Useri(id))
    ==> inj event(begin_Useri(id)) .
query id:bitstring; inj event(
    end_ServiceProviderj(id)) ==> inj event(
    begin_ServiceProviderj(id)) .
```

(c) Main

```
(***************** processes ***************)
(****************** User ******************)
let Useri =
in(PCh,(Ppub:bitstring));
in(PCh,(K1:bitstring));
new b:bitstring;
let Kij = OWH2(BLP(K1,Si,b)) in
let K2 =SUM(MULT(b,Ppub),MULT(OWH(IDj),ECPM(b,P
    ))) in
let w = SUM(MULT(b,Ppub),MULT(OWH(IDi),ECPM(b,P
    ))) in
let si = MULT(INVRS(SUM(b,OWH3(CONCAT(IDi,(Z,
    IDj,w,Kij))))),Si) in
let C1 = XOR(Kij,CONCAT(IDi,(si,w))) in
out(PCh,(C1,K2));
in(PCh,(Di:bitstring));
let Di' = OWH4(CONCAT(Kij,(Z,IDi,IDj))) in
if(Di' = Di) then
0.
(************ Server Provider *************)
let ServiceProviderj =
in(PCh,(Ppub:bitstring));
new a:bitstring;
let Z = BLP(P,P,a) in
let K1 = SUM(MULT(a,Ppub),MULT(OWH(IDi),ECPM(a,
    P))) in
out(PCh,(K1));
in(PCh,(C1:bitstring,K2:bitstring));
let Kij = OWH2(BLP(K2,Sj,a)) in
let (IDi:bitstring,si:bitstring,w:bitstring) =
    XOR(C1,Kij) in
let Qi = SUM(Ppub,ECPM(OWH(IDi),P)) in
if ( BLP0(si,MULT(SUM(w,OWH3(CONCAT(IDi,(Z,IDj,
    w,Kij)))),Qi)) = BLP0(P,P)) then
let Di = OWH4(CONCAT(Kij,(Z,IDi,IDj))) in
out(PCh,(Di))
else
0.
(********* Smart Card Generator **********)
let SCG =
let Ppub = ECPM(s,P) in
out(PCh,(Ppub));
0.
```

(b) Processes

Figure 11.5: ProVerif Validation

11.5 Protocol verification through ProVerif

We model the proposed scheme in ProVerif in order to analyze its robustness through automated tool. We have modeled the steps illustrated in section 11.3 and shown in Fig. 11.4. The modeled code in ProVerif is shown in Fig 11.5. The verification is performed on ProVerif 1.88 (latest version), the results are as follows:

RESULT inj-event(end˙ServiceProviderj(id)) ==> inj-event(begin˙ServiceProviderj(id)) is true.

RESULT inj-event(end˙Useri(id_17079)) ==> inj-event(begin˙Useri(id_17079)) is true.

RESULT not attacker (Kij) is true.

The results indicates that the both service provider and user events started and terminated successfully, while not *attacker (Kij) is true,* verifies that attacker is not able to find session key. Hence, proposed scheme posses authentication property.

11.6 Security and Performance Comparisons

This section illustrates the security and performance comparison of proposed scheme with Tsai and Lo's scheme. Referring Table 11.2 it can be easily seen that proposed scheme resists forgery attack, while Tsai and Lo's scheme is vulnerable to forgery attack. Following notations are solicited to elaborate computation comparison:

- t_{ebp}: Time to compute a bilinear mapping operation
- t_{mec}: Time to compute an ECC point multiplication operation
- t_{aec}: Time to compute a addition of two points
- t_{emp}: Time to compute map to point hash

For analysis we have adopted the same model as described in Tsai and Lo's scheme which is based on [191]. The running time considered by Tsai and Lo for t_{mec} on 1 GHz mobile HTC Desire HD is 42 ms, while t_{mec} and t_{ebp} on a device equipped with Intel Core2 Quad-core 2.40 GHz CPU and 3 GB RAM are 2.841 ms and 7.234 ms as mentioned on the website [192] of JPBC library [191]. We have also adopted the similar analogy as of Tsai and Lo, where some of the calculations are assumed to be precomputed. The proposed scheme achieves same computation overhead as of Tsai and Lo's scheme provided S_j precomputed $Z = e(P, P)^a$ and $K_1 = aP_{pub} + H_1(ID_i)aP$, while U_i precomputed $K_2 = bP_{pub} + H_1(ID_j)bP$. Furthermore, the computation times for oneway hash, bitwise XOR and concatenation operations are negligible, therefore ignored in analysis. From Table 11.3, it is verifiable that the proposed scheme achieves the authentication in approximately 152 ms. Proposed scheme also achieves same communication overhead as of Tsai and Lo's scheme because in both schemes the transmitted values are of same size and quantity.

Table 11.3: Computation Overhead Analysis

Scheme	User	Server
[189]	$4t_{mec}$	$6t_{mec} + 2t_{aec}$
[190]	$2t_{mec} + 1t_{aec} + 1t_{emp}$	$1t_{mec} + 1t_{aec} + 2t_{ebp} + 1t_{emp}$
[12]	$3t_{mec}$	$4t_{mec} + 2t_{ebp}$
Our	$3t_{mec} + 1t_{ebp}$	$4t_{mec} + 2t_{ebp}$

11.7 Chapter Summary

This chapter cryptanalyzed a recent authentication scheme for mobile cloud computing services proposed by Tsai and Lo. The analysis showed that Tsai and Lo's scheme is vulnerable

to server forgery attack. Furthermore, we proposed an improved authentication scheme incorporating user anonymity and untraceability. It is shown that proposed scheme while maintaining the computation and communication costs of Tsai and Lo's scheme also provides resistance to all known attacks. Hence, proposed scheme is more suitable for mobile cloud computing environments.

Chapter 12

A Signcryption Scheme and its Application in Electronic Payment Systems

With the rapid development of information and communication technologies, e-commerce has emerged as a viable solution to online shopping. During recent times the purchase of digital contents has been greatly increased, as per the statistics of U.S. Bureau of census, the online sale augmented from USD 99.50 billion to USD 343.43 billion during a thirteen years time span. Very similarly china's online market achieved USD 110.04 billion worth of business despite a number of challenges [193, 194]. Such growth in e-commerce is because of its speed, digitization and accessibility [195]. Electronic payment systems are considered as an integral part of any e-commerce system. Electronic payment systems are categorized into three basic types: Business to Business (B2B), Consumer to Consumer (C2C) and Business to Consumer (B2C). B2C e-payment got popularity after universalization of the Internet in early 90's. A number of B2C payment systems require credit cards for online payments. With the advent of e-payment systems, the users are having the expediency to save the time and money by using a number of services online (like payment of bills, purchase of goods etc.).

The primitive e-payment system was proposed by Chaum [196], after then many e-payment systems are proposed [195, 197–202]. While e-commerce is on its way to make daily life more convenient and easy, the main concerns in any e-payment system are security and privacy of participant and contents. The existing e-payment schemes make use of signatures to ensure user's authenticity and message integrity, while they cannot ensure user anonymity. Recently, Yang et al. [13] pointed out that in signature based schemes sender's signature is generated.

Further, the signature is verified on receiver side, this generation and verification of sender's signature burdened the system. Furthermore, the signature is sent on public network which may cause its illegal use. Therefore, Yang et al. [13] proposed a novel signcryption scheme and an e-payment system based on their signcryption scheme. In Yang et al.'s scheme, the sender makes use of his own private key and receiver's public key to form a symmetric key. The same symmetric key is generated by receiver by using his private key. They claimed to achieve the sender authenticity, message confidentiality and user anonymity as the symmetric key can only be generated by legitimate sender and reconstructed by intended legitimate receiver without generating and verifying the sender's signature.

In this chapter, we cryptanalyzed Yang et al.'s [13] signcryption scheme and e-payment system. We find both of their schemes to be vulnerable to impersonation attack. We show that an adversary just having the knowledge of public parameters can impersonate as a legitimate user. The attacker can easily exploit the weakness of Yang et al.'s scheme and can fraudulently purchase digital contents by deceiving the bank and merchant. Furthermore, we improved both Yang et al.'s signcryption scheme and e-payment system. We prove the security of our improved schemes using automated tool ProVerif.

Rest of the chapter is organized as follows. In section 12.1, we briefly describes signcryption and e-payment systems. In section 12.2, we review Yang et al.'s Signcryption scheme and its application in e-payment system. In section 12.3, we performed cryptanalysis of Yang et al.'s signcryption and e-payment schemes. Our improved signcryption scheme and e-payments are described in section 12.4. We prove the security of our proposed scheme in section 12.5. In section 12.6, we performed automated correctness and security verification of our scheme using ProVerif. The performance comparison is shown in section 12.7. Finally, chapter's summary is provided in section 12.8.

12.1 Preliminaries

This subsection briefly illustrates signcryption and e-payment systems.

12.1.1 Signcryption

The concept of signcryption (also termed as authenticated encryption) was first introduced by Zhang et al. [203]. Traditionally, authentication [59, 204, 205] and confidentiality [206] were considered two distinct tasks and to achieve them the sender first digitally signs

the message then performs encryption. Unfortunately, this approach is not suitable for resources constrained environments as it double-folds the computation and other requirements. Signcryption combines both the processes into a single process to reduce computation, communication and storage costs. A signcryption scheme involves two participants: the sender and the recipient. Initially, the sender generates a key, then encrypts the message and generates digital signatures based on message and public key of sender. Finally, the sender sends encrypted message and signature tuple to recipient. Upon reception of encrypted message and signature tuple, the recipient generates the same key and decrypts the message. Finally, the recipient verifies the signatures [129, 203, 207–209].

12.1.2 E-payment System

An e-payment system facilitates for transecting digital products. A general e-payment system consists of a customer, bank, merchant and a trusted third party to resolve a dispute. The basic aim of an e-payment scheme is to provide framework for online purchase of digital products while ensuring user's anonymity, fair exchange and dispute resolution. Fair exchange employees that none of the participant should have unfair advantage. In case of any dispute between the participants, the trusted third party is responsible for its resolution. A typical e-payment system is illustrated in Fig. 12.1. Before making any transaction, all the participants are supposed to register with the system, which in turns assigns a unique identity. Further, both merchant and customer must open some account to benefit secure e-payment. The participants are then required to select their private keys and compute and link their public keys with their bank account. A transactions in e-payment system is consisting of following five phases:

1. **Buying Phase:** The customer selects his desired goods from merchant's website, then he downloads the bill information from merchant's website. The customer then makes a valid payment order tuple and sends the payment order to the bank.

2. **Paying Phase:** Upon receiving the payment orders from customer, the bank checks the legality of the customer and validity of the payment order, if legality of the customer is not proved, the session is aborted by the bank. Otherwise, the bank deducts bill amount from customer's account and stores the bill amount in some temporary account. Finally, bank sends a unique payment voucher with some arbitrary expiry date to the customer.

3. **Exchanging Phase:** For the received payment voucher, the customer checks its

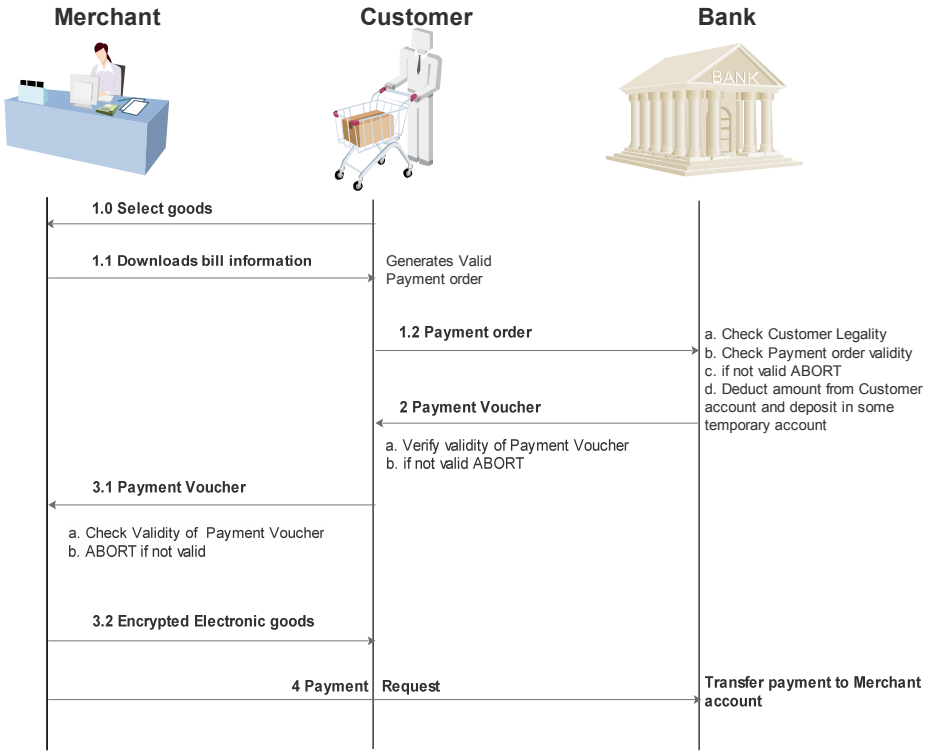


Figure 12.1: e-payment System

validity. If the voucher is not valid customer aborts the session, otherwise the customer generates a new message tuple based on payment voucher and sends it to the merchant. The merchant after receiving payment voucher checks the customer and voucher legality. The session is aborted if legality is not proved, otherwise merchant sends the encrypted electronic goods to the customer, which upon reception decrypts and use it.

4. **Transferring Phase:** The merchant sends the payment voucher to bank before expiry date. For the valid payment voucher the bank transfers the voucher amount to merchant's account.

5. **Dispute Resolution Phase:** This is an optional phase and can be committed either by customer or merchant if their arise some dispute among both.

12.1.3 E-payment Security Requirements

During e-payment transaction, the financial information is sent over insecure public network. So, it requires a robust security mechanism which can ensure mutual authentication, confidentiality, integrity, non-repudiation, privacy and prevention of double spending for a single

transaction. Following are the requisite security factors to be considered in an e-payment system.

- **Authentication:** The customer, bank and the merchant should authenticate each other during an e-transaction to avoid false transactions.
- **Confidentiality:** The transaction information must be hidden to outsiders. Further, each of the participant should only know his desired information.
- **Integrity:** No one should be allowed to modify the transaction data.
- **Non-repudiation:** The participants must not deny their role during a transaction.
- **Privacy Protection:** Each of the participants should only know his desired information. The bank should know only the amount to be billed not the goods information. Furthermore, information regarding the transactions must be hidden from outsiders.
- **Double Spending Prevention:** The merchant should be able to use the payment voucher only once. The system must refuse the replay of a previous payment voucher.

12.2 Review of Yang et al.'s Signcryption Scheme and E-payment System

This section reviews Yang et al.'s signcryption scheme and its application in e-payment. The scheme is based on elliptic curve cryptography [210–212]. Further, it does not require digital signatures for verification. The scheme and its e-payment version is described in the following subsections:

12.2.1 Yang et al.'s Signcryption Scheme

Yang et al.'s signcryption scheme consists of three phases initialization, signcryption and verification phases. The notation guide is illustrated in Table 12.1.

12.2.1.1 System Initialization Phase

During this phase, system selects finite field F_p over a large prime $p \geq 2^{160}$ and an elliptic curve $E_p(a, b)$. Further, it selects a base point P in $E_p(a, b)$ and symmetric key algorithm $E_k(.)/D_k(.)$, each legal participant chooses his private key d_i and computes his public key

Table 12.1: Notation Guide

Notations	Meaning		
p:	A large prime number ($p \geq 2^{160}$)	$E_p(a,b)$:	Selected elliptic curve
P:	A base point over $E_p(a,b)$	d_i:	Private key of i^{th} legal user
$Y_i = d_i \times P$:	Public key of i^{th} legal user	M:	Message (plaintext)
E_k/D_k:	Encryption/Decryption	T_i:	i^{th} Timestamp
$H(.)$:	A oneway hash function	$\mathcal{U}_i$:	Legal user/customer
$\mathcal{M}$, $\mathcal{B}$:	Merchant, Bank	$\mathcal{A}$:	Adversary

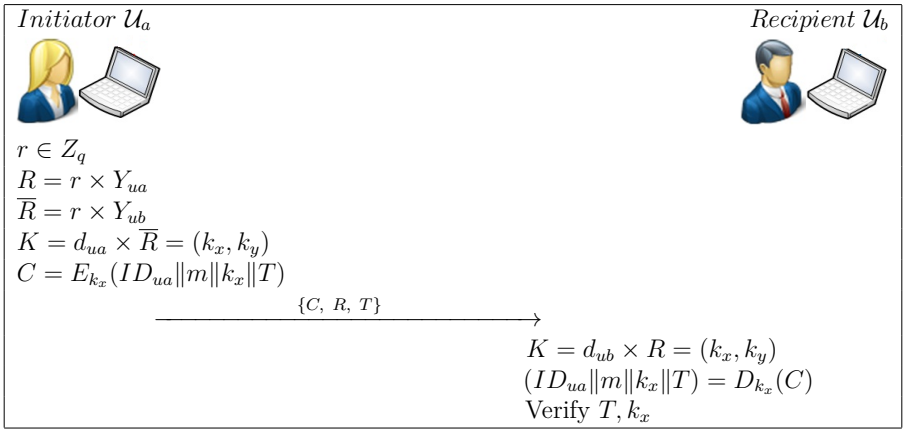


Figure 12.2: Yang et al.'s Signcryption Scheme

$Y_i = d_i \times P$. Finally, system parameters and each participant's public key are published, while each participant keeps his private key secret.

12.2.1.2 Signcryption Phase

During this phase a legal user $\mathcal{U}_a$ performs signcryption after obtaining another legal user $\mathcal{U}_b$'s public key Y_{ub}. $\mathcal{U}_a$ chooses a random number $r \in Z_q$ and computes $R = r \times Y_{ua}$, $\overline{R} = r \times Y_{ub}$ and $K = d_{ua} \times \overline{R} = (k_x, k_y)$, where d_{ua} is the private key of $\mathcal{U}_a$. Further $\mathcal{U}_a$ computes $C = E_{k_x}(ID_{ua} \| m \| k_x \| T)$. Finally, $\mathcal{U}_a$ sends (C, R, T) tuple to $\mathcal{U}_b$.

12.2.1.3 Verification Phase

Upon receiving (C, R, T), $\mathcal{U}_b$ uses his private key d_{ub} to compute $K = d_{ub} \times R = (k_x, k_y)$ then decrypts C using k_x to obtain $(ID_{ua} \| m \| k_x \| T)$. Further, it verifies whether T is valid or not. If T is valid then $\mathcal{U}_b$ verifies k_x, if both T and k_x are valid then $\mathcal{U}_b$ consider the message is from legitimate user $\mathcal{U}_a$.

12.2.2 Yang et al.'s e-payment System

In this subsection, we review Yang et al.'s proposed e-payment system. The e-payment system involves three participants a legal user/customer $\mathcal{U}$, the merchant $\mathcal{M}$ and the bank $\mathcal{B}$. Yang et al.'s scheme consists of following five phases:

12.2.2.1 Initialization Phase

In this phase, the system's public parameters are initialized. This phase is analogous to subsection 12.2.1.1, where $E_p(a,b)$, $E_k(.)$, $D_k(.)$ and base point P are defined and published. Further, $\mathcal{U}$ selects his private key d_u and computes his public key $Y_u = d_u \times P$. Similarly, $\mathcal{M}$ and $\mathcal{B}$ choose their private keys d_m and d_b, and compute their public keys $Y_m = d_m \times P$ and $Y_b = d_b \times P$. Finally, all the participants publish their public keys and keep their private keys secret.

12.2.2.2 Buying Phase

$\mathcal{U}$ initiates the buying phase by first selecting some electronic goods. $\mathcal{U}$ downloads the electronic goods information GI from $\mathcal{M}$'s website then $\mathcal{U}$ selects a random number $r \in Z_q$ and computes $R = r \times Y_u$, $\overline{R} = r \times Y_b$ and $K = d_u \times \overline{R} = (k_x, k_y)$, where k_x is x coordinate of K, while k_y is y coordinate of K. Then $\mathcal{U}$ accumulates the goods payment $p = \sum_{i=1}^{l} price_i$ and computes the payment information as $m = H(GI \| p \| ID_b)$. $\mathcal{U}$ computes $C_1 = E_{k_x}(ID_a \| m \| p \| k_x \| T_1)$ by using k_x. Finally, $\mathcal{U}$ sends (C_1, R, T_1) to $\mathcal{B}$, where T_1 is current timestamp.

12.2.2.3 Paying Phase

Upon receiving (C_1, R, T_1) from $\mathcal{U}$, $\mathcal{B}$ computes $K = d_b \times R = (k_x, k_y)$. Then $\mathcal{B}$ uses k_x to decrypt C_1. After decryption $\mathcal{B}$ obtains $(ID_u \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$. $\mathcal{B}$ further verifies whether T_1 and k_x are valid, if any of these is invalid $\mathcal{B}$ aborts the session. Otherwise, $\mathcal{B}$ deducts amount p from $\mathcal{U}$'s account and deposit p into a temporary account. $\mathcal{B}$ further generates the expiry date E and computes $M = m \| E$. Then $\mathcal{B}$ generates his digital signature (DS) by using his private key d_b and M and stores $\{DS, M\}$ in his database. $\mathcal{B}$ uses k_x and current timestamp T_2 to compute $C_2 = E_{k_x}(DS \| E \| k_x \| T_2)$. Finally, $\mathcal{B}$ sends (C_2, T_2) to $\mathcal{U}$. Upon receiving (C_2, T_2), $\mathcal{U}$ decrypts C_2 by using k_x and gets $(DS \| E \| k_x \| T_2) = D_{k_x}(C_2)$.

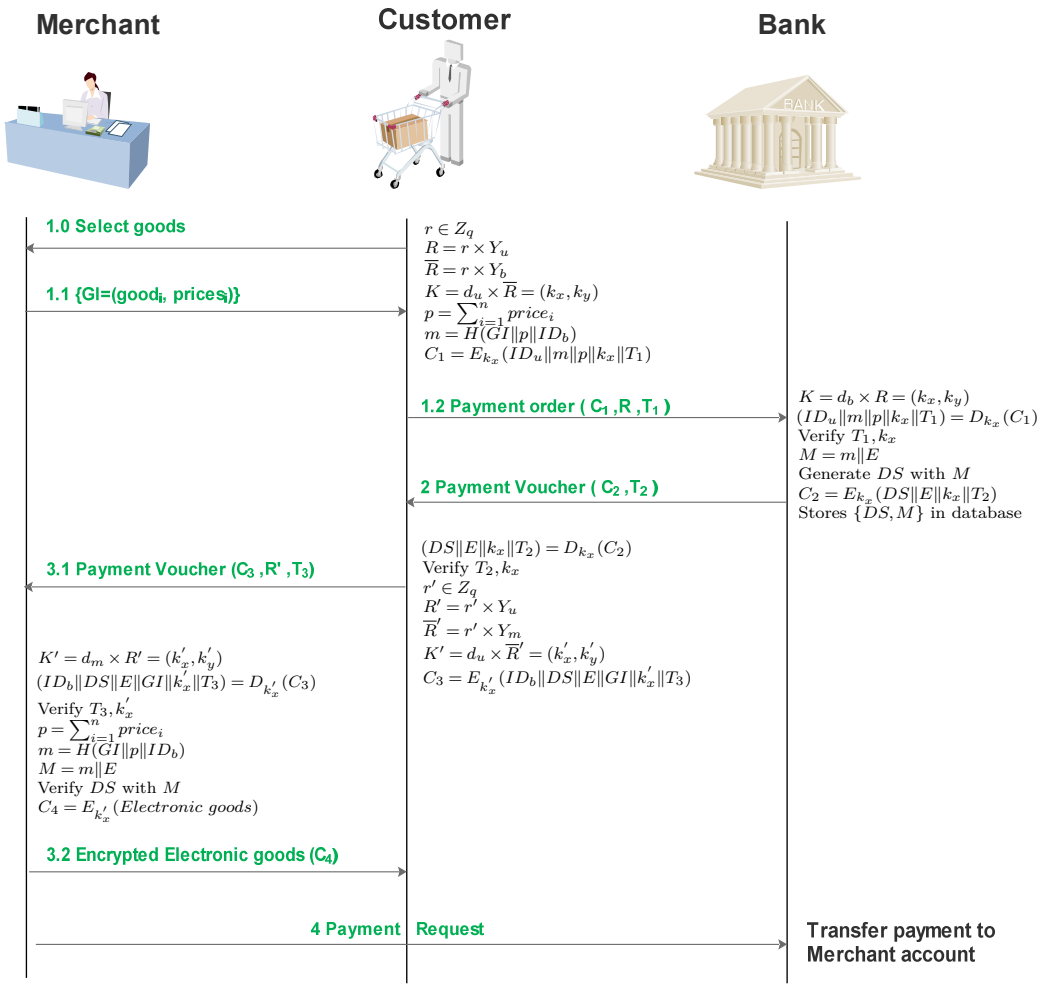


Figure 12.3: Yang et al.'s e-payment System

Then $\mathcal{U}$ verifies the validity of k_x and T_2 if any of these is invalid the session is terminated by $\mathcal{U}$. Otherwise, $\mathcal{U}$ accepts the digital signature DS.

12.2.2.4 Exchanging Phase

Initially, $\mathcal{U}$ selects a random number $r' \in Z_q$ and computes $R' = r' \times Y_u$, $\overline{R}' = r' \times Y_m$ and $K' = d_u \times \overline{R}' = (k'_x, k'_y)$. Finally, $\mathcal{U}$ using DS as payment proof computes $C_3 = E_{k'_x}(ID_b \| DS \| E \| GI \| k'_x \| T_3)$ and sends (C_3, R', T_3) to $\mathcal{M}$. Upon receiving (C_3, R', T_3), $\mathcal{M}$ computes $K' = d_m \times R' = (k'_x, k'_y)$. Then $\mathcal{M}$ uses k'_x to decrypt C_3 and obtains $(ID_b \| DS \| E \| GI \| k'_x \| T_3) = D_{k'_x}(C_3)$. $\mathcal{M}$ verifies k'_x and T_2 and aborts the session if any of these is invalid. Otherwise, $\mathcal{M}$ computes goods prices $p = \sum_{i=1}^{l} price_i$, $m = H(GI \| p \| ID_b)$ and $M = m \| E$. $\mathcal{M}$ further verifies DS with M, if digital signature DS proves to be valid. $\mathcal{M}$ encrypts electronic goods as $C_4 = E_{k'_x}(Electronic\ goods)$ and sends C_4 to $\mathcal{U}$. Finally, $\mathcal{U}$

decrypts C_4 to get desired electronic goods.

12.2.2.5 Transferring Phase

$\mathcal{M}$ sends the payment voucher to $\mathcal{B}$ before expiry date, if $\mathcal{U}$ does not receive the goods, he can ask $\mathcal{B}$ to stop the payment. Otherwise, $\mathcal{B}$ transfers the payment to $\mathcal{M}$'s account from temporary account and deletes $\{DS, M\}$ from his database.

12.3 Cryptanalysis of Yang et al.'s Schemes

This section indicates that signcryption scheme and e-payment system by Yang et al. are vulnerable to impersonation attack. We show that an adversary $\mathcal{A}$ can easily masquerade as a legitimate user by just knowing the public key of the receiver. Before proceeding further, some common assumptions are made as follows:

- $\mathcal{A}$ is having full control over communication channel, $\mathcal{A}$ can intercept, modify, insert or delete any message.
- $\mathcal{A}$ is having access to identities and public keys of communicating parties.

12.3.1 Impersonation Attack on Signcryption

Let $\mathcal{U}_a$ and $\mathcal{U}_b$ are the two legal users and $\mathcal{A}$ be the adversary. $\mathcal{A}$ will perform following steps in order to masquerade $\mathcal{U}_a$ to deceive the receiver $\mathcal{U}_b$.

Step 1: $\mathcal{A}$ computes following:

$$R = P \tag{12.1}$$

$$K = Y_{ub} = (k_x, k_y) \tag{12.2}$$

Step 2: Then $\mathcal{A}$ encrypts the message m along with ID_{ua}, k_x and T, as follows:

$$C = E_{k_x}(ID_{ua} \| m \| k_x \| T) \tag{12.3}$$

Step 3: $\mathcal{A}$ further sends (C, R, T) tuple to $\mathcal{U}_b$.

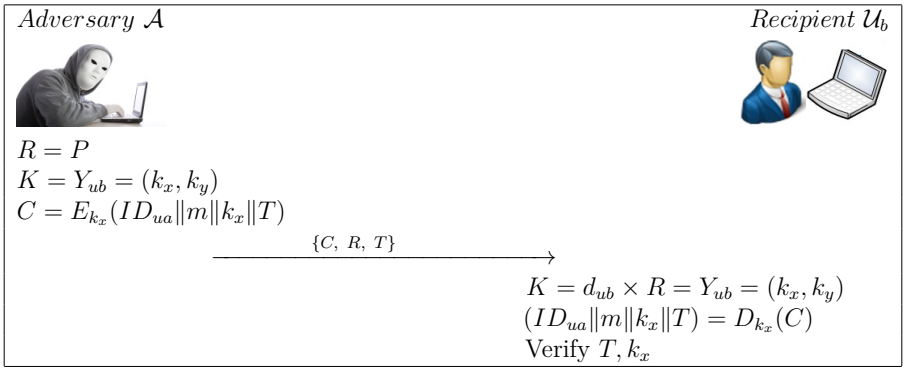


Figure 12.4: Impersonation Attack on Yang et al.'s Signcryption Scheme

Step 4: $\mathcal{U}_b$ upon receiving the tuple (C, R, T), computes $K = (k_x, k_y)$ by using his private key d_{ub} as follows:

$$K = d_{ub} \times R = d_{ub} \times P = Y_{ub} = (k_x, k_y) \tag{12.4}$$

Step 5: Then $\mathcal{U}_b$ decrypts C by using k_x as follows:

$$(ID_{ua} \| m \| k_x \| T) = D_{k_x}(C) \tag{12.5}$$

Step 6: $\mathcal{U}_b$ verifies the timestamp T, then checks k_x (decryption key) with k_x received within decrypted message. If both T and k_x are same $\mathcal{U}_b$ perceives $\mathcal{A}$ as the legitimate $\mathcal{U}_a$.

12.3.2 Impersonation Attack on E-payment System

Let $\mathcal{U}$ be a legal user, $\mathcal{B}$ be the bank, $\mathcal{M}$ a merchant and $\mathcal{A}$ be the adversary. $\mathcal{A}$ will perform following steps in-order to masquerade $\mathcal{U}$ to deceive the bank $\mathcal{B}$ and merchant $\mathcal{M}$ for fraudulent purchase of electronic goods.

Step 1: $\mathcal{A}$ selects and downloads the goods information GI from $\mathcal{M}$'s website and computes following:

$$R = P \tag{12.6}$$

$$K = Y_b = (k_x, k_y) \tag{12.7}$$

$$C_1 = E_{k_x}(ID_u \| m \| k_x \| T_1) \tag{12.8}$$

Step 2: $\mathcal{A}$ sends $\{C_1,\ R,\ T_1\}$ to $\mathcal{B}$, where T_1 is current timestamp.

Step 3: Upon receiving $\{C_1,\ R,\ T_1\}$, $\mathcal{B}$ computes following:

$$K = d_b \times R = d_b \times P = Y_b = (k_x, k_y) \tag{12.9}$$

$$(ID_u \| m \| k_x \| T) = D_{k_x}(C_1) \tag{12.10}$$

Step 4: $\mathcal{B}$ verifies the correctness of T_1 and k_x after performing decryption, if both T_1 and k_x are correct, $\mathcal{B}$ generates the expiry date E and $M = m \| E$. Then $\mathcal{B}$ computes digital signature DS with M and computes:

$$C_2 = E_{k_x}(DS \| E \| k_x \| T_2) \tag{12.11}$$

Step 5: $\mathcal{B}$ deducts money from $\mathcal{U}$'s account and stores $\{DS, M\}$ in his database. Finally, $\mathcal{B}$ sends $\{C_2, T_2\}$ to $\mathcal{U}$, where T_2 is fresh timestamp.

Step 6: $\mathcal{A}$ intercepts the message and use the same key k_x to compute:

$$(DS \| E \| k_x \| T_2) = D_{k_x}(C_2) \tag{12.12}$$

Step 7: $\mathcal{A}$ verifies T_2 and k_x, then computes:

$$R' = P \tag{12.13}$$

$$K' = Y_m = (k_x^{'}, k_y^{'}) \tag{12.14}$$

$$C_3 = E_{k_x'}(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) \tag{12.15}$$

Step 8: $\mathcal{A}$ sends $\{C_3, R', T_3\}$ to $\mathcal{M}$, where T_3 is fresh timestamp.

Step 9: Upon receiving $\{C_3, R', T_3\}$, $\mathcal{M}$ computes following:

$$K' = d_m \times R' = (k_x^{'}, k_y^{'}) \tag{12.16}$$

$$(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) = D_{k_x'}(C_3) \tag{12.17}$$

Step 10: $\mathcal{M}$ verifies the validity of $k_x^{'}$ and T_3, computes following if both are correct.

$$p = \sum_{i=1}^{n} price_i \tag{12.18}$$

$$m = H(GI \| p \| ID_b) \tag{12.19}$$

$$M = m \| E \tag{12.20}$$

Step 11: Further, $\mathcal{M}$ computes digital signature DS based on M and checks it's validity by comparing it to the DS obtained in eq. 12.17, if it is valid then $\mathcal{M}$ computes:

$$C_4 = E_{k'_x}(Electronic\ goods) \tag{12.21}$$

Step 12: Finally, $\mathcal{M}$ sends encrypted electronic goods C_4 to $\mathcal{U}$.

Step 13: $\mathcal{A}$ intercepts the message and retrieves $Electronic\ goods = D_{k'_x}(C_4)$.

12.3.3 Discussion on Security Weakness of Yang et al.'s E-payment Scheme

To understand the impact of weakness of Yang et al.'s e-payment scheme, we take an example. Let Bob is an e-payment user with an account in Bank $\mathcal{B}$, he has also initiated his private key and linked his public key with his account. It is well understood that public keys and identities are accessible to any one in the system. Let Alice be an adversary who wants to purchase electronic goods on behalf of Bob. He can impersonate by following the method as described earlier in subsection 12.3.2 to deceive bank $\mathcal{B}$ and merchant $\mathcal{M}$.

Alice initially visits $\mathcal{M}$'s website, then selects and downloads the goods and bill information. Alice generates (C_1, R, T_1) tuple as in Eqs. 12.6, 12.7, 12.8. Alice sends payment order (C_1, R, T_1) to $\mathcal{B}$.

The bank $\mathcal{B}$ upon receiving payment order computes K and C_1, then $\mathcal{B}$ verifies correctness of K_x and T_1 and finds both valid, so $\mathcal{B}$ deducts bill amount from Bob's account and store it in some temporary account. $\mathcal{B}$ computes and sends payment voucher to Alice (apparently Bob). Alice then computes (C_3, R', T_3) as in Eqs. 12.13,12.14,12.15, and sends it to the merchant $\mathcal{M}$.

$\mathcal{M}$ upon reception, computes K' and decrypts C_3 as in Eqs. 12.16, 12.17. Then $\mathcal{M}$ verifies T_3 and k'_x and finds that both are correct. $\mathcal{M}$ sends electronic goods to Alice (apparently

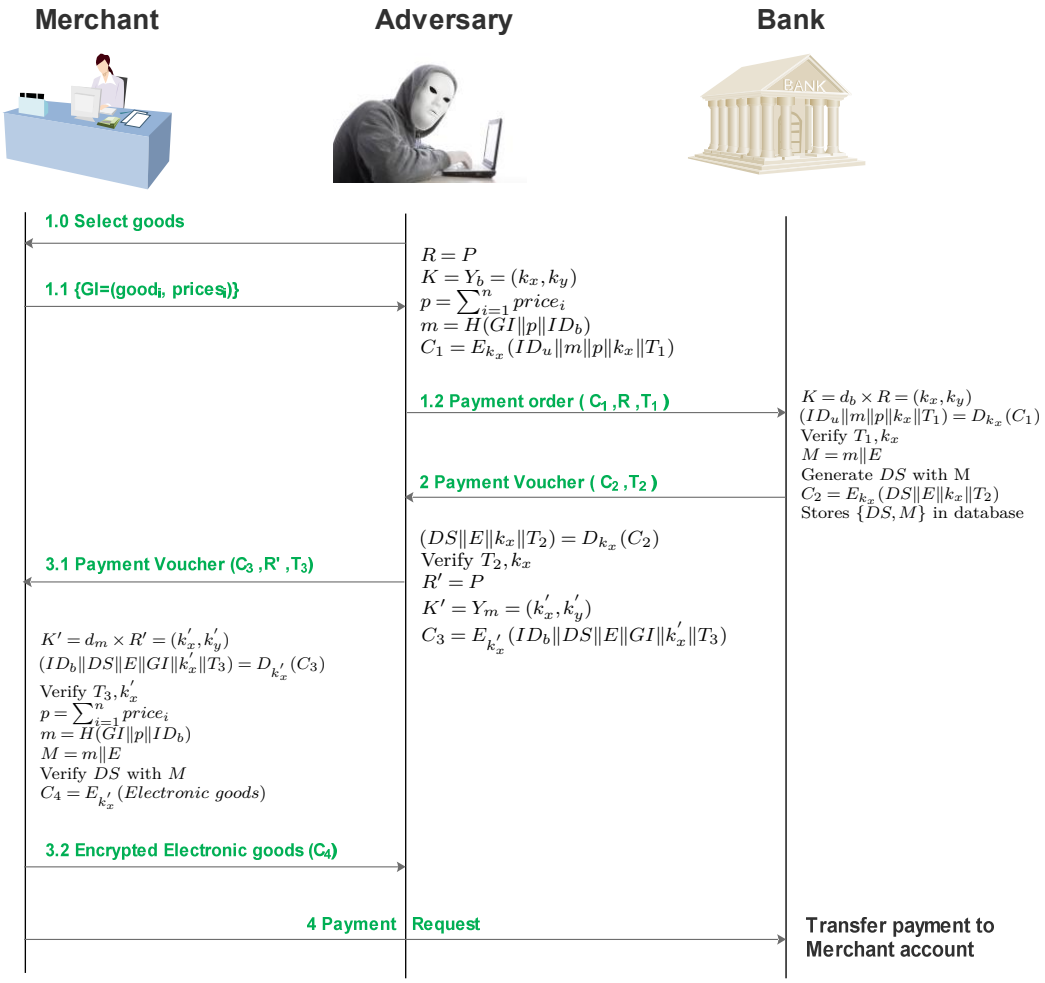


Figure 12.5: Impersonation Attack on Yang et al.'s e-payment System

Bob). Finally $\mathcal{M}$ sends received payment voucher to the bank $\mathcal{B}$. The bank transfers the billed amount to $\mathcal{M}$'s account. Hence Alice has purchased electronic goods on behalf of Bob.

12.4 Proposed Signcryption scheme and E-payment system

In following subsections, we describe the proposed signcryption scheme and e-payment system based on proposed signcryption scheme.

12.4.1 Proposed Signcryption Scheme

It can be easily verified that the security weakness present in Yang et al.'s scheme was due to the design of R and K, so we just improve the calculations of both of these parameters during signcryption and verification phases, while there is no change in initialization phase. Proposed signcryption scheme is shown in Figure 12.6 and is also explained in following subsection:

12.4.1.1 Signcryption Phase

Signcryption is performed by a legal user $\mathcal{U}_a$ when he wants to send a message m to another user $\mathcal{U}_b$. $\mathcal{U}_a$ performs following steps:

Step 1: $\mathcal{U}_a$ chooses a random $r \in Z_p$ and computes $R = r(d_{ua} + T)^{-1}$ by using his private key d_{ua} and current timestamp T.

Step 2: $\mathcal{U}_a$ further computes $K = r \times Y_{ub} = (k_x, k_y)$, where (k_x, k_y) are x and y coordinates of K, respectively.

Step 3: $\mathcal{U}_a$ performs symmetric encryption to compute $C = E_{k_x}(ID_{ua} \| m \| k_x \| T)$ using k_x as common shared key and sends (C, R, T) to $\mathcal{U}_b$.

12.4.1.2 Verification Phase

During this phase user $\mathcal{U}_b$ receives (C, R, T), decrypts and verifies that the message is sent by another legitimate user $\mathcal{U}_a$. For verification $\mathcal{U}_b$ performs following steps:

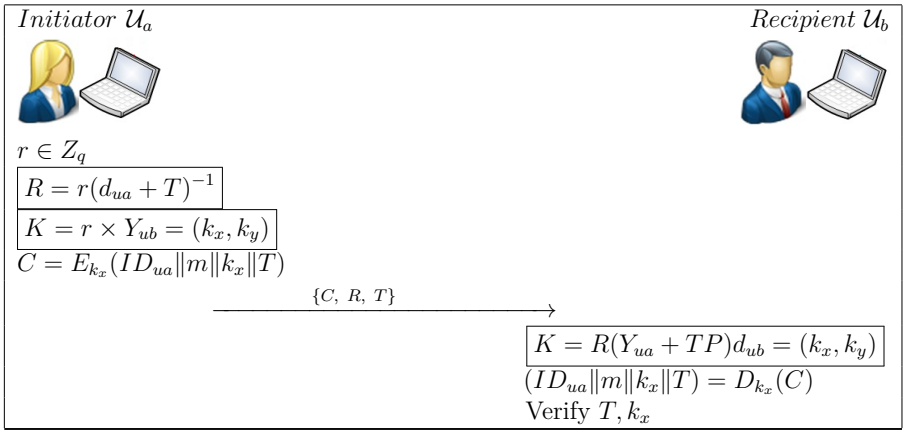


Figure 12.6: Proposed Signcryption Scheme

Step 1: $\mathcal{U}_b$ computes $K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y)$ and gets decryption key k_x.

Step 2: $\mathcal{U}_b$ decrypts C using k_x as key to obtain $(ID_{ua} \| m \| k_x \| T)$.

Step 3: $\mathcal{U}_b$ verifies whether the received T and computed k_x are same as they are present in decrypted message, if both are same then surefire it came from real $\mathcal{U}_a$.

12.4.2 The Improved e-payment using Proposed Scheme

As proved in in subsection 12.3.1 and 12.3.2, Yang et al.'s scheme is vulnerable to impersonation attack, hence unsuitable for e-payment system, e-voting and similar applications. We have also improved Yang et al.'s signcryption scheme to work in e-payment systems. The improved e-payment is shown in Figure 12.7. The e-payment system is based on proposed signcryption scheme and is consisting of five phases: (1)initialization; (2)buying; (3)paying; (4)exchange; and (5)transferring phases. The detail is as follows:

12.4.2.1 Initialization Phase

In this phase, the system sets and publishes the public parameters $E_p(a, b), E_k(.), D_k(.), P$, similar to Yang et al.'s scheme as mentioned in subsection 12.2.1.1. Each participant 'i' selects his private key d_i then computes and publishes his public key $Y_i = d_i \times P$.

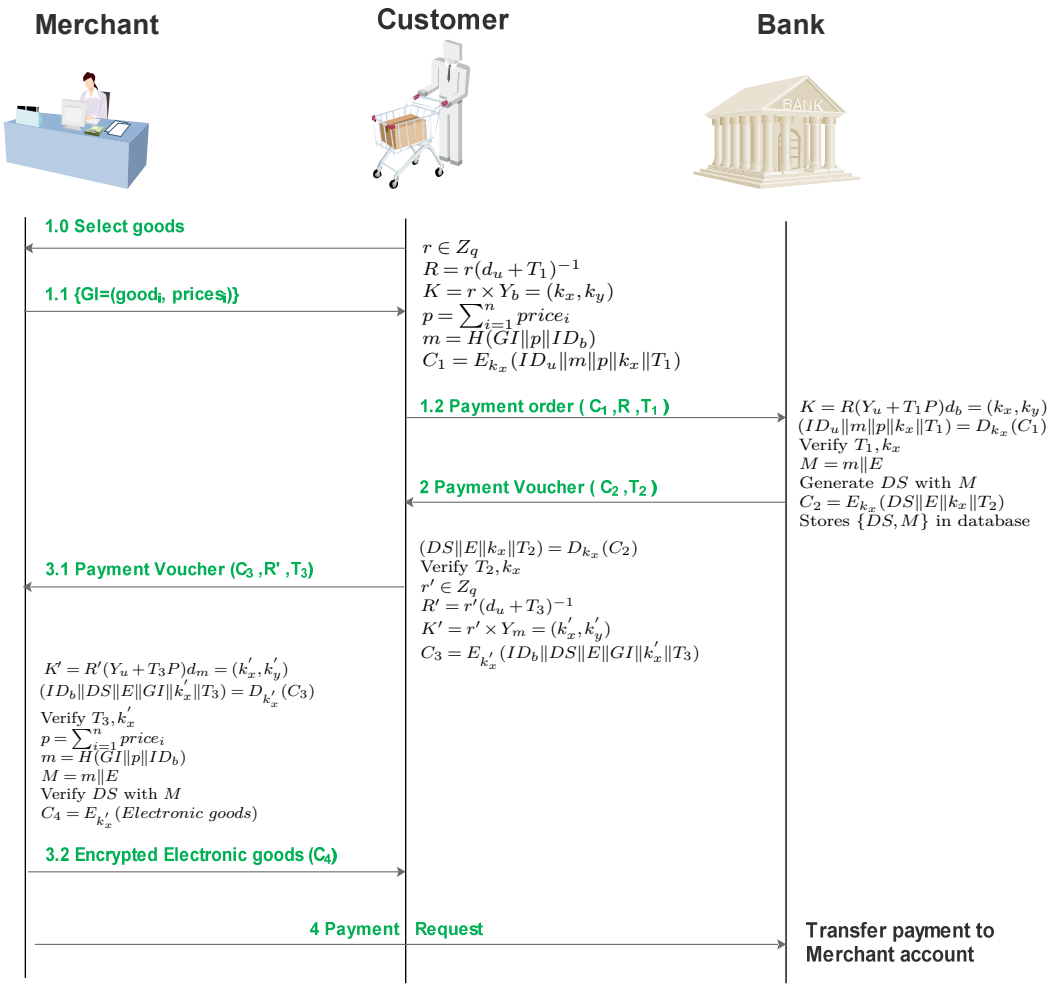


Figure 12.7: Proposed e-payment system

12.4.2.2 Buying Phase

This phase starts when a legal user $\mathcal{U}$ wants to purchase some electronic goods. Initially, $\mathcal{U}$ downloads GI (the goods information) from merchant $\mathcal{M}$'s website. Then $\mathcal{U}$ selects a random number $r \in Z_q$, and computes $R = r(d_u + T_1)^{-1}$ and $K = r \times Y_b$. Further, $\mathcal{U}$ accumulates the goods price $p = \sum_{i=1}^{n} price_i$ and generates payment text $m = H(GI \| p \| ID_b)$. Finally, $\mathcal{U}$ generates $C_1 = E_{k_x}(ID_u \| m \| p \| k_x \| T_1)$ and sends the tuple (C_1, R, T_1) to the bank $\mathcal{B}$.

12.4.2.3 Paying Phase

Upon receiving the authenticated encrypted message tuple (C_1, R, T_1), the bank $\mathcal{B}$ first computes $K = R(Y_u + T_1 P)d_b = (k_x, k_y)$ then use k_x to compute $(ID_u \| m \| p \| k_x \| T_1) = D_{k_x}(C_1)$. Further, $\mathcal{B}$ checks the validity of timestamp T_1 and verifies whether k_x is same as found after

decryption of C_1. $\mathcal{B}$ accepts the message if both T_1 and k_x are valid. Otherwise, rejects the message. Further, $\mathcal{B}$ deducts the money amounting P from $\mathcal{U}$'s account and transfer p in to a temporary account. $\mathcal{B}$ selects an expiry date E and computes $M = m\|E$. Further, $\mathcal{B}$ creates M's digital signature DS based on elliptic curve cryptography as mentioned in [213]. Finally, $\mathcal{B}$ computes and sends $C_2 = E_{k_x}(DS\|E\|k_x\|T_2)$ to $\mathcal{U}$ and stores $\{DS, M\}$ in his database.

12.4.2.4 Exchange Phase

The exchange phase consists of following three steps:

Step 1: $\mathcal{U}$ after receiving encrypted message, first decrypts C_2 to obtain DS and expiry date E. Then $\mathcal{U}$ selects $r' \in Z_q$, and computes $R' = r'(d_s + T_3)^{-1}$, $K^{'} = r' \times Y_m = (k_x^{'}, k_y^{'})$, $C_3 = E_{k_x}(ID_b\|DS\|E\|GI\|k_x^{'}\|T_3)$. Finally, $\mathcal{U}$ sends (C_3, R', T_3) to $\mathcal{M}$.

Step 2: Upon Receiving (C_3, R', T_3), $\mathcal{M}$ computes $K^{'} = R'(Y_u + T_3P)d_m = (k_x^{'}, k_y^{'})$, then decrypts C_3 using $k_x^{'}$ as decryption key. Then $\mathcal{M}$ verifies validity of T_3 and $k_x^{'}$, if both are valid, $\mathcal{M}$ computes p and $m = H(GI\|p\|ID_b)$. Further $\mathcal{M}$ calculates $M = m\|E$ and verifies the signature DS by using $\mathcal{B}$'s public key, if DS is not valid $\mathcal{M}$ aborts the session. Otherwise, $\mathcal{M}$ computes and sends $C_4 = E_{k_x^{'}}(Electronic\ goods)$ to $\mathcal{U}$.

Step 3: $\mathcal{U}$ decrypts C_4 to acquire electronic goods.

12.4.2.5 Transferring Phase

$\mathcal{M}$ sends the payment proof to $\mathcal{B}$ before expiry date. $\mathcal{U}$ is having the facility to ask $\mathcal{B}$ to terminate the transaction if he does not receive the goods, in that case $\mathcal{B}$ transfers back the money from temporary account to $\mathcal{U}$'s account. After expiry date $\mathcal{B}$ transfer the money to $\mathcal{M}$'s account and removes $\{DS, M\}$ from his database.

12.4.2.6 Dispute Resolution Phase

If user does not get the desired product or merchant fails to get the correct payment voucher then they can initiate dispute resolution phase. A Trusted Third Party (TTP) is responsible for dispute resolution, in either cases TTP will be given the merchant's private key to verify the correctness of key $k_x^{'}$. TTP after getting message $\{C_3, R', T_3\}$ can verify legality of

customer by computing following:

$$K' = R'(Y_u + T_3P)d_m = (k_x^{'}, k_y^{'}) \tag{12.22}$$

$$(ID_b \| DS \| E \| GI \| k_x^{'} \| T_3) = D_{k_x^{'}}(C_3) \tag{12.23}$$

TTP compares T_3 received in plain text and got after decryption. Similarly, TTP compares $k_x^{'}$ computed in eq. 12.22 and decrypted in eq. 12.23, if both are equal the customer and merchant both are legal. TTP can further verify the encrypted digital signature DS and product's information. Hence, TTP can resolve the dispute among both customer and merchant.

12.5 Security Analysis

This section briefly describes the security analysis of our proposed schemes. The improved schemes satisfies all the security requirements mentioned by Yang et al. It is shown that the proposed schemes remain robust even if an adversary intercepts the messages among sender $\mathcal{U}_a$ and receiver $\mathcal{U}_b$. The security of the proposed scheme relies on encryption/decryption key k_x, to generate valid k_x, the adversary $\mathcal{A}$ has to generate valid R. The detailed security analysis is described in following subsections:

12.5.1 Replay Attack

The adversary $\mathcal{A}$ can replay a past message tuple (C, R, T) as it is to receiver $\mathcal{U}_b$, when $\mathcal{U}_b$ will receive the message, it will first check the validity of timestamp T, as T is not fresh, $\mathcal{U}_b$ will realize that message is sent by adversary $\mathcal{A}$ and will simply discard the message.

12.5.2 Outsider Attack

An outsider $\mathcal{A}$ can intercept (C, R, T) of past communication among $\mathcal{U}_a$ and $\mathcal{U}_b$. However, he cannot succeed in getting m from C as it requires decryption key k_x, which can only be computed as follows:

$$K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y) \tag{12.24}$$

$\mathcal{A}$ can easily get R, Y_{ua}, T, P but having all these values computing K is ECDLP, as $\mathcal{A}$ is not having private key d_{ub}.

12.5.3 Impersonation Attack

Impersonation attack is only possible if $\mathcal{A}$ can generate valid R and K pair, to generate valid R, $\mathcal{A}$ needs private key of sender $\mathcal{U}_a$. If $\mathcal{A}$ tries to forge R by selecting a random number $\overline{r} \in Z_q$ and computes $\overline{R} = \overline{r}(\overline{d}_{ua} + T)^{-1}$, $\overline{K} = \overline{r} \times Y_{ub}$, then after receiving $(C, \overline{R}, T)$ tuple, $\mathcal{U}_b$ will compute $K = \overline{R}(Y_{ua} + TP)d_{ub}$, which will not be equal to $\overline{r} \times Y_{ub}$. Hence, $\mathcal{U}_b$ will be aware that message is sent by $\mathcal{A}$.

12.5.4 Server Spoofing Attack

$\mathcal{A}$ can pretend to be a bank server if he can generate $\overline{C_2} = E_{k_x}(\overline{DS}\|\overline{E}\|k_x\|T_2)$ and send $(\overline{C_2}, T_2)$ to $\mathcal{U}_a$. However $\mathcal{A}$ has to obtain d_b to compute $K = R(Y_{ua} + TP)d_b = (k_x, k_y)$ in order to get correct k_x, which is infeasible.

12.5.5 Man-in-middle Attack

If $\mathcal{A}$ intercepts the payment information message (C_i, R, T_i) and then replace the timestamp T_i with fresh timestamp T_{fresh}, $\mathcal{U}_b$ after decrypting C_i will compare T_{fresh} with timestamp T_i got after decryption, if both are not same, $\mathcal{U}_b$ will terminate the session. Henceforth, man in middle attack is not viable on proposed schemes.

12.5.6 ID Theft Attack

The proposed schemes make use of private and public keys of sender and receiver to generate and verify authenticated message. So if the identity of any or both parties is/are revealed to the adversary. It will have no effect on security of the scheme.

12.5.7 Confidentiality

The confidentiality can be broken if $\mathcal{A}$ can decrypt the cipher text C, in proposed scheme all the messages are encrypted by using a symmetric key k_x, it has already been proved in subsection 12.5.2 that k_x can only be computed by first getting d_{ub} from eq. 12.24, which is an ECDLP. Hence, it is not feasible.

12.5.8 Authenticity

Proposed schemes ensure the sender's authenticity as receiver extracts k_x by computing $K = R(Y_{ua} + TP)d_{ub} = (k_x, k_y)$, which require $\mathcal{U}_a$'s public key and $\mathcal{U}_b$'s private key, further $\mathcal{U}_b$ verifies the validity of k_x after decryption of C and compares computed k_x and decrypted k_x from cipher text C.

12.5.9 Integrity

Proposed schemes provide message integrity as if any of the parameter (C, R, T) is modified then receiver $\mathcal{U}_b$ will not be able to verify validity of k_x or T and will simply terminate the session.

12.5.10 Privacy Protection

ID of all the participants are sent in cipher text C, no ID is sent in plain text over public network. Similarly, user $\mathcal{U}$ sends GI (goods information) to bank after protecting it by oneway hash function $m = H(GI \| p \| ID_b)$. Furthermore, the digital signature DS does not reveal payer's information. Hence, buying privacy is protected in proposed scheme.

12.5.11 Non-repudiation

In proposed e-payment scheme, none of the participant can deny the transaction, as trusted third party can check the validity of messages between both customer and merchant as described in subsection 12.4.2.6.

12.5.12 Double Spending Prevention

The bank keeps $\{DS, M\}$ information in database until payment is transferred to merchant's account. Once payment is transferred to merchant's account, the bank $\mathcal{B}$ deletes the corresponding $\{DS, M\}$ entry. Therefore, $\{DS, M\}$ can be used only once. Hence, the proposed scheme prevents double spending of same payment voucher.

Table 12.2: Security Analysis

Scheme→ Security Properties↓	Yang et al.	Proposed
Resistance to Replay attack	Yes	Yes
Resistance to Outsider attack	Yes	Yes
Resistance to Impersonation attack	No	Yes
Resistance to Server spoofing attack	Yes	Yes
Resistance to Man-in-middle attack	No	Yes
Resistance to ID theft attack	No	Yes
Confidentiality	Yes	Yes
Authenticity	No	Yes
Integrity	Yes	Yes
Privacy protection	Yes	Yes
Non-repudiation	No	Yes
Double spending prevention	Yes	Yes

12.6 Protocol Verification using ProVerif

For Verification purpose, we model the whole protocol steps according to each participant (User, Merchant, Bank) in ProVerif. Then we check the secrecy of the session key and the reachability property as shown in Fig. 12.8. Finally, we got the results as follows:

```
RESULT inj event(endMerchant(id)) ==> inj event(beginMerchant(id)) is true.
RESULT inj event(endBank(id_2234)) ==> inj event(beginBank(id_2234)) is true.
RESULT inj event(endUser(id_4043)) ==> inj event(beginUser(id_4043)) is true.
RESULT not attacker(K[]) is true.
RESULT not attacker(K1[]) is true.
```

The results shows that all the three processes started and terminated successfully. While *not attacker* on both K and $K1$ shows that (1) secrecy of K and $K1$ is true against attacks (2) authentication is satisfied among user and bank as well as between user and merchant.

12.7 Performance Analysis

In this section, we evaluate the performance of proposed scheme by comparing it with Yang et al's scheme [13], before proceeding further, we define some notations as follows:

- T_{pm}: Time for elliptic curve point multiplication
- T_{sy}: Time for symmetric encryption/ decryption operation
- T_h: Time for oneway hash function

Table 12.3 illustrates the performance comparison of proposed scheme with Yang et al.'s scheme. The computation time of different cryptographic operations mentioned by Farash [214] are as follows: T_{pm} and T_{sy} takes $0.86ms$ and $0.001ms$ respectively while time for T_h is negligible.

```
(************** Channels **************)
free ch1:channel.     (*  U to B  *)
free ch2:channel.     (*  U to M  *)
(********** Names & Variables **********)
const p: bitstring.
const P: bitstring.
free GI: bitstring.
free Db: bitstring [private].
free Du: bitstring [private].
free Dm: bitstring [private].
const IDu: bitstring.
const IDb: bitstring.
const IDm: bitstring.
(**Constructors*destructors*Equations **)
fun consset(bitstring,bitstring):bitstring.
fun add(bitstring,bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun syme(bitstring,bitstring):bitstring.(*
    encryption*)
fun inverse(bitstring):bitstring.
fun getx(bitstring):bitstring.(*get x
    coordinate*)
fun sig(bitstring):bitstring.(*signature*)
fun h(bitstring):bitstring.(*hash*)
reduc forall m:bitstring,key:bitstring;
symd(syme(m,key),key)=m.(*decryption*)
equation forall a:bitstring; inverse(inverse(a)
    )=a.
event beginUser(bitstring).
event endUser(bitstring).
event beginBank(bitstring).
event endBank(bitstring).
event beginMerchant(bitstring).
event endMerchant(bitstring).
```

(a) Declarations

```
(********Process Replication***********)
process ( (!pUser) | (!pBank) | (!pMerchant) )
(************** *queries* **************)
free K1:bitstring [private].
free K:bitstring [private].
query attacker(K1).
query attacker(K).
query id:bitstring;inj_event(endUser(id))==>
    inj_event(beginUser(id))
query id:bitstring;inj_event(endBank(id))==>
    inj_event(beginBank(id))
query id:bitstring;inj_event(endMerchant(id))
    ==>inj_event(beginMerchant(id)).
```

(c) Main

```
(*************User Process *************)
let pUser=
let Yu=mult(Du,P) in
out(ch1,(Yu));
out(ch2,(Yu));
in(ch1,(XYb:bitstring));
in(ch2,(XYm:bitstring));
new r:bitstring;
new T1: bitstring;
event beginUser(IDu);
let R=mult(r,inverse(add(Du,T1))) in
let K=mult(r,XYb) in
let m=h(consset(GI,consset(p,IDb))) in
let C1=syme((IDu,m,p,getx(K),T1),getx(K)) in
out(ch1,(C1,R,T1)); (*To bank*)
in(ch1,(XC2:bitstring,XT2:bitstring));
let (XDs:bitstring,XE:bitstring,XXkx:bitstring,
    XXT2:bitstring) = symd(XC2,getx(K)) in
if(XT2=XXT2) then
if(getx(K)=XXkx) then
new r1: bitstring;
new T3: bitstring;
let R1=mult(r1,inverse(add(Du,T3))) in
let K1=mult(r1,XYm) in
let C3=syme((IDb,XDs,XE,GI,getx(K1),T3),getx(K1
    )) in
out(ch2,(C3,R1,T3)); (*To merchant*)
event endUser(IDu).
(************** Bank ****************)
let pBank=
let Yb= mult(Db,P) in
in(ch1,(XYu:bitstring));
out(ch1,(Yb));
in(ch1,(XC1:bitstring,XR:bitstring,XT1:
    bitstring));
event beginBank(IDb);
let K=mult(mult(XR,add(XYu,mult(XT1,P))),Db) in
let (=IDu, Xm:bitstring, Xp:bitstring, Xkx:
    bitstring, XXT1:bitstring) = symd(XC1,getx
    (K)) in
if(getx(K)=Xkx) then
if(XT1=XXT1) then
new E: bitstring;
new T2: bitstring;
let M=consset(Xm,E) in
let DS=sig(M) in
let C2=syme((DS,E,Xkx,T2),Xkx) in
out(ch1,(C2,T2));
event endBank(IDb).
(************** Merchant ****************)
let pMerchant=
in(ch2,(XYu:bitstring));
let Ym= mult(Dm,P)in
out(ch2,(Ym));
in(ch2,(XC3:bitstring,XR1:bitstring,XT3:
    bitstring));
event beginMerchant(IDm);
let K1=mult(mult(XR1,add(XYu,mult(XT3,P))),Dm)
    in
let (=IDb, XDs:bitstring, XE:bitstring,XGI:
    bitstring,
Xkx:bitstring,XXT3:bitstring)=symd(XC3,getx(K1)
    ) in
if(XT3=XXT3) then
if(getx(K1)= Xkx) then
let m=h(consset(GI,consset(p,IDb))) in
let M=consset(m,XE) in
if(sig(M)=XDs) then
let C4=syme(GI,getx(K1)) in
out(ch2, (C4));
event endMerchant(IDm).
```

(b) Processes

Figure 12.8: ProVerif Validation

Table 12.3: Computation Cost Analysis

Scheme→ Participant↓	Yang et al.	Proposed
$User\ \mathcal{U}_A$	$6T_{pm} + 3T_{sy} = 5.163ms$	$2T_{pm} + 3T_{sy} = 1.723ms$
$Bank\ \mathcal{B}$	$1T_{pm} + 2T_{sy} = 0.862ms$	$1T_{pm} + 2T_{sy} = 0.862ms$
$Merchant\ \mathcal{M}$	$1T_{pm} + 2T_{sy} = 0.862ms$	$1T_{pm} + 2T_{sy} = 0.862ms$
Total	$8T_{pm} + 7T_{sy} = 6.887ms$	$4T_{pm} + 7T_{sy} = 3.447ms$

In Yang et al.'s e-payment system the total operations performed by $\mathcal{U}$ are $6T_{pm} + 3T_{sy}$, while $\mathcal{B}$ performs $1T_{pm} + 2T_{sy}$ operations and $\mathcal{M}$ performs $1T_{pm} + 2T_{sy}$ operations. The total computation time taken by $\mathcal{U}$ is 5.163 ms, $\mathcal{B}$ and $\mathcal{M}$ takes 0.862 ms, so total time taken by all participants during execution of Yang et al.'s e-payment system is 6.887 ms. $\mathcal{U}$ in proposed scheme performs $2T_{pm} + 3T_{sy}$ operations, number of operations performed by $\mathcal{B}$ are $1T_{pm} + 2T_{sy}$, while $\mathcal{M}$ performs $1T_{pm} + 2T_{sy}$ operations. Total time taken by $\mathcal{U}$ in proposed scheme is 1.723 ms, which is roughly one third of the time taken by $\mathcal{U}$ in Yang et al.'s scheme. While $\mathcal{B}$ and $\mathcal{M}$ takes 0.862 ms, which are equal to time taken by both $\mathcal{B}$ and $\mathcal{M}$ in Yang et al.'s scheme, total time taken by all participants during execution of proposed e-payment system is 3.447 ms as shown in table 12.3. Hence in proposed scheme user $\mathcal{U}$ takes approximately 66% less computation time as compared with Yang et al.'s scheme. Therefore, proposed scheme provides more robustness against attacks and is more lightweight as compared to Yang et al.'s scheme.

12.8 Chapter Summary

In this chapter, we cryptanalyzed Yang et al.'s signcryption and e-payment schemes. We proved that both of Yang et al.'s schemes are vulnerable to impersonation attack. As a remedy, we proposed improved signcryption scheme to overcome security weaknesses of Yang et al.'s scheme. Furthermore, we also improved e-payment system of Yang et al. We have performed informal and formal verification of our improved protocol using widespread automated tool ProVerif. The proposed schemes ensured robustness against all known attacks, while reducing about 66% computation cost on user side as compared to Yang et al.'s scheme. Hence proposed schemes improved the security as well as reduced the computation overhead and is more suitable for resource constrained environments.

Chapter 13

Conclusions and Future Directions

This thesis is devoted to develop some lightweight and secure cryptographic schemes/protocols majoring in five sub areas: (1) Two-factor authentication, (2) Three-factor authentication (3) Mobile handover authentication, (4) Multi-server authentication and (5) Authenticated encryption. Chapter 1 introduced the thesis. The main emphasis is to explain the objectives, research contributions and organization of the thesis. Chapter 2 is devoted to explain some mathematical background useful to understand thesis contributions along with common adversarial model, computational hard problems, BioHashing and an introduction to the formal security model of widespread automated tool ProVerif. Chapter 3 to 12 are the main contributions of the thesis. Each of the cited chapter is devoted for a cryptographic solution. All the cryptographic solution are designed to cope with security requirements of the current technologies. The solution presented in this thesis are analyzed under random oracle model as well as under formal threat model of automated tool ProVerif. The main contributions of the thesis are as underlined:

- Four two-factor authentication and key agreement schemes/protocols. (Chapter 3 to 6) are designed for different environment. The schemes are based on ECC and symmetric key primitives. All these schemes are developed to provide confidentiality, authenticity, anonymity and non-repudiation etc. While incurring low overhead these schemes are proved to resist all the known attacks.
- Two biometric based three-factor authentication schemes/protocols (Chapter 7 to 8) are proposed for Telecare medicine-information systems, one based on ECC and other on symmetric cryptography primitives. The three-factor proposed schemes are also provably secure under random oracle model as well as under formal threat model of ProVerif.

- An authentication scheme (Chapter 9) for securing mobile handover process is proposed. The handover authentication scheme is provably secure under random oracle model and under the formal threat model of ProVerif. Proposed handover authentication scheme incurs the lowest communication and computation overhead as compared with related recent schemes.
- Two authentication schemes (Chapter 10 to 11) are proposed to secure multi-server architectures. Such architecture is quite different than single server architecture. The designed schemes does not need the intervention of registration server for each authentication request. Furthermore, the second scheme (Chapter 11) based on bilinear mapping and identity based cryptosystem also considers the adversarial model under which all the servers are not trusted. Both the schemes are provably secure under random oracle model as well as under the threat model of ProVerif, while incurring quite low overhead.
- A signcryption scheme (Chapter 12) is proposed using ECC. Furthermore, an e-payment system based on proposed signcryption scheme is also developed to secure online transactions. The proposed signcryption scheme and e-payment system are designed to provide: (1)confidentiality, (2) integrity, (3) authenticity and (4) non-repudiation. Furthermore, e-payment system also prevent double spending of same voucher. Moreover, both the schemes are robust against the known attacks. The signcryption scheme and e-payment system are also secure under the formal threat model of ProVerif.
- An investigation of the security requirements and existing protocols in internet of things (IoT) and cloud computing may be a valuable future work.

Bibliography

[1] Li X, Niu J, Kumari S, Liao J, Liang W. An enhancement of a smart card authentication scheme for multi-server architecture. *Wireless Personal Communications* 2014; :1–18.

[2] Tu H, Kumar N, Chilamkurti N, Rho S. An improved authentication protocol for session initiation protocol using smart card. *Peer-to-Peer Networking and Applications* 2014; :1–8.

[3] Farash M. Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. *Peer-to-Peer Networking and Applications* 2014; :1–10doi: 10.1007/s12083-014-0315-x. URL `http://dx.doi.org/10.1007/s12083-014-0315-x`.

[4] Huang B, Khan M, Wu L, Muhaya F, He D. An efficient remote user authentication with key agreement scheme using elliptic curve cryptography. *Wireless Personal Communications* 2015; :1–16doi: 10.1007/s11277-015-2735-1. URL `http://dx.doi.org/10.1007/s11277-015-2735-1`.

[5] Kumari S, Khan MK, Li X. An improved remote user authentication scheme with key agreement. *Computers & Electrical Engineering* 2014; **40**(6):1997–2012.

[6] Islam S, Khan M. Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. *Journal of Medical Systems* 2014; **38**(10):135, doi: 10.1007/s10916-014-0135-9. URL `http://dx.doi.org/10.1007/s10916-014-0135-9`.

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

[8] Mir O, Nikooghadam M. A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. *Wireless Personal Communica-*

tions ; :1–23.

[9] Li G, Jiang Q, Wei F, Ma C. A new privacy-aware handover authentication scheme for wireless networks. *Wireless Personal Communications* 2014; :1–9.

[10] Lu Y, Li L, Peng H, Yang Y. A biometrics and smart cards-based authentication scheme for multi-server environments. *Security and Communication Networks* 2015; :1–10doi: 10.1002/sec.1246.

[11] Lu Y, Li L, Yang X, Yang Y. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. *PloS ONE* 2015; **10**(5), doi: e0126323.doi:10.1371/journal.pone.0126323.

[12] Tsai JL, Lo NW. A privacy-aware authentication scheme for distributed mobile cloud computing services. *Systems Journal, IEEE* Sept 2015; **9**(3):805–815, doi: 10.1109/JSYST.2014.2322973.

[13] Yang JH, Chang YF, Chen YH. An efficient authenticated encryption scheme based on ecc and its application for electronic payment. *Information Technology And Control* 2013; **42**(4):315–324.

[14] William S, Stallings W. *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.

[15] Shamir A. Identity-based cryptosystems and signature schemes. *Advances in cryptology*, Springer, 1985; 47–53.

[16] Menezes AJ, Okamoto T, Vanstone S, *et al.*. Reducing elliptic curve logarithms to logarithms in a finite field. *Information Theory, IEEE Transactions on* 1993; **39**(5):1639–1646.

[17] He D, Khan MK, Kumar N. A new handover authentication protocol based on bilinear pairing functions for wireless networks. *International Journal of Ad Hoc and Ubiquitous Computing* 2015; **18**(1-2):67–74.

[18] Amin R, Biswas G. Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. *Wireless Personal Communications* 2015; :1–24.

[19] Farash MS, Attari MA. An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems. *Information Technology And Control* 2014; **43**(2):143–150.

[20] Wang C, Zhang Y. New authentication scheme for wireless body area networks using the bilinear pairing. *Journal of Medical Systems* 2015; **39**(11):1–8.

[21] Hsu CL, Chuang YH, Kuo Cl. A novel remote user authentication scheme from bilinear pairings via internet. *Wireless Personal Communications* 2015; :1–12.

[22] Pandit T, Barua R, Tripathy S. eck secure single round id-based authenticated key exchange protocols with master perfect forward secrecy. *Network and System Security*. Springer, 2014; 435–447.

[23] Zhang Y, Chen J, Huang B, Peng C. An efficient password authentication scheme using smart card based on elliptic curve cryptography. *Information Technology And Control* 2014; **43**(4):390–401.

[24] Sun HM, He BZ, Chen CM, Wu TY, Lin CH, Wang H. A provable authenticated group key agreement protocol for mobile environment. *Information Sciences* 2015; .

[25] Eisenbarth T, Kasper T, Moradi A, Paar C, Salmasizadeh M, Shalmani M. On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme. *Advances in Cryptology, CRYPTO 2008*, *Lecture Notes in Computer Science*, vol. 5157, Wagner D (ed.). Springer Berlin Heidelberg, 2008; 203–220, doi: 10.1007/978-3-540-85174-5_12. URL `http://dx.doi.org/10.1007/978-3-540-85174-5_12`.

[26] Dolev D, Yao AC. On the security of public key protocols. *Information Theory, IEEE Transactions on* Mar 1983; **29**(2):198–208, doi: 10.1109/TIT.1983.1056650.

[27] Cao X, Zhong S. Breaking a remote user authentication scheme for multi-server architecture. *Communications Letters, IEEE* Aug 2006; **10**(8):580–581, doi: 10.1109/LCOMM.2006.1665116.

[28] Kocher P, Jaffe J, Jun B. Differential power analysis. *Advances in Cryptology CRYPTO 99*, Springer, 1999; 388–397.

[29] Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. *Computers, IEEE Transactions on* 2002; **51**(5):541–552.

[30] Jin ATB, Ling DNC, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition* 2004; **37**(11):2245–2255.

[31] Lumini A, Nanni L. An improved biohashing for human authentication. *Pattern recognition* 2007; **40**(3):1057–1065.

[32] Belguechi R, Rosenberger C, Ait-Aoudia S. Biohashing for securing minutiae template. *Pattern Recognition (ICPR), 2010 20th International Conference on*, IEEE, 2010; 1168–1171.

[33] Xie Q, Dong N, Tan X, Wong DS, Wang G. Improvement of a three-party password-based key exchange protocol with formal verification. *Information Technology And Control* 2013; **42**(3):231–237.

[34] Chaudhry SA, Farash MS, Naqvi H, Kumari S, Khan MK. An enhanced privacy preserving remote user authentication scheme with provable security. *Security and Communication Networks* 2015; :1–13doi: 10.1002/sec.1299.

[35] Chaudhry SA, Farash M, Naqvi H, Sher M. A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. *Electronic Commerce Research* 2015; :1–27doi: 10.1007/s10660-015-9192-5.

[36] Xie Q, Dong N, Wong DS, Hu B. Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol. *International Journal of Communication Systems* 2014; :n/a–n/adoi: 10.1002/dac.2858.

[37] Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan Mu. An improved and provably secure privacy preserving authentication protocol for sip. *Peer-to-Peer Networking and Applications* 2015; :1–14doi: 10.1007/s12083-015-0400-9.

[38] Hu B, Xie Q, Li Y. Automatic verification of password-based authentication protocols using smart card. *Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Conference on*, vol. 1, IEEE, 2011; 34–39.

[39] Cheval V, Blanchet B. Proving more observational equivalences with proverif. *Principles of Security and Trust*. Springer, 2013; 226–246.

[40] Chang CC, Wu TC. Remote password authentication with smart cards. *Computers and Digital Techniques, IEE Proceedings E* 1991; **138**(3):165–168.

[41] Abi-Char PE, Mhamed A, El-Hassan B. A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications. *Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST'07. The 2007 International Conference on*, IEEE, 2007; 235–240.

[42] Arshad H, Nikooghadam M. An efficient and secure authentication and key agreement scheme for session initiation protocol using ecc. *Multimedia Tools and Applications* 2014; :1–17.

[43] Harn L, Lin HY. Authenticated key agreement without using one-way hash functions. *Electronics Letters* 2001; **37**(10):629–630.

[44] Liao YP, Wang SS. A new secure password authenticated key agreement scheme for sip using self-certified public keys on elliptic curves. *Computer Communications* 2010; **33**(3):372–380.

[45] Abi-Char PE, Mhamed A, El-Hassan B. A secure authenticated key agreement protocol based on elliptic curve cryptography. *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*, IEEE, 2007; 89–94.

[46] Ryu EK, Yoon EJ, Yoo KY. An efficient id-based authenticated key agreement protocol from pairings. *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*. Springer, 2004; 1458–1463.

[47] Debiao H, Jianhua C, Jin H. An id-based client authentication with key agreement protocol for mobile client–server environment on ecc with provable security. *Information Fusion* 2012; **13**(3):223–230.

[48] Xie Q. A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems* 2012; **25**(1):47–54.

[49] Islam S, Biswas G. A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Journal of Systems and Software* 2011; **84**(11):1892–1898.

[50] Chuang MC, Chen MC. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Systems with Applications* 2014; **41**(4):1411–1418.

[51] Asad M, uddin N, Chaudhry SA, Amin N. An authenticated key agreement with rekeying for secured body sensor networks based on hybrid cryptosystem. *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*, IEEE, 2012; 118–121.

[52] Mehmood Z, Nizamuddin N, Ch S, Nasar W, Ghani A. An efficient key agreement with rekeying for secured body sensor networks. *Digital Information Processing and Communications (ICDIPC), 2012 Second International Conference on*, IEEE, 2012; 164–167.

[53] Irshad A, Sher M, Faisal MS, Ghani A, Ul Hassan M, Ch SA. A secure authentication scheme for session initiation protocol by using ecc on the basis of the tang and liu scheme. *Security and Communication Networks* 2013; .

[54] Bala S, Sharma G, Verma AK. An improved forward secure elliptic curve signcryption key management scheme for wireless sensor networks. *IT Convergence and Security 2012*. Springer, 2013; 141–149.

[55] Sharma G, Bala S, Verma AK. Extending certificateless authentication for wireless sensor networks: A novel insight. *International Journal of Computer Science Issues (IJCSI)* 2013; **10**(6).

[56] Chou CH, Tsai KY, Lu CF. Two id-based authenticated schemes with key agreement for mobile environments. *The Journal of Supercomputing* 2013; **66**(2):973–988.

[57] Farash MS, Attari MA. A secure and efficient identity-based authenticated key exchange protocol for mobile client–server networks. *The Journal of Supercomputing* 2014; :1–17.

[58] Irshad A, Sher M, Rehman E, Ch SA, Hassan MU, Ghani A. A single round-trip sip authentication scheme for voice over internet protocol using smart card. *Multimedia Tools and Applications* 2013; :1–18doi: 10.1007/s11042-013-1807-z. URL `http://dx.doi.org/10.1007/s11042-013-1807-z`.

[59] Chaudhry S, Naqvi H, Shon T, Sher M, Farash M. Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. *Journal of Medical Systems* 2015; **39**(6):66, doi: 10.1007/s10916-015-0244-0.

[60] Zhao Z. An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. *Journal of Medical Systems* 2014; **38**(2):1–7.

[61] Xu X, Zhu P, Wen Q, Jin Z, Zhang H, He L. A secure and efficient authentication and key agreement scheme based on ecc for telecare medicine information systems. *Journal of Medical Systems* 2014; **38**(1):1–7.

[62] Nicanfar H, Leung VC. Multilayer consensus ecc-based password authenticated key-exchange (mcepak) protocol for smart grid system. *Smart Grid, IEEE Transactions on* 2013; **4**(1):253–264.

[63] Zhang L, Tang S, Cai Z. Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. *International Journal of Communication Systems* 2013; .

[64] Zhang L, Tang S, Cai Z. Cryptanalysis and improvement of password-authenticated key agreement for session initiation protocol using smart cards. *Security and Communication Networks* 2014; **7**(12):2405–2411, doi: 10.1002/sec.951. URL `http://dx.doi.org/10.1002/sec.951`.

[65] Farash MS. An improved password-based authentication scheme for session initiation protocol using smart cards without verification table. *International Journal of Communication Systems* 2014; doi: 10.1002/dac.2879.

[66] Bellare M, Rogaway P. Entity authentication and key distribution. *Advances in Cryptology, CRYPTO 93*, Springer, 1994; 232–249.

[67] Bellare M, Rogaway P. Provably secure session key distribution: the three party case. *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, ACM, 1995; 57–66.

[68] Abadi M, Blanchet B, Comon-Lundh H. Models and proofs of protocol security: A progress report. *Computer Aided Verification*, Springer, 2009; 35–49.

[69] Kilinc H, Yanik T. A survey of sip authentication and key agreement schemes. *Communications Surveys Tutorials, IEEE* Second 2014; **16**(2):1005–1023, doi: 10.1109/SURV.2013.091513.00050.

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

[71] Sun DZ, Huai JP, Sun JZ, Li JX, Zhang JW, Feng ZY. Improvements of juang's password-authenticated key agreement scheme using smart cards. *Industrial Electronics, IEEE Transactions on* 2009; **56**(6):2284–2291.

[72] Lu R, Lin X, Liang X, Shen X. A dynamic privacy-preserving key management scheme for location-based services in vanets. *Intelligent Transportation Systems, IEEE Transactions on* 2012; **13**(1):127–139.

[73] Zhao D, Peng H, Li L, Yang Y. A secure and effective anonymous authentication scheme for roaming service in global mobility networks. *Wireless Personal Communications* 2014; **78**(1):247–269.

[74] Lu Y, Li L, Yang Y. Robust and efficient authentication scheme for session initiation protocol. *Math. Probl. Eng* 2015; **2015**.

[75] He D. An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. *Ad Hoc Networks* 2012; **10**(6):1009–1016.

[76] Farash MS, Attari MA. An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. *International Journal of Communication Systems* 2014; doi: 10.1002/dac.2848. URL `http://dx.doi.org/10.1002/dac.2848`.

[77] Farash MS, Attari MA. Cryptanalysis and improvement of a chaotic map-based key agreement protocol using chebyshev sequence membership testing. *Nonlinear Dynamics* 2014; **76**(2):1203–1213.

[78] Jiang Q, Ma J, Tian Y. Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of zhang et al. *International Journal of Communication Systems* 2014; :n/a–n/adoi: 10.1002/dac.2767.

[79] Zhang L, Tang S, Cai Z. Robust and efficient password authenticated key agreement with user anonymity for session initiation protocol-based communications. *IET Communications* 2014; **8**(1):83–91.

[80] Tsai JL. Efficient multi-server authentication scheme based on one-way hash function without verification table. *Computers & Security* 2008; **27**(3):115–121.

[81] Lu R, Lin X, Zhu H, Liang X, Shen X. Becan: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on* 2012; **23**(1):32–43.

[82] Liao YP, Wang SS. A secure dynamic id based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces* 2009; **31**(1):24–29.

[83] Lee CC, Lin TH, Chang RX. A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards. *Expert Systems with Applications* 2011; **38**(11):13 863–13 870.

[84] Wang D, Wang P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Computer Networks* 2014; **73**:41–57.

[85] Wang D, He D, Wang P, Chu C. Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *Dependable and Secure Computing, IEEE Transactions on* 2014; **PP**(99):1–1, doi: 10.1109/TDSC.2014.2355850.

[86] Juang WS, Chen ST, Liaw HT. Robust and efficient password-authenticated key agreement using smart cards. *Industrial Electronics, IEEE Transactions on* 2008; **55**(6):2551–2556.

[87] Xu J, Zhu WT, Feng DG. An improved smart card based password authentication scheme with provable security. *Computer Standards & Interfaces* 2009; **31**(4):723–728.

[88] Lee SW, Kim HS, Yoo KY. Improvement of chien et al.'s remote user authentication scheme using smart cards. *Computer Standards & Interfaces* 2005; **27**(2):181–183.

[89] Lee NY, Chiu YC. Improved remote authentication scheme with smart card. *Computer Standards & Interfaces* 2005; **27**(2):177–180.

[90] Sood SK, Sarje AK, Singh K. An improvement of xu et al.'s authentication scheme using smart cards. *Proceedings of the third annual ACM Bangalore conference*, ACM, 2010; 15.

[91] Song R. Advanced smart card based password authentication protocol. *Computer Standards & Interfaces* 2010; **32**(5):321–325.

[92] Chen BL, Kuo WC, Wuu LC. Robust smart-card-based remote user password authentication scheme. *International Journal of Communication Systems* 2014; **27**(2):377–389.

[93] Qu J, Tan XL. Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem. *Journal of Electrical and Computer Engineering* 2014; **2014**:16.

[94] Mishra D, Das AK, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications* 2014; **41**(18):8129–8143.

[95] Chang CC, Cheng TF, Hsueh WY. A robust and efficient dynamic identity-based multi-server authentication scheme using smart cards. *International Journal of Communication Systems* 2014; :n/a–n/adoi: 10.1002/dac.2830.

[96] Wei J, Hu X, Liu W. Two-factor authentication scheme using attribute and password. *International Journal of Communication Systems* 2014; :n/a–n/adoi: 10.1002/dac.2915.

[97] Chen CT, Lee CC. A two-factor authentication scheme with anonymity for multi-server environments. *Security and Communication Networks* 2014; :n/a–n/adoi: 10.1002/sec.1109. URL `http://dx.doi.org/10.1002/sec.1109`.

[98] Li X, Ma J, Wang W, Xiong Y, Zhang J. A novel smart card and dynamic id based remote user authentication scheme for multi-server environments. *Mathematical and Computer Modelling* 2013; **58**(1):85–95.

[99] Wang Yy, Liu Jy, Xiao Fx, Dan J. A more efficient and secure dynamic id-based remote user authentication scheme. *Computer communications* 2009; **32**(4):583–585.

[100] Das ML, Saxena A, Gulati VP. A dynamic id-based remote user authentication scheme. *Consumer Electronics, IEEE Transactions on* 2004; **50**(2):629–631.

[101] Chang YF, Tai WL, Chang HC. Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. *International Journal of Communication Systems* 2014; **27**(11):3430–3440.

[102] Kumari S, Khan MK. More secure smart card-based remote user password authentication scheme with user anonymity. *Security and Communication Networks* 2014; **7**(11):2039–2053.

[103] Kumari S, Khan MK, Atiquzzaman M. User authentication schemes for wireless sensor networks: A review. *Ad Hoc Networks* 2014; .

[104] Chaudhry SA. Comment on 'robust and efficient password authenticated key agreement with user anonymity for session initiation protocol-based communications'. *IET Communications* May 2015; **9**:1034–1034(1).

[105] Wen F, Li X. An improved dynamic id-based remote user authentication with key agreement scheme. *Computers & Electrical Engineering* 2012; **38**(2):381–387.

[106] Tang Hb, Liu Xs. Cryptanalysis of a dynamic id-based remote user authentication with key agreement scheme. *International Journal of Communication Systems* 2012; **25**(12):1639–1644.

[107] An YH. Security improvements of dynamic id-based remote user authentication scheme with session key agreement. *Advanced Communication Technology (ICACT), 2013 15th International Conference on*, 2013; 1072–1076.

[108] Yang H, Kim H, Mtonga K. An efficient privacy-preserving authentication scheme with adaptive key evolution in remote health monitoring system. *Peer-to-Peer Networking and Applications* 2014; :1–11doi: 10.1007/s12083-014-0299-6. URL `http://dx.doi.org/10.1007/s12083-014-0299-6`.

[109] Diffie W, Hellman ME. New directions in cryptography. *Information Theory, IEEE Transactions on* 1976; **22**(6):644–654.

[110] Lee CC, Chen CL, Wu CY, Huang SY. An extended chaotic maps-based key agreement protocol with user anonymity. *Nonlinear Dynamics* 2012; **69**(1-2):79–87.

[111] Jiang Q, Ma J, Ma Z, Li G. A privacy enhanced authentication scheme for telecare medical information systems. *Journal of Medical Systems* 2013; **37**(1):9897, doi: 10.1007/s10916-012-9897-0. URL `http://dx.doi.org/10.1007/s10916-012-9897-0`.

[112] Xu L, Wu F. Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. *Journal of Medical Systems* 2015; **39**(2):10, doi: 10.1007/s10916-014-0179-x. URL `http://dx.doi.org/10.1007/s10916-014-0179-x`.

[113] Das A. A secure and robust password-based remote user authentication scheme using smart cards for the integrated epr information system. *Journal of Medical Systems* 2015; **39**(3):25, doi: 10.1007/s10916-015-0204-8. URL `http://dx.doi.org/10.1007/s10916-015-0204-8`.

[114] Kumari S, Khan MK, Kumar R. Cryptanalysis and improvement of Śa privacy enhanced scheme for telecare medical information systemsŠ. *Journal of Medical Systems* 2013; **37**(4):1–11.

[115] Liu J, Zhang Z, Chen X, Kwak KS. Certificateless remote anonymous authentication schemes for wirelessbody area networks. *Parallel and Distributed Systems, IEEE Transactions on* 2014; **25**(2):332–342.

[116] Wei J, Hu X, Liu W. An improved authentication scheme for telecare medicine information systems. *Journal of Medical Systems* 2012; **36**(6):3597–3604, doi: 10.1007/s10916-012-9835-1. URL `http://dx.doi.org/10.1007/s10916-012-9835-1`.

[117] Wu S, Chen K. An efficient key-management scheme for hierarchical access control in e-medicine system. *Journal of Medical Systems* 2012; **36**(4):2325–2337.

[118] Khan MK, Kumari S. Cryptanalysis and improvement of an efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *Security and Communication Networks* 2014; **7**(2):399–408, doi: 10.1002/sec.791. URL `http://dx.doi.org/10.1002/sec.791`.

[119] Wang Z, Huo Z, Shi W. A dynamic identity based authentication scheme using chaotic maps for telecare medicine information systems. *Journal of Medical Systems* 2014; **39**(1):158, doi: 10.1007/s10916-014-0158-2. URL `http://dx.doi.org/10.1007/s10916-014-0158-2`.

[120] Farash MS, Attari MA. An efficient and provably secure three-party password-based authenticated key exchange protocol based on chebyshev chaotic maps. *Nonlinear*

Dynamics 2014; **77**(1-2):399–411.

[121] Giri D, Maitra T, Amin R, Srivastava P. An efficient and robust rsa-based remote user authentication for telecare medical information systems. *Journal of Medical Systems* 2014; **39**(1):145, doi: 10.1007/s10916-014-0145-7. URL `http://dx.doi.org/10.1007/s10916-014-0145-7`.

[122] Wu ZY, Lee YC, Lai F, Lee HC, Chung Y. A secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems* 2012; **36**(3):1529–1535, doi: 10.1007/s10916-010-9614-9. URL `http://dx.doi.org/10.1007/s10916-010-9614-9`.

[123] Debiao H, Jianhua C, Rui Z. A more secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems* 2012; **36**(3):1989–1995, doi: 10.1007/s10916-011-9658-5. URL `http://dx.doi.org/10.1007/s10916-011-9658-5`.

[124] Zhu Z. An efficient authentication scheme for telecare medicine information systems. *Journal of Medical Systems* 2012; **36**(6):3833–3838, doi: 10.1007/s10916-012-9856-9. URL `http://dx.doi.org/10.1007/s10916-012-9856-9`.

[125] Khan MK, Kim SK, Alghathbar K. Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme. *Computer Communications* 2011; **34**(3):305 – 309, doi: http://dx.doi.org/10.1016/j.comcom.2010.02.011. Special Issue of Computer Communications on Information and Future Communication Security.

[126] Chen HM, Lo JW, Yeh CK. An efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *Journal of Medical Systems* 2012; **36**(6):3907–3915, doi: 10.1007/s10916-012-9862-y. URL `http://dx.doi.org/10.1007/s10916-012-9862-y`.

[127] Yoon EJ, Ryu EK, Yoo KY. Attacks and solutions of yang et al.'s protected password changing scheme. *Informatica* 2005; **16**(2):285–294.

[128] Xiang T, Wong KW, Liao X. On the security of a novel key agreement protocol based on chaotic maps. *Chaos, Solitons & Fractals* 2009; **40**(2):672 – 675, doi: http://dx.doi.org/10.1016/j.chaos.2007.08.012.

[129] Ch SA, uddin N, Sher M, Ghani A, Naqvi H, Irshad A. An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography. *Multimedia Tools and Applications* 2014; :1–13doi: 10.1007/s11042-014-2283-9.

[130] Witteman M. Advances in smartcard security. *Information Security Bulletin* 2002; **7**(2002):11–22.

[131] Liu JY, Zhou AM, Gao MX. A new mutual authentication scheme based on nonce and smart cards. *Computer Communications* 2008; **31**(10):2205–2209.

[132] Lee T, Chang J, Chan C, Liu H. Password-based mutual authentication scheme using smart cards. *The E-learning and Information Technology Symposium 2010 (EITS2010)*, 2010.

[133] He D, Wang D. Robust biometrics-based authentication scheme for multiserver environment. *System Journal, IEEE* 2014; **4**(1):253–264, doi: 10.1109/JSYST.2014.2301517.

[134] Mishra D, Mukhopadhyay S, Chaturvedi A, Kumari S, Khan MK. Cryptanalysis and improvement of yan et al.'s biometric-based authentication scheme for telecare medicine information systems. *Journal of medical systems* 2014; **38**(6):1–12.

[135] Li X, Wen Q, Li W, Zhang H, Jin Z. Secure privacy-preserving biometric authentication scheme for telecare medicine information systems. *Journal of medical systems* 2014; **38**(11):1–8.

[136] Khan MK. Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world. *IETE Technical Review* 2009; **26**(3):191–195.

[137] Khan MK, Zhang J. An efficient and practical fingerprint-based remote user authentication scheme with smart cards. *Information Security Practice and Experience*. Springer, 2006; 260–268.

[138] Awasthi AK, Srivastava K. A biometric authentication scheme for telecare medicine information systems with nonce. *Journal of medical systems* 2013; **37**(5):1–4.

[139] Tan Z. A user anonymity preserving three-factor authentication scheme for telecare medicine information systems. *Journal of medical systems* 2014; **38**(3):1–9.

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

[141] Leng L, Teoh ABJ, Li M, Khan MK. A remote cancelable palmprint authentication protocol based on multi-directional two-dimensional palmphasor-fusion. *Security and Communication Networks* 2014; **7**(11):1860–1871.

[142] Leng L, Teoh ABJ. Alignment-free row-co-occurrence cancelable palmprint fuzzy vault. *Pattern Recognition* 2015; **48**(7):2290–2303.

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

[144] Wen F, Guo D. An improved anonymous authentication scheme for telecare medical information systems. *Journal of medical systems* 2014; **38**(5):1–11.

[145] Lal Das M, Saxena A, Gulati VP. A dynamic id-based remote user authentication scheme. *arXiv preprint arXiv:0712.2235* 2007; .

[146] Amin R, Biswas G. An improved rsa based user authentication and session key agreement protocol usable in tmis. *Journal of Medical Systems* 2015; **39**(8):79, doi: 10.1007/s10916-015-0262-y. URL `http://dx.doi.org/10.1007/s10916-015-0262-y`.

[147] Amin R, Biswas G. A secure three-factor user authentication and key agreement protocol for tmis with user anonymity. *Journal of Medical Systems* 2015; **39**(8):78, doi: 10.1007/s10916-015-0258-7. URL `http://dx.doi.org/10.1007/s10916-015-0258-7`.

[148] Srivastava K, Awasthi AK, Kaul SD, Mittal R. A hash based mutual rfid tag authentication protocol in telecare medicine information system. *Journal of medical systems* 2015; **39**(1):1–5.

[149] Islam S, Biswas GP. Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography. *International Journal of Computer Mathematics* 2013; **90(11)**:2244–2258.

[150] Khan MK, Kumari S. An improved biometrics-based remote user authentication scheme with user anonymity. *BioMed research international* 2013; **2013**.

[151] Zhang M, Zhang J, Zhang Y. Remote three-factor authentication scheme based on fuzzy extractors. *Security and Communication Networks* 2015; **8**(4):682–693, doi: 10.1002/sec.1016.

[152] Das AK. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *International Journal of Communication Systems* 2015; doi: 10.1002/dac.2933.

[153] He D, Kumar N, Lee JH, Sherratt R. Enhanced three-factor security protocol for consumer usb mass storage devices. *Consumer Electronics, IEEE Transactions on* February 2014; **60**(1):30–37, doi: 10.1109/TCE.2014.6780922.

[154] Leu JS, Hsieh WB. Efficient and secure dynamic id-based remote user authentication scheme for distributed systems using smart cards. *Information Security, IET* 2014; **8**(2):104–113.

[155] Yan X, Li W, Li P, Wang J, Hao X, Gong P. A secure biometrics-based authentication scheme for telecare medicine information systems. *Journal of Medical Systems* 2013; **37**(5):9972, doi: 10.1007/s10916-013-9972-1. URL `http://dx.doi.org/10.1007/s10916-013-9972-1`.

[156] Mishra D, Mukhopadhyay S, Chaturvedi A, Kumari S, Khan MK. Cryptanalysis and improvement of yan et al.'s biometric-based authentication scheme for telecare medicine information systems. *Journal of medical systems* 2014; **38**(6):1–12.

[157] Cao J, Ma M, Li H. An uniform handover authentication between e-utran and non-3gpp access networks. *Wireless Communications, IEEE Transactions on* 2012; **11**(10):3644–3650.

[158] Farash MS, Chaudhry SA, Heydari M, Sajad Sadough SM, Kumari S, Khan MK. A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security. *International Journal of Communication Systems* 2015; :n/a–n/adoi: 10.1002/dac.3019. URL `http://dx.doi.org/10.1002/dac.3019`.

[159] He D, Chen C, Chan S, Bu J. Secure and efficient handover authentication based on bilinear pairing functions. *Wireless Communications, IEEE Transactions on* 2012; **11**(1):48–53.

[160] He D, Chen C, Chan S, Bu J. Analysis and improvement of a secure and efficient handover authentication for wireless networks. *Communications Letters, IEEE* 2012; **16**(8):1270–1273.

[161] Yeo SL, Yap WS, Liu JK, Henricksen M. Comments on" analysis and improvement of a secure and efficient handover authentication based on bilinear pairing functions". *Communications Letters, IEEE* 2013; **17**(8):1521–1523.

[162] Tsai JL, Lo NW, Wu TC. Secure handover authentication protocol based on bilinear pairings. *Wireless personal communications* 2013; **73**(3):1037–1047.

[163] Islam S, Khan MK. Provably secure and pairing-free identity-based handover authentication protocol for wireless mobile networks. *International Journal of Communication Systems* 2014; .

[164] He D, Bu J, Chan S, Chen C. Handauth: Efficient handover authentication with conditional privacy for wireless networks. *Computers, IEEE Transactions on* 2013; **62**(3):616–622.

[165] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. *Journal of cryptology* 2000; **13**(3):361–396.

[166] Debiao H, Jianhua C, Jin H. An id-based proxy signature schemes without bilinear pairings. *annals of telecommunications-annales des télécommunications* 2011; **66**(11-12):657–662.

[167] Yoon EJ, Yoo KY. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *The Journal of Supercomputing* 2013; **63**(1):235–255.

[168] Xue K, Hong P, Ma C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer and System Sciences* 2014; **80**(1):195–206.

[169] Lu Y, Li L, Peng H, Yang X, Yang Y. A lightweight id based authentication and key agreement protocol for multiserver architecture. *International Journal of Distributed Sensor Networks* 2015; **2015**.

[170] Li X, Niu J, Khan MK, Liao J, Zhao X. Robust three-factor remote user authentication scheme with key agreement for multimedia systems. *Security and Communication Networks* 2014; doi: 10.1002/sec.961.

[171] Mishra D, Kumari S, Khan MK, Mukhopadhyay S. An anonymous biometric-based remote user-authenticated key agreement scheme for multimedia systems. *International Journal of Communication Systems* 2015; :n/a–n/adoi: 10.1002/dac.2946.

[172] Li X, Khan M, Kumari S, Liao J, Liang W. Cryptanalysis of a robust smart card authentication scheme for multi-server architecture. *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on*, 2014; 120–123, doi: 10.1109/ISBAST.2014.7013106.

[173] Dinh HT, Lee C, Niyato D, Wang P. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing* 2013; **13**(18):1587–1611.

[174] Khan AN, Kiah MM, Khan SU, Madani SA. Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems* 2013; **29**(5):1278–1299.

[175] Fernando N, Loke SW, Rahayu W. Mobile cloud computing: A survey. *Future Generation Computer Systems* 2013; **29**(1):84–106.

[176] Tsai JL, Lo NW, Wu TC. Secure delegation-based authentication protocol for wireless roaming service. *Communications Letters, IEEE* 2012; **16**(7):1100–1102.

[177] Xiao Z, Xiao Y. Security and privacy in cloud computing. *Communications Surveys & Tutorials, IEEE* 2013; **15**(2):843–859.

[178] Armando A, Carbone R, Compagna L, Cuéllar J, Pellegrino G, Sorniotti A. An authentication flaw in browser-based single sign-on protocols: Impact and remediations. *Computers & Security* 2013; **33**:41–58.

[179] Koblitz N. Elliptic curve cryptosystems. *Mathematics of computation* 1987; **48**(177):203–209.

[180] Miller VS. Use of elliptic curves in cryptography. *Advances in CryptologyŮCRYPTOŠ85 Proceedings*, Springer, 1986; 417–426.

[181] Barker E, Barker W, Burr W, Polk W, Smid M. Recommendation for key management-part 1: General (revised. *NIST special publication*, Citeseer, 2006.

[182] Du H, Wen Q. An efficient identity-based short signature scheme from bilinear pairings. *Computational Intelligence and Security, 2007 International Conference on*, IEEE, 2007; 725–729.

[183] Lim HW, Robshaw MJ. On identity-based cryptography and grid computing. *Computational Science-ICCS 2004*. Springer, 2004; 474–477.

[184] Lim HW, Robshaw MJ. A dynamic key infrastructure for grid. *Advances in Grid Computing-EGC 2005*. Springer, 2005; 255–264.

[185] Li H, Dai Y, Tian L, Yang H. Identity-based authentication for cloud computing. *Cloud computing*. Springer, 2009; 157–166.

[186] Hughes D, Shmatikov V. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer security* 2004; **12**(1):3–36.

[187] Tsai JL, Lo NW, Wu TC. Novel anonymous authentication scheme using smart cards. *Industrial Informatics, IEEE Transactions on* 2013; **9**(4):2004–2013.

[188] Wang D, Mei Y, Ma Cg, Cui Zs. Comments on an advanced dynamic id-based authentication scheme for cloud computing. *Web Information Systems and Mining*. Springer, 2012; 246–253.

[189] Sun H, Wen Q, Zhang H, Jin Z. A novel remote user authentication and key agreement scheme for mobile client-server environment. *Appl. Math* 2013; **7**(4):1365–1374.

[190] Goriparthi T, Das ML, Saxena A. An improved bilinear pairing based remote user authentication scheme. *Computer Standards & Interfaces* 2009; **31**(1):181–185.

[191] De Caro A, Iovino V. jpbc: Java pairing based cryptography. *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, IEEE: Kerkyra, Corfu, Greece, June 28 - July 1, 2011; 850–855.

[192] jpbc: Java pairing based cryptography. 2015. URL `http://gas.dia.unisa.it/projects/jpbc/VcUnwbU0rlw`, [Online; accessed 7-August-2015].

[193] Chen S, Ning J. Constraints on e-commerce in less developed countries: The case of china. *Electronic Commerce Research* 2002; **2**(1-2):31–42, doi: 10.1023/A:1013331817147.

[194] Kshetri N. Cybercrime and cyber-security issues associated with china: some economic and institutional considerations. *Electronic Commerce Research* 2013; **13**(1):41–69, doi: 10.1007/s10660-013-9105-4. URL `http://dx.doi.org/10.1007/s10660-013-9105-4`.

[195] Huang X, Dai X, Liang W. Bulapay: a novel web service based third-party payment system for e-commerce. *Electronic Commerce Research* 2014; **14**(4):611–633, doi: 10.1007/s10660-014-9172-1. URL `http://dx.doi.org/10.1007/s10660-014-9172-1`.

[196] Chaum D. Blind signatures for untraceable payments. *Advances in cryptology*, Springer, 1983; 199–203.

[197] Lysyanskaya A, Ramzan Z. Group blind digital signatures: A scalable solution to electronic cash. *Financial Cryptography*, Springer, 1998; 184–197.

[198] Zhang L, Zhang F, Qin B, Liu S. Provably-secure electronic cash based on certificate-less partially-blind signatures. *Electronic Commerce Research and Applications* 2011; **10**(5):545–552.

[199] Xiaojun W. An e-payment system based on quantum group signature. *Physica Scripta* 2010; **82**(6):065 403.

[200] Eslami Z, Talebi M. A new untraceable off-line electronic cash system. *Electronic Commerce Research and Applications* 2011; **10**(1):59–66.

[201] Yen YC, Wu TC, Lo NW, Tsai KY. A fair-exchange e-payment protocol for digital products with customer unlinkability. *KSII Transactions on Internet and Information Systems (TIIS)* 2012; **6**(11):2956–2979.

[202] Chen X, Li J, Ma J, Lou W, Wong DS. New and efficient conditional e-payment systems with transferability. *Future Generation Computer Systems* 2014; **37**:252–258.

[203] Zheng Y. Digital signcryption or how to achieve cost (signature & encryption) « cost (signature)+ cost (encryption). *Advances in Cryptology,CRYPTO'97*. Springer, 1997; 165–179.

[204] He D, Kumar N, Chilamkurti N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Information Sciences* 2015; .

[205] He D, Zeadally S. Authentication protocol for an ambient assisted living system. *Communications Magazine, IEEE* 2015; **53**(1):71–77.

[206] Abdalla M, Benhamouda F, Pointcheval D. Public-key encryption indistinguishable under plaintext-checkable attacks 2015; .

[207] Ch SA, Sher M, *et al.*. Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. *Information Systems, Technology and Management*. Springer, 2012; 135–142.

[208] Ch SA, Nasar W, Javaid Q, *et al.*. Efficient signcryption schemes based on hyperelliptic curve cryptosystem. *Emerging Technologies (ICET), 2011 7th International Conference on*, IEEE, 2011; 1–4.

[209] uddin N, Ch SA, Amin N. Signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem. *High Capacity Optical Networks and Enabling Technologies (HONET), 2011*, IEEE, 2011; 244–247.

[210] Li CT. Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control* 2011; **40**(2):157–162.

[211] Hong JW, Yoon SY, Park DI, Choi MJ, Yoon EJ, Yoo KY. A new efficient key agreement scheme for vsat satellite communications based on elliptic curve cryptosystem. *Information Technology and Control* 2011; **40**(3):252–259.

[212] Farash MS, Attari MA. A provably secure and efficient authentication scheme for access control in mobile pay-tv systems. *Multimedia Tools and Applications* 2014; :1–20.

[213] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ecdsa). *International Journal of Information Security* 2001; **1**(1):36–63.

[214] Farash MS. Cryptanalysis and improvement of an improved authentication with key agreement scheme on elliptic curve cryptosystem for global mobility networks. *International Journal of Network Management* 2014; .