Design of Authentication Protocols for Mobile Online Social Networks

Thesis submitted by

Munmun Bhattacharya

Doctor of Philosophy (Engineering)

Department of Information Technology
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2023

Design of Authentication Protocols for Mobile Online Social Networks

by

Munmun Bhattacharya

Registration Number - 1021909004

Thesis submitted for the

Doctor of Philosophy (Engineering)

Degree of Jadavpur University, Kolkata, India

Supervisors:

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

Dr. Sandip Roy
Assistant Professor
Dept. of Computer Science and Engineering,
Asansol Engineering College,
Asansol, WB 713305, India

2023

Jadavpur University
Kolkata 700 032, India

INDEX NO. D-7/E/452/19

1. Title of The Thesis:

Design of Authentication Protocols for Mobile Online Social Networks

2. Name, Designation and Institution of the Supervisors:

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

2. Dr. Sandip Roy
   Assistant Professor
   Department of Computer Science and Engineering
   Asansol Engineering College
   Asansol, WB 713305, India

3. List of Publications:

1. Journal papers

1. Munmun Bhattacharya, Sandip Roy, Kamlesh Mistry, Hubert PH Shum, and Samiran Chattopadhyay. "A privacy-preserving efficient location-sharing scheme for mobile online social network applications," in ***IEEE Access***, Vol. 8, pp. 221330-221351, 2020. (2020 SCI Impact Factor: 4.48)

2. Munmun Bhattacharya, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Soumya Banerjee, and Ankush Mitra. "DDoS attack resisting authentication protocol for mobile based online social network applications," in ***Journal of Information Security and Applications***, Vol. 65, pp. 103115-103134, 2022. (2022 SCI Impact Factor: 3.872)

3. Munmun Bhattacharya, Sandip Roy, Samiran Chattopadhyay, Ashok Kumar Das, and Sajjad Shaukat Jamal. "ASPA-MOSN: An Efficient User Authentication Scheme for Phishing Attack Detection in Mobile Online Social Networks," in ***IEEE Systems Journal***, Vol. 17, No. 1, pp. 234-245, March 2023, doi: 10.1109/JSYST.2022.3168234. (2022 SCI Impact Factor: 4.802)

4. Munmun Bhattacharya, Sandip Roy, Samiran Chattopadhyay, Ashok Kumar Das, and Sachin Shetty. "A comprehensive survey on online social networks security and privacy issues: threats, machine learning-based solutions and open challenges," in ***Security and Privacy, Wiley Online Library***, Vol. 6, No. 1, pp. e275, 2023. doi:10.1002/spy2.275.

2. International conference papers

1. Munmun Bhattacharya, Sandip Roy, Soumya Banerjee, and Samiran Chattopadhyay. "Cryptanalysis of a Centralized Location-Sharing Scheme for Mobile Online Social Networks", in ***Advanced Computing and Systems for Security (ACSS '20)***, Vol. 11, pp. 17-30, Kolkata, India, February 9-10, 2020.
2. Munmun Bhattacharya, Sandip Roy, and Samiran Chattopadhyay. "An Efficient Authentication Scheme for Mobile Online Social Networks", in ***Advanced Computing and Systems for Security (ACSS '21)***, Vol. 14, pp. 31–41, Kolkata, India, April 9-10, 2021.

4. List of Patents: Nil

5. List of Presentations in National/International Conferences

1. Munmun Bhattacharya, Sandip Roy, Soumya Banerjee, and Samiran Chattopadhyay. "Cryptanalysis of a Centralized Location-Sharing Scheme for Mobile Online Social Networks", in ***Advanced Computing and Systems for Security (ACSS '20)***, Kolkata, India, February 9-10, 2020.
2. Munmun Bhattacharya, Sandip Roy, and Samiran Chattopadhyay. "An Efficient Authentication Scheme for Mobile Online Social Networks", in ***Advanced Computing and Systems for Security (ACSS '21)***, Kolkata, India, April 9-10, 2021.

PROFORMA – 1

"Statement of Originality"

I ***Ms. Munmun Bhattacharya*** registered on ***25th June, 2019*** do hereby declare that this thesis entitled ***"Design of Authentication Protocols for Mobile Online Social Networks"*** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 8 %.

Signature of Candidate:

Munmun Bhattacharya

(Munmun Bhattacharya)
Date :

Certified by Supervisor(s):

Samiran Chattopadhyay
PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block - LB, Plot - 8, Sector - 3
Salt Lake, Kolkata - 700098, India

1. ---

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

2. Sandip Roy

Dr. Sandip Roy
Assistant Professor
Dept. of Computer Science and Engineering,
Asansol Engineering College,
Asansol, WB 713305, India

Dr. Sandip Roy
Assistant Professor
Deptt. of Computer Science & Engg.
Asansol Engineering College
Asansol-713305

CERTIFICATE FROM THE SUPERVISORS

This is to certify that the thesis entitled ***"Design of Authentication Protocols for Mobile Online Social Networks"****, submitted by* ***Ms. Munmun Bhattacharya****, who got her name registered on* 25^{th} ***June, 2019*** *for the award of* ***Ph.D. (Engg.)*** *degree of Jadavpur University is absolutely based upon her own work under the supervision of* ***Dr. Samiran Chattopadhyay****, and* ***Dr. Sandip Roy*** *and that neither her thesis nor any part of the thesis has been submitted for any degree or any academic award anywhere before.*

Samiran Chattopadhyay

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block - LB, Plot - 8, Sector - 3
Salt Lake, Kolkata - 700098, India

1. _____

Signature of the Supervisor and date with Official Seal

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

2. Sandip Roy.

Signature of the Supervisor and date with Official Seal

Dr. Sandip Roy
Assistant Professor
Dept. of Computer Science and Engineering,
Asansol Engineering College,
Asansol, WB 713305, India

Dr. Sandip Roy
Assistant Professor
Deptt. of Computer Science & Engg.
Asansol Engineering College
Asansol-713305

Acknowledgments

Although a thesis is known by the name of its author, there are many without whose direct or indirect help, a thesis does not materialize. It is time to thank them.

I would like to acknowledge the moral and intellectual support given to me by my supervisors Dr. Samiran Chattopadhyay and Dr. Sandip Roy during my Ph.D. program. Thanks to my supervisors' constant guidance and approaches for which this long and difficult journey becomes smooth and also very interesting.

First and foremost, I am profoundly grateful to my supervisor Dr. Samiran Chattopadhyay for giving me the guidance, encouragement, and counsel throughout my research, his way to do research as well as his attitude towards study and analysis of any particular subject influenced me immensely, and I still feel there is a lot to learn from him. Without his invaluable advice and assistance, it would not have been possible for me to complete this thesis. I would like to thank him for allowing me to work under his supervision.

I am extremely grateful to my other supervisor, Dr. Sandip Roy for his constant motivation, steady supervision, and valuable suggestions. Above all, his friendly approach worked as a driving force that provided me an insight into the subject of study without which the study would not have materialized. Among other things, most important of all was probably his calm and constant belief in my ability to do the Ph.D. work.

I would like to express my gratitude to my co-author Dr. Ashok Kumar Das for his help, advice, and mentoring during my research work. I am deeply obligated to him for showing me how to do successful research, and how to communicate research results effectively.

I thank Dr. Parama Bhaumik, Head, Department of Information Technology, Jadavpur University for her moral support, fruitful cooperation, and for serving on my Doctoral Scrutiny Committee.

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

Munmun Bhattacharya

Department of Information Technology
Jadavpur University
Kolkata, India

Abstract

Over the past few years, online social networks (OSNs) have become an inseparable part of people's daily lives. Instead of being passive readers, people are now enjoying their role as content contributors. OSN has permitted its users to share their information, and multimedia content. OSN users can express themselves in virtual communities, providing their opinion, and interacting with others. As a consequence, phishing attacks, replay attacks, man-in-the-middle attacks, denial-of-service attacks, impersonation attacks, privileged-insider attacks, etc. such security threats in OSNs have emerged as a major concern. Hence, to achieve hazard-free social network services, the design of security protocol, user authentication, and remote access control mechanisms is highly essential in online social network applications. In this thesis, I aim to study privacy and security issues in the following areas: 1) privacy preservation in location sharing of mobile OSN Applications, 2) DDoS attack resisting protocol for mobile-based online social network applications, and 3) Phishing Attack resisting mechanisms of Mobile Online Social Networks.

In the first study, I have proposed a privacy preserving, secure and efficient location sharing scheme for mOSNs, which shows both efficiency and flexibility in the location update, sharing, and query of social friends and social strangers. According to the proposed location sharing scheme, the Social Network Servers (SNS) for storing social network data and Location Based Server (LBS) for storing location information are integrated into one single entity. This eliminates the possibility of a Location Based Server(LBS) to reveal the social network topology structure of a social user. The user biometric along with the password is used for authorization and at the time of authentication, a session key is established between the SNS and an OSN user. All location updates and friend's location query messages are encrypted with this session key before transmission. The proposed scheme allows a user to decide a distance threshold, up to which he/she wants to make himself/herself visible to social friends. This imposes a much better user-controlled restriction on location sharing. This significantly makes the authentication process secure, faster, and efficient and reduces communication costs. The proposed scheme is lightweight compared to other related schemes. The integration of LBS and SNS into a set of single entity servers reduces their internal communication overhead also. The security of the proposed scheme is validated using random oracle-based formal security proof and Burrows-Abadi-Needham (BAN) logic-based authentication proof, followed by informal security analysis. Additionally, we have used ProVerif 1.93 to verify the security of the system. The efficiency and practicability of the proposed scheme are demonstrated through

experimental implementation and evaluation.

The second study is based on a multi-faceted, secure, and lightweight authentication scheme that resists DDoS and other security attacks in mobile OSN environments. In DDoS attacks, the adversary sends the same login message to the OSN server repeatedly to deny the services. According to this remedy for DDoS attacks in the OSN environment, after a certain threshold, the scheme discards further user login attempts and blocks an adversary who intends to overload the network server. I use the pre-loaded shadow identity and emergency key pairs, and a key-refilling strategy that rebuilds the essential synchronization between a blocked naive user and the OSN server. This technique restores the intended unlinkability property of the protocol. Using NS3 simulation, I study the impact of DDoS attackers on network throughput and network delay. Moreover, I validate and compare the proposed scheme against state-of-the-art solutions using real attacks and benign datasets. The Canadian Institute for Cyber Security (CIC) DoS dataset 2017, which is generated by capturing the normal and DoS attack packets separately with subsequent pre-processing for testing is used as the dataset. I also use machine learning (ML) algorithms, such as K-Nearest Neighbor (KNN), Gaussian Naive Bayes, and Multilayer Perceptron (MLP) to demonstrate the performance of the proposed solution in a practical attack detection scenario. It is observed that these algorithms provide 97.05%, 95.48%, and 96.6% DDoS attack detection accuracy, respectively.

The third and final study involves the design of a secure and lightweight cryptography-based authentication scheme (ASPA-mOSN) that provides resistance to phishing and other related attacks in OSNs. The phishing attack is carried out by the exercise of sending fraudulent communications (called phishing hook) that pretends to come from a reputable source. In OSN phishing, phishers steal the login or other sensitive information of a victim user using a phishing hook. In my proposed ASPA-mOSN scheme for resisting phishing attacks, a secure authentication will be done using only the password and biometric of the user without asking for other credentials. As a result, attackers cannot acquire the information for promoting attacks. The security of the proposed scheme is explained using both informal security analysis, and formal security analysis through Real-Or-Random (ROR) model. Finally, I compare the security, functionality, computation cost, and communication cost of the proposed scheme with other existing related schemes. The comparison result shows that the proposed scheme outperforms and needs less computation and communication costs compared to the other competing schemes.

Dissemination of Work

Chapter #4.

Munmun Bhattacharya, Sandip Roy, Kamlesh Mistry, Hubert PH Shum, and Samiran Chattopadhyay. "A privacy-preserving efficient location-sharing scheme for mobile online social network applications," in ***IEEE Access***, Vol. 8, pp. 221330-221351, 2020. (2020 SCI Impact Factor: 4.48)

Chapter #5.

Munmun Bhattacharya, Sandip Roy, Ashok Kumar Das, Samiran Chattopadhyay, Soumya Banerjee, and Ankush Mitra. "DDoS attack resisting authentication protocol for mobile based online social network applications," in ***Journal of Information Security and Applications***, Vol. 65, pp. 103115-103134, 2022. (2022 SCI Impact Factor: 3.872)

Chapter #6.

Munmun Bhattacharya, Sandip Roy, Samiran Chattopadhyay, Ashok Kumar Das, and Sajjad Shaukat Jamal. "ASPA-MOSN: An Efficient User Authentication Scheme for Phishing Attack Detection in Mobile Online Social Networks," in ***IEEE Systems Journal***, Vol. 17, No. 1, pp. 234-245, March 2023, doi: 10.1109/JSYST.2022.3168234. (2022 SCI Impact Factor: 4.802)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online Social Networking (OSN) sites have become an inseparable part of today's life. People use these online platforms to connect with their family, friends and share their personal opinions and information. Facebook, Instagram, Twitter - the list of popular social networking sites is increasing day by day. Before using these web services, people must have to register with their credentials or sensitive information, such as date of birth, address, mobile number, etc. A person joins the network, sets up virtual connectivity with other users of similar interests, and shares his/her emotions, information, activities, photos, videos, etc. [29], [138]. In this virtual world, people have the perception that all their interactions or sharing will be secure and private as they are connected mainly with their family, friends, and colleagues [46]. But in reality, people are sharing their data on Facebook, professional information on LinkedIn, etc. Such multi-directional exposure puts users' privacy and security at greater risks [135]. Sharing of this huge volume of sensitive and personal information makes such OSN sites soft targets for attackers.

Figure 1.1 shows the total number of OSN users in India, who use different social media [167]. Online social networks are used for establishing social relationships, entertainment, sharing common interests with other individuals, improving business opportunities, developing professional careers, etc. As per a published report of Zephoria, 1.91 billion users log onto Facebook and there is a 7% increase of users year over year that results in a 43% increase in average revenue per user of Facebook [50]. Figure 1.2 depicts the number of users who use different online social networking sites [166]. Such a huge number of users upload an enormous amount of data and thus OSN sites become alluring targets of attackers. Malicious content can be posted using shortened uniform resource locators (URLs) or hiding inside any multimedia content. Every day around 2 lakhs websites are hacked [191].

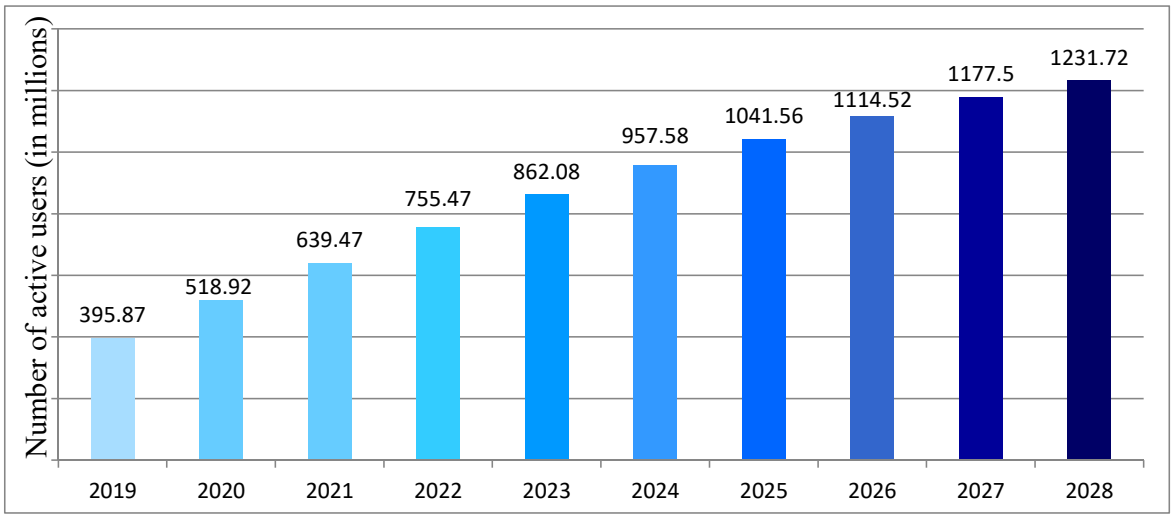

Figure 1.1: Number of social network users in India [167]

In this chapter, I start with providing a brief introduction of online social networks, their characteristics, evolution, and network structure. Next, I describe the privacy and security issues of online social networks, which have motivated me to build new schemes for enhancing secure applications in the future. Finally, I describe the organization of the thesis in brief.

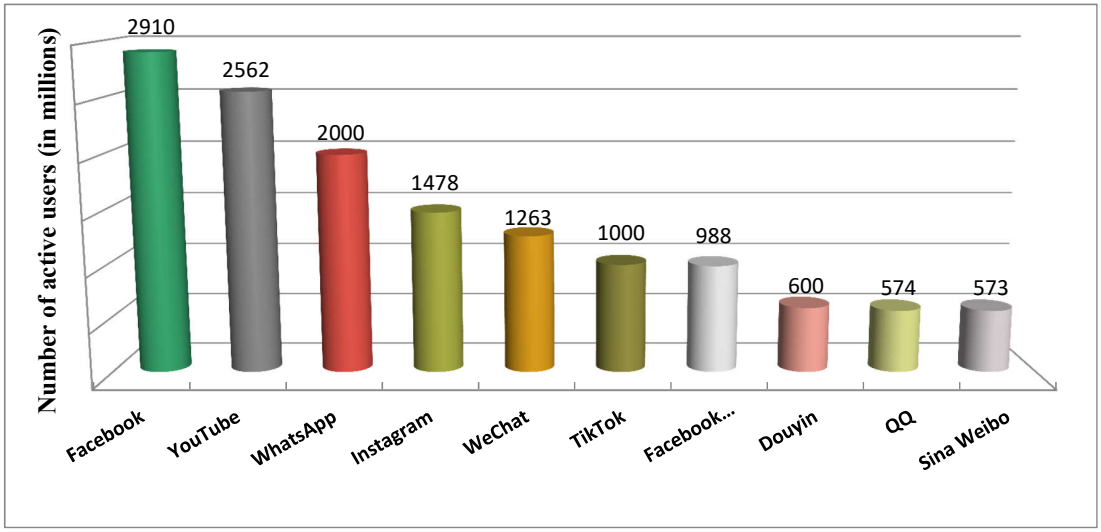

Figure 1.2: Most popular social networks worldwide as of January 2022 [166]

1.1 Brief Introduction of Online Social Network

In this section, I discuss the definition, basic characteristics, network structure of Online Social Networks and their various types.

1.1.1 Definition

Online social networks (OSN) are web services for building virtual relationships between users with similar interests, activities, and backgrounds. OSNs permit its users to (a) create a semi-public or public profile, (b) set up virtual connections with a list of other OSN users, (c) navigate the profiles and links of other users by browsing the OSN [29].

Online social networks are used to serve several purposes, but across all sites, three major roles are most common. Firstly, OSNs are used to continue and strengthen existing social relations or create new social contacts. OSN users can build their social networks visible to other users, thus users can browse other's networks, and communicate with users who are already a part of their extended social network [29]. Secondly, OSNs allow their users to upload their information, photos, videos, etc. for sharing. Content sharing may vary from service provider to service provider, sometimes the sharing content can be the user's profile only. Thirdly, using the facility of organizing, filtering, and recommending, OSN services can be used to find new, interesting and innovative content.

1.1.2 Characteristics

Specific features may vary to some extent between different social networking services but according to theoretical and empirical analyses, there are six attributes of social networking services as follows.

- **Virtual Identity/Digital Persona:** Social networking services provide a facility to create an online or digital persona. Here, a persona is a representation or image of a user. Though the structure and features of that representation are given by the OSN service providers, the persona can be developed and controlled by the user himself/herself. A persona is a projected image of a user. It is not necessary to have more likeliness with the user's original identity that he/she understands himself/herself or is understood by others. An OSN profile can be treated as a digital identity that is the first meaningful visualization experienced by another user.

- **Network Building:** The OSN service providers offer tools and opportunities to build the virtual network(s) by a user. They encourage to search other users of similar themes and interests. They recruit tools for users so that users can meet or be introduced to other users, and groups of users offline. Users can build online communities inside the OSN with their friends, family, colleagues, contacts with common interests, and so on. Online networks and communities are independent, but sometimes they overlap and significantly interact with users' real-world contacts. The target of OSN service providers is to build up an expressive network with sufficient users.
- **Network Maintenance:** For persistence, OSN service providers provide features so that networks of users can grow, and can make persistent changes to users' persona. They also maintain the connections between networked users in their online and real-world circumstances irrespective of other changes.
- **Network Interaction:** OSN service providers provide tools for their users for interacting via direct communication, sharing information, games, or activities, exchanging virtual objects. Hardships in any physical interaction, like time, geographical separation, or mobility are overcome in the virtual world.
- **Formation of Virtual Content:** OSN users manage their digital profile; they can also share virtual content such as pictures, videos, audio clips, text, or any other programs or applications. These are all exchanged as primary components of the network interaction and this information is important in terms of the virtual identity of the user.
- **Network Self-Governance:** The network depicts noticeable social measures, social conventions, informal codes of behaviour, and rules and regulations. Rules and regulations on the software functions like what is allowed or disallowed are partially implemented by the service providers but network members are primarily responsible for online interactions and other actions.

1.1.3 Network structure

Social networks are typically represented as graphs. Individuals are represented as vertices and lines connecting individuals are edges or links. The basic components of an online social network is described as follows.

- **Vertex:** Individuals or users are represented as vertices or nodes, who use the services and share information to other social friends.

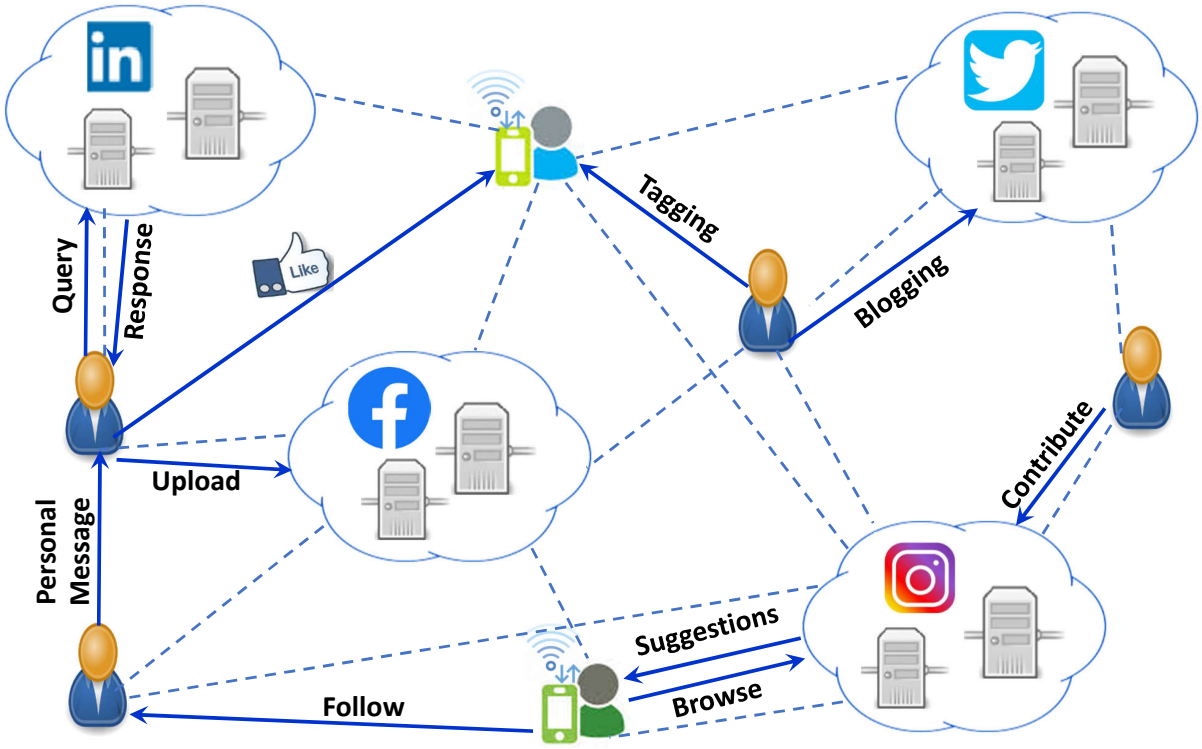

Figure 1.3: Network structure of OSN

- **Edge:** Relationships between individuals are represented as links or edges. Edges can have weights to represent the strength of an interaction and can either be directed or un-directed.

- **Label:** Additional information about the individuals or relationships can be stored as labels. Information about an individual, such as age, gender, education, location, etc. can be stored in vertex labels and information about the relationship, such as friendship, association, co-authorship, etc. weights of relationship, e.g. frequency of message communication, trustworthiness, etc. can be stored in edge labels.

- **OSN Server:** Each OSN service provider has its own sets of server for providing the services and storing the content of the users.

First, a user (Node) registers with an OSN service provider (OSN servers). This is a one-time operation and all the information provided by the user is stored in the database of OSN servers. A user needs to login into the registered OSN servers for further interactions. Thereafter, a user can build up his/her own social relationships (Edges) with other social users,

share information, and surf the OSN. Social networks can vary in size and heterogeneity. Some OSNs have a huge number of users; they maintain a huge number of relationships and data. One user can be registered on multiple OSN sites at the same time. Figure 1.3 shows the interconnectivity of OSNs and their users.

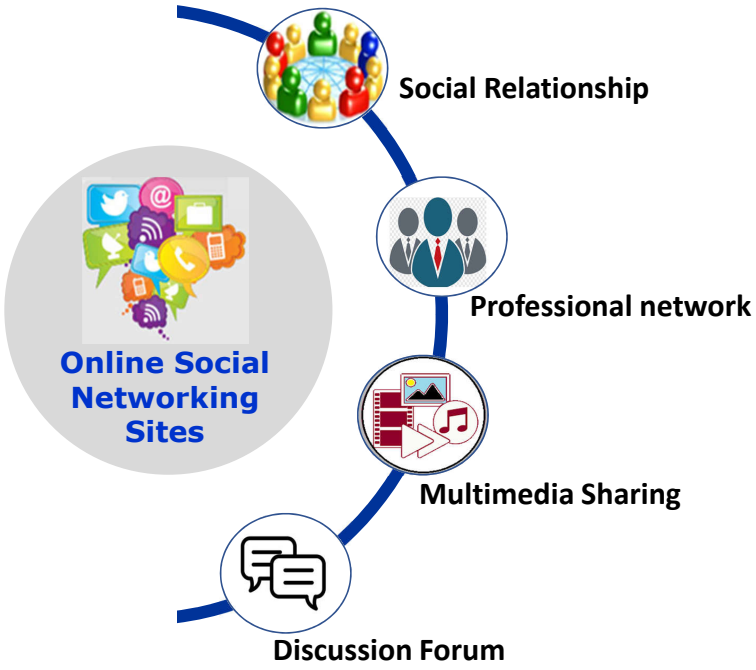

Figure 1.4: Types of online social networks

1.1.4 OSN types

Based on the uses of an online social network, it can be classified in one of many different types. An OSN may belong to one of four general classes: 1) social relationships, 2) professional networks, 3) multimedia sharing, and 4) discussion forums. These four types of network classification are displayed in figure 1.4.

1) Social relationships: People use these types of networks to build and maintain social relationships on the online platform. Facebook, WhatsApp, MySpace, Hi5 etc. are some examples of such groups.

2) Professional networks: To build or improve their professional career, people use such online platforms. LinkedIn, Meetup, Xing, etc. are some examples of such OSNs.

3) Multimedia sharing: These types of OSNs are used only to host and share photos, videos, and other multimedia content. Some examples are Flickr, Google Photos, Dropbox.

4) Discussion forums: These types of networks are extremely community-based where people looking for information or discussions about specific questions, topics, and ideas. Reddit, Quora, and Twitter are some popular discussion forums.

1.1.5 Data in OSNs

Boyd and Ellison [29] described that there are mainly two types of user related data in OSN: 1) Profile and 2) connection.

1. **Profiles**: A profile in an OSN is a representation of a user in the corresponding social network.

2. **Connections**: Different types of users like colleagues, friends, relatives, fans, etc. are connected to each other in an OSN through connections. A set of such connections can be used to form a graph.

However, based on the several features provided by OSNs, the information relevant to users can be categorized as follows.

- **Messages**: Message is an important part of the OSN. The information exchanged among users or groups of users is called messages. The interaction among the OSN users has been considered a primary source of information on the corresponding social network.

- **Multimedia**: The information exchanged among the users in an OSN may contain multi-media messages. Also, this information can be uploaded on or shared with some public data spaces such as photo albums, Facebook "Wall", blogs, etc. along with private ones. Some examples of multi-media messages are audio clips (music, voice recordings), videos, photos, etc.

- **Tags**: A tag is a keyword that can be associated with the information they are sharing on the social network. It allows OSN users to link other users or entities into a discussion or information shared on the underlying social networks. As an example, in the case of Facebook, users can tag himself/herself or other users in a photo.

- **Preferences**: A few OSNs offer users various types of matching or suggestion or recommendation-related functionality for either peers or content. Usually, users explicitly select their preferences, so that their private content can not be visible publicly. Users' preferences may be derived implicitly from the behaviour of users.
- **Groups** - A collection of OSN users form a group for sharing common interests, attributes, resources, or privileges. A collaborative document, common backgrounds or preferences, or sharing of a common space can be an example of sharing resources.
- **Behavioral information** - information about the browsing history or the various types of actions that are performed by an OSN user within the OSN environment is categorized as behavioural information. Benevenuto and others [21] mentioned that this type of meta-data is specifically very rich with information. Information such as friendships, user preferences, some implicit data like the physical location of a user can also be deduced from it. Although the navigation of a social network is not related to the behaviour of a user in traditional networks, behavioral data is also available on traditional websites.
- **Login credentials** Most OSN platforms mandate login credentials to enter and use the OSN services. These login credentials are included with user ids, passwords, mobile numbers, or some other secure information. Traditional websites also have such login credentials.

1.2 Privacy and Security Issues in OSNs

OSN is developed primarily to share and display various non-sensitive data available to social friends. However, through posts, photos, videos, etc., naive users may share a lot of private and sensitive information unknowingly and unintentionally. For example, along with a shared photo, users may share current location information and other metadata. For their own commercial purpose, this collective set of data may be accessed and used by OSN service providers. Moreover, OSN service providers may hand over this user-sensitive information to adversary users, who uses to leverage and invade the privacy of an individual [164]. One the one hand, information must be retrieved for benevolent purposes and on the other hand, data have to be kept secured and private. These are two opposite demands and therefore, need significant research effort.

Privacy of a piece of information can be defined as difficulty to recognize an individual or a group or an organization from some information disclosed about them. An online social

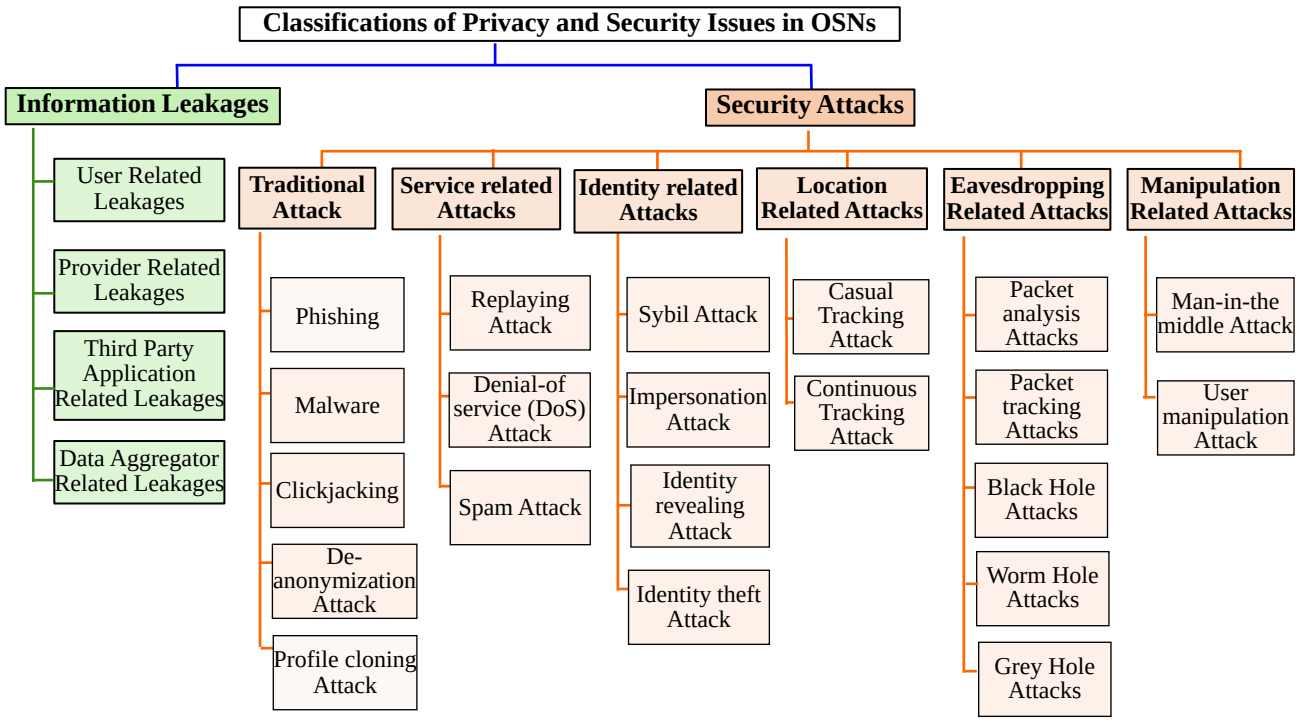

Figure 1.5: Classification of privacy and security issues in OSNs

network is an environment made up of several types of stakeholders with different privacy concerns. The primary stakeholders of any OSN are its users, who enjoy the service; another important stakeholder is the OSN service providers, who provide the service. Without these two major entities, many others such as third-party application providers, data aggregators, researchers, and advertising agencies are also in the fray. The privacy issues begins with the gap between the user's intention to disclose his or her private information and the actual revelation of his or her privacy. It has to be borne in mind that privacy concerns of an individual user may vary. Sometimes, in reality, a person may disclose more private information than their concerns. But the OSN environment brings much more security hazards than their expectation [23].

There are various types of privacy and security threats in an OSN. All these security issues can be classified into two main categories – 1) Information Leakages and 2) Security Attacks, which can be further divided based on their nature as depicted in Figure 1.5. From this large set of privacy and security issues, I have selected the following threats in the scope of this work.

1.2.1 Privacy and security issues in location sharing

Nowadays, the development of GPS technology empowers a mobile user to avoid manual check-in process and facilitates the mOSN users to share, update and inform current mobile locations very easily. People can access mOSN applications anywhere and anytime. These capabilities, such as global position system (GPS) receiver, sensing modules (cameras, sensors, etc.), and multiple radios (third/fourth generation cellular, WiFi, Bluetooth, WiFi Direct, etc.) enable mOSNs to enhance conventional social networks with additional features, such as location-awareness [199], location-based service, the ability to capture and tag media [90]. In general, built-in GPS is not that much available on laptops. Moreover, it does not exploit 4G or the current standard of cellular networks. Hence, location-based services cannot be easily accessed using laptops[85].

Location sharing through mOSN may end up in catastrophic consequences, especially when privacy and security measures are not implemented properly. On the one hand, the popularity and usage of mOSN based applications are increasing every day. On the other hand, different malicious users and attackers continuously engineer innovative attacks to unlawfully access and modify various social and physical information of the registered mOSN users. Although modern smartphones have privacy and security-based location sharing features, those services require improvements in the security aspects. Moreover, in current systems, location sharing to a large number of friends may incur substantial security hazards which are as follows.

First, although popular online social networks provide many facilities for social life, direct and indirect location sharing increases the risk of user privacy breaches. Some studies have attempted to address location privacy issues in mOSNs [120], [115], [116]. Recently, H. Li et. al presented empirical research to quantify private information leaking issues arising from location sharing in popular OSNs such as Facebook and Twitter [112]. They conducted a three-week experiment with 30 real-world participants and discovered that direct and indirect location sharing by popular OSNs could disclose 16% and 33% of the true points of interest (POIs) of the users respectively. An external adversary was able to infer the demographics (e.g., gender, age, education) after monitoring the disclosed users' location profiles. They also implemented such an attack in a large real-world dataset involving 22,843 mobile users [112]. Many popular social networks provide location-based sharing functionalities like geolocation tags and check-in services. Attackers can easily acquire the location information of users by crawling the social networking sites' data and extracting their point of interest (POI) from the collected data [115], [131].

Second, it is possible for a privileged insider to execute location spoofing intentionally,

providing fake locations on the location-based features of Facebook, WhatsApp, and Snapchat (e.g. Nearby Friends and Snap Map). This is done using downloadable apps like FakeGPS, in order to deceive social friends for malicious purposes [128].

Third, sharing location information is less safe especially when a person has a large number of friends or followers whom he/she might not actually know. Location sharing and friend's location queries should be done on a restricted basis where the communicating parties can limit the distance threshold by which they can find each other.

1.2.2 Security issues related to DDoS attack

One of the heavily exploited security threats in mOSNs is DDoS attack. In this attack, adversaries attempt to overburden the server or the network through a flood of fake login requests, messages, etc. so that the service providers fail to respond to a valid user for a stipulated time interval.

In a DDoS attack, the attackers send a large number of service requests to the service providers that can overload the server and the server denies the request for acceptance [77]. The goal is to overwhelm the network with huge traffic than the network or the server can accommodate. DDoS is a continuous threat faced by businesses of all types, regardless of target market or geographic location. The attacks are gradually becoming more complex in nature that often combines various DDoS methods to execute larger, catastrophic assault. Between January 2020 and March 2021, DDoS attacks increased by 55% and are becoming more complex, with 54% of incidents using multiple attack vectors [185]. Cisco's analysis indicates a consistent growth in the actual and predicted number of DDoS attacks (in millions) from 2018 to 2023 [12].

1.2.3 Security issues related to phishing attacks

In OSN, the connectivity between users is built with trust. Using this faith relationship, attackers can promote spam, malware, phishing attacks, etc. very easily over the social network. Attackers actually use several types of policies, such as deploy programmatic community bots, create fake OSN profiles, build fake websites and steal credentials of OSN accounts [151], [125], [40], [8].

A very popular classical technique of stealing information on the web is by phishing. First, attackers develop a website that looks almost identical compared to the original website. Next, this fake link will be posted on the OSN user's wall [180]. When an OSN user opens that URL

and enters its credentials, attackers will collect that personal or sensitive information, banking details, credit card information, etc. [162], [38]. Finally, the attacker uses those victim users' credentials to login into the original website.

The first popular phishing attack occurred in 1990 on America Online (AOL) Network Systems [176]. To create a new account and access the AOL resources, a user needs to provide his or her credit card information. Attackers had created a fake AOL website and using that had stolen the credit card information. Since then, phishers have targeted online banking services, online trading, and the most popular OSN websites.

According to the published "State of the Phish Report" in Wombat, the rate of attacks has increased in 2018 compared to 2017 [154]. As per the "Internet Crime Report 2020" of the Federal Bureau of Investigation [2], phishing is the most popular crime type on the internet. It has increased by more than two times from 2019 to 2020.

Phishing attacks are a crucial problem in current mOSNs, using some malware or normal keystrokes the phishers steal the personal credentials of the users. Securing mOSN against phishing and other security attacks has become a crucial challenge for researchers.

1.3 Motivation and Objective

Among the various challenges of using online social networks, privacy and security issues are of prime importance. A mobile-based online social network (mOSN) involves the interaction between users with similar interests and objectives through their mobile devices and/or tablet within that virtual social network.

Location sharing through mOSN may lead to explicit and/or implicit privacy and security breach. Implementation of a secured, privacy-preserving location sharing strategy, sustaining the modern-day facilities of various mOSN applications remains a serious research challenge.

Previous research approaches considered a social network server (SNS) for storing social network data and the location-based server (LBS) for storing location information as two separate entities. In order to achieve efficiency, the communication cost between the social network server (SNS) and the location-based server (LBS) should be as little as possible. Moreover, less message exchange would give an attacker less exposure to execute attacks in a wireless public channel. The Location sharing mechanism should not depend on a third-party location-based server. This should be done to minimize the chance of privacy leakage and to minimize the establishment cost. The location-based server (LBS) must not be able to discover the topological structures of users' social networks. By collusion with the social

network server, LBS should not be able to reveal users' social information.

In OSN, DDoS attacks are observed to be both fast-spreading and unpredictable in nature. For launching a DDoS attack, an adversary needs to capture a request message and sends the same message to the mOSN server repeatedly so that the server becomes overburdened and fails to provide the expected service. Till now, many efforts have been made for designing authentication protocols for different cloud server applications. However, most of such protocols are vulnerable to DDoS attacks, which occurred due to loss of synchronization between participants. Furthermore, to rebuild synchronization between the participants, a protocol may need to compromise the un-linkability property. Therefore, designing an authentication protocol for resistance to DDoS attacks is still a challenging research problem.

In mOSN phishing, phishers create a phishing hook, which is a website, an identical copy of the authentic website. Phishers use this hook to steal the login or other sensitive information of a victim user. While a victim user enters his or her credentials in a hook website, attackers capture that information. If an authentication scheme can authenticate its user without complete login information or with the credentials masked with other parameters, attackers do not acquire the details of login information for promoting attacks.

Till date, several authentication protocols have been developed for the OSN environment to provide secure authentication. But the comparative study of the earlier authentication schemes as discussed in section 3.2 designed for mobile networking environments has disclosed that most of the schemes suffer from security flaws. For the sake of hazard-free, trusted communication through a public wireless medium, an authentication protocol between the user and OSN server is essential. Moreover, a mobile device is operated through limited battery equipment, and the authentication process of any mobile user should be completed with minimum computation and communication costs. While the earlier authentication protocols for mobile networking environments were based on time-consuming computations of cryptographic functions and resource requirements. This compels me to propose new authentication schemes for mobile OSNs that can be efficient and provide complete security for the system avoiding time-consuming resource-intesive computation.

1.4 Summary of contributions

The contributions of this thesis are summarized in this section.

1.4.1 Privacy-preserving efficient location-sharing scheme for mobile online social network applications

The first contribution of the thesis is to design a privacy-preserving, secure and efficient location sharing scheme for mobile OSNs, which is both efficient and flexible during location update, sharing, and queries of social friends and strangers. According to the proposed scheme, a mobile user and location-based social network server, first, separately establish a shared symmetric session key with a cellular tower. All location updates and friend's location query messages are encrypted with this session key before transmission. Because of this end-to-end encryption, an adversary has little chance to reveal the location information of a registered user. Furthermore, unlike the location-based services of existing OSNs, my proposed scheme allows a user to decide on a distance threshold, up to which he/she wants to make himself/herself visible to social friends. This imposes a much better user-controlled restriction on location sharing, as unrestricted location sharing can lead to security vulnerabilities. The security of the proposed scheme is validated using random oracle-based formal security proof and Burrows-Abadi-Needham (BAN) logic-based authentication proof, followed by informal security analysis. Additionally, I have used ProVerif 1.93 to verify the security of the system. The efficiency and practicability of the proposed scheme are demonstrated through experimental implementation and evaluation.

1.4.2 DDoS attack resisting authentication protocol for mobile based online social network applications

In the second contribution of this thesis, I have aimed to propose a secure and lightweight authentication scheme ($PRDoS$) that resists DDoS and other security attacks in mobile OSN environments. I provide a multi-faceted solution towards the remedy of DDoS attacks in the OSN environment. After a certain threshold, the scheme discards further user login attempts and blocks an adversary who intends to overload the network server. I use the preloaded shadow identity and emergency key pairs, and a key-refilling strategy that rebuilds the essential synchronization between a blocked naive user and the OSN server. This technique restores the intended un-likability property of the protocol. Using NS3 simulation, I have studied the impact of DDoS attackers on network throughput and network delay. Moreover, I have validated and compared the proposed scheme against state-of-the-art solutions using real attacks and benign datasets. I have used the Canadian Institute for Cybersecurity (CIC) DoS dataset 2017, which is generated by capturing the normal and DoS attack packets separately

with subsequent pre-processing for testing. I also use machine learning (ML) algorithms, such as K-Nearest Neighbor (KNN), Gaussian Naive Bayes, and Multilayer Perceptron (MLP) to demonstrate the performance of the proposed solution in a practical attack detection scenario. I observe that these algorithms provide 97.05%, 95.48%, and 96.6% DDoS attack detection accuracy, respectively.

1.4.3 An efficient user authentication scheme for phishing attack detection in mobile online social networks

In the third contribution of this thesis, I have proposed a secure and lightweight cryptography-based authentication scheme, called ASPA-mOSN that provides resistance to phishing attacks in OSNs. I mainly targeted to resist phishing attacks in the mOSN environment but my proposed ASPA-mOSN can withstand several other security attacks such as denial of service attacks, replay attacks, man-in-the-middle attacks, etc. The security of the proposed scheme is explained using both informal security analysis, and formal security analysis through the widely-recognized Real-Or-Random (ROR) model and ProVerif simulation tool. Furthermore, using BAN logic I have proved the mutual authentication of the proposed system. Finally, I compare the security, functionality, computation, and communication costs of the proposed ASPA-mOSN with related schemes. The comparison results show that ASPA-mOSN outperforms other existing competing schemes.

1.5 Organization of the thesis

The chapters of this thesis are organized as follows:

In **Chapter 1**, a detailed description of Online Social Networks(OSN) is presented. In this chapter, a formal definition, characteristics and network structure of OSN are also presented. I have also discussed the privacy and security issues in OSNs. The motivation and objective of this research work are also presented. The contributions of this research work are also summarized in this chapter.

In **Chapter 2**, I have discussed the mathematical preliminaries used in this thesis. The fundamentals of biometrics verification including biohashing and fuzzy extractors are also presented. I then discuss the Chebyshev polynomial and chaotic map and its properties. Next, I discuss on fundamentals of bilinear pairing and attribute-based encryption. Finally, I

discuss the BAN logic and its properties.

In **Chapter 3**, I have given an overview of the related works on privacy and security issues of location sharing in various mOSN applications describing three different streams of existing solutions that were adopted for solving those issues, their merits, and demerits. I have also discussed the related works on Distributed Denial-of-Service (DDoS) Attacks, and phishing attacks in mOSN environments describing the pros and cons of those existing solutions.

In **Chapter 4**, I have discussed a new location sharing scheme for mobile online social network applications. The proposed system adopts a model, where the location-based server (LBS) and the social network server (SNS) are integrated into one single entity. To share privacy-preserving locations, the proposed scheme exploits dummy locations, a dedicated mapping protocol among the Cellular Tower (CT) and a set of location-storing social network servers. I have validated the security of the proposed scheme using Burrows-Abadi-Needham (BAN) logic-based authentication proof, followed by an informal security analysis. Additionally, I have provided simulation results of security verification using ProVerif 1.93 tool. Compared to all recent schemes, the proposed scheme incurs much low computation and communication overheads.

In **Chapter 5**, I have discussed a secure and lightweight DDoS attack resisting authentication scheme ($PRDoS$) for mOSN applications. The proposed ($PRDoS$) scheme requires low communication and computation costs compared to other existing related schemes. Using NS3 simulation, I have shown the impact of DDoS attackers on the proposed authentication scheme. I also have validated and compared the proposed scheme against state-of-the-art solutions using real attacks and benign datasets, like Canadian Institute for Cybersecurity (CIC) DoS dataset, 2017. I have used the machine learning algorithms, like KNN, Gaussian Naive Bayes and Multilayer Perceptron (MLP) to demonstrate the performance of the proposed scheme in a practical attack detection scenario.

In **Chapter 6**, I have proposed a secure and lightweight cryptography-based authentication scheme, called ASPA-mOSN, that provides resistance to phishing and other related attacks in OSNs. The security of the proposed scheme is explained using both informal security analysis, and formal security analysis through the widely-recognized Real-Or-Random (ROR) model and ProVerif simulation tool. The comparison results with related schemes show that $ASPA-mOSN$ is more secure, efficient, and outperforms other existing competing schemes.

Finally, **Chapter 7** concludes the thesis by providing a summary of contributions, highlighting its limitations and outstanding issues and suggesting some directions for possible future research work.

Chapter 2

Mathematical Preliminaries

In this chapter, I have described some fundamental mathematical preliminaries that are used to describe and analyze the proposed schemes in Chapters **4 - 6**. In section 2.1, I have described the fundamental concept of the collision-resistant one-way hash function. I have briefly discussed the properties of the Chebyshev polynomial along with the chaotic map-based discrete logarithm problem in Section 2.2. The fundamental concepts of biometrics verification and fuzzy extractor function are described in Section 2.3. Bilinear pairing is described in Section 2.4. To prove the mutual authentication of the proposed protocols, I have applied the Burrows-Abadi-Needham (BAN) logic. Hence, the basic notations and logical postulates of BAN logic are discussed in Section 2.5.

2.1 Collision-resistant One-way Cryptographic Hash Function

One-Way Collision-resistant hash function h accepts a string of 0 and 1, say $b \in \{0,1\}^*$ as its input and returns another fixed length binary string $h(b) \in \{0,1\}^i$ as its output where i is a fixed number. The one-way hash function can be correctly described as follows [160].

Definition 2.1. *The advantage probability for finding any collision of the hash function by an adversary $\mathcal{A}$ within the execution time t_n can be defined as $Adv_{\mathcal{A}}^{hash}(t_n) = Prob[(x,y) \in_R \mathcal{A}$: $[(x \neq y \text{ and } h(x) = h(y)]$. If $\mathcal{A}$ has selected a random pair (a,b), using (ϵ, t_n), it is denoted that adversary $\mathcal{A}$ requires t_n as execution time while attacking the collision resistance of $h(\cdot)$ and $Adv_{\mathcal{A}}^{hash}(t_n) \leq \epsilon$.*

2.2 Definition and Properties of Chebyshev Polynomial

The Chebyshev polynomial $T_n(x)$ of degree n is defined as [100]:

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

The following recurrence relation is an alternative formulation for the same polynomial.

$$T_n(x) = \begin{cases} 1 & \text{when } n \textit{ is equal to } 0 \\ x & \text{when } n \textit{ is equal to } 1 \\ 2xT_{n-1}(x) - T_{n-2}(x) & \text{when } n \textit{ is } \geq 2. \end{cases}$$

Definition 2.2. *The semi-group property of the enhanced Chebyshev polynomial holds on the interval* $(-\infty, +\infty)$*. It is formalized in the following manner [201].* $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \pmod p$*.* p *is a large prime number. Here,* $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x)) \pmod p$*, where* Z_p^* *= the set of all integers* x*, suc that* x *is less than* p *and* x *and* p *are co-prime.*

Definition 2.3. *Suppose that* x *and* y *are given and we are asked to find an* m *such that* $T_m(x) = y$*., It is computationally infeasible to find such an* m*. This is also known as the "Chaotic map-based discrete logarithm problem (CMDLP)" [36]. The advantage probability of* $\mathcal{A}$ *to solve CMDLP is* $Adv_{\mathcal{A}}^{CMDLP}(t_2) = Prob[\mathcal{A}(a, b) = m : m \in Z_p^*, b = T_m(a) \pmod p]$.

2.3 Biometrics and Fuzzy Extractor

Several authentication protocols secure their scheme using certain biometric features, such as thump impression, iris etc. as key for their uniqueness property [54], [165]. But there may be various types of uncertainties and noises, or repeated attainments of the same individual may have differences up to certain range. The Fuzzy extractor technique can help us to output the indistinguishable string, if the input biometric vary from the saved up to permissible range.

Using two algorithms $Gen(\cdot)$ and $Rep(\cdot)$ along with five tuples $(\mathcal{M}, l, \kappa, \mu, d)$, the fuzzy extractor is defined.

- The metric space of biometric records is represented by $\mathcal{M} = \{0, 1\}^k$ with finite dimension.
- The similarity of two different inputed biometric $\mathcal{B}_i$ and $\mathcal{B}_j$ is assessed by the distance function $\Delta : \mathcal{M} \times \mathcal{M} \to \mathbb{Z}^+$

- l is used to represent the length of a (in bits).
- μ denotes min-entropy on metric space $\mathcal{M}$ with the probability distribution $\mathcal{W}$.
- κ denotes an acceptable error tolerance.
- d is the permissible extreme statistical distance of two probability distributions $\langle a_1, b\rangle$ and $\langle a_2, b\rangle$.

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

- *Gen*: The function is described as $\langle m, n\rangle \leftarrow Gen(\mathcal{B})$, such as $m \in \{0,1\}^l$ and $\mathcal{B} \in \mathcal{M}$, where the statistical distance of the probability distributions $\langle m, n\rangle$ and $\langle m_1, n\rangle$, $SD(\langle m, n\rangle, \langle m_1, n\rangle) \leq d$. Where, m_1 refers an identical binary string of length l, where $l = \mu - 2\log(\frac{1}{d}) + O(1)$.
- *Rep*: The function is described as : $\forall \mathcal{B} \in \mathcal{M}$, $\forall \mathcal{B}' \in \mathcal{M}$ and $\Delta(\mathcal{B}, \mathcal{B}') \leq \kappa$ such as if $\langle m, n\rangle \leftarrow Gen(\mathcal{B})$, then $Rep(\mathcal{B}', n) = m$.

A detailed discussion on fuzzy extractor is available in [157] and use of biometrics and fuzzy extractor method in authentication protocol becomes very popular nowadays [36], [156].

2.4 Bilinear Pairing

In this section, I have discussed the bilinear pairing and the assumption of the decisional bilinear Diffie-Hellman problem.

Let $\mathbb{F}_p$ is a Finite field over prime order p, $\mathbb{G}_1$, $\mathbb{G}_2$ are two cyclic additive groups of prime order p and $\mathbb{G}_T$ is another multiplicative cyclic group of prime order p. Further, let g_a and g_b are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear pairing is an injective mapping function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following three properties:

- ***Bilinearity :*** For all $X \in \mathbb{G}_1$, $Y \in \mathbb{G}_2$, a, $b \in \mathbb{F}_p$, $e(X^a, Y^b) = e(X, Y)^{ab}$.
- ***Non-degeneracy:*** $e(g_a, g_b) \neq 1$, 1 is the identity in $\mathbb{G}_T$.
- ***Computability :*** There exists an efficient algorithm for computing $e(X, Y)$ for each $X \in \mathbb{G}_1$ and $Y \in \mathbb{G}_2$.

We can say that $\mathbb{G}_1$ can be a bilinear group if computation of the group operation in $\mathbb{G}_1$ and the bilinear mapping $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ both are efficient. It is also noticed that the map e is symmetric since $e(X^a, X^b) = e(X, X)^{ab} = e(X^b, X^a)$ [71].

Definition 2.4 (**Decisional Bilinear Diffie-Hellman (DBDH) assumption**). *Suppose a challenger selects $\alpha, \beta, \gamma, z \in \mathbb{Z}_p$ at random. The DBDH assumption says that no probabilistic polynomial-time algorithm $\mathcal{B}$ can be able to distinguish the tuple $\langle A = g^\alpha, B = g^\beta, C = g^\gamma, Z = e(g,g)^{\alpha\beta\gamma}\rangle$ from the tuple $\langle A = g^\alpha, B = g^\beta, C = g^\gamma, Z = e(g,g)^z\rangle$ with more than a negligible advantage [159]. The advantage is then given by*

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

where the probability is taken over the random choice of the generator g, the random choice of α, β, γ, $z \in \mathbb{Z}_p$ [71].

Definition 2.5 (**Decisional Modified Bilinear Diffie-Hellman (DMBDH) assumption**). *Suppose a challenger selects α, β, γ, $z \in \mathbb{Z}_p$ at random. The DMBDH assumption says that no polynomial-time adversary can be able to distinguish the tuple $\langle A = g^\alpha, B = g^\beta, C = g^\gamma, Z = e(g,g)^{\frac{\alpha\beta}{\gamma}}\rangle$ from the tuple $\langle A = g^\alpha, B = g^\beta, C = g^\gamma, Z = e(g,g)^z\rangle$ with more than a negligible advantage [159].*

2.5 BAN Logic and Its Properties

In general, BAN logic is widely-used tool to analyze the security of the mutual authentication protocol designed for two communicating parties in a network [31]. The BAN logic based model contains the following items, namely, the logic rules, idealized messages, security goals, initial assumptions, declaration, and the anticipated goal. The basic notations used in BAN logic listed in Table 2.1.

The following set of rules will explain the logical postulates of the BAN logic [31], [172]:

- **Rule 1:** Message Meaning Rule (MMR)
  $\frac{R|\equiv Q \overset{Y}{\rightleftharpoons} R, R \triangleleft \langle X \rangle_Y}{R|\equiv Q|\sim X}$.
- **Rule 2:** Freshness Conjuncatenation Rule (FCR)
  $\frac{R|\equiv \#(X)}{R|\equiv \#(X,Y)}$.
- **Rule 3:** Nonce Verification Rule (NVR)
  $\frac{R|\equiv \#(X), R|\equiv Q|\sim X}{R|\equiv Q|\equiv X}$.

Table 2.1: Notations of BAN logic

Notations	Description
$R \mid\equiv X$	R trusts that the statement X is true
$R \lhd X$	R can see the received statement X
$R \mid\sim X$	R once said the statement X
$\#(X)$	Formula X is fresh
$R \Rightarrow X$	R has jurisdiction over the statement X
$\langle X \rangle_Y$	Formula X is combined with the formula Y
$R \overset{K}{\longleftrightarrow} Q$	The key K is secret. It is used for message communication between R and Q. They only know the value.
$R \overset{X}{\rightleftharpoons} Q$	The statement X is secret. Only R and Q know it. Principals trusted by R and Q may know X
$SKey$	Current session key

- **Rule 4:** Jurisdiction Rule (JR)
  $\frac{R\mid\equiv Q\Rightarrow X, R\mid\equiv Q\mid\equiv X}{R\mid\equiv X}$.
- **Rule 5:** Additional inference Rule (AR)
  $\frac{R\mid\equiv(X,Y)}{R\mid\equiv X}$, $\frac{R\lhd(X,Y)}{R\lhd X}$, $\frac{R\mid\equiv Q\sim(X,Y)}{R\mid\equiv Q\sim X}$.

To prove the security of the proposed protocol, the following two goals must hold for the protocol to be secure.
The security analysis of any authentication scheme using the BAN logic can be done using following steps [172]:

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

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

2.6 Summary

In this chapter, I have discussed some mathematical preliminaries on several topics that have been used in the schemes described later. One way hash function, biometric verification with fuzzy extractor techniques, Chebyshev polynomial with chaotic map-based cryptography, bilinear pairing, and attribute-based encryption are included here. I have discussed only those cryptographic techniques which can be used for designing lightweight authentication protocols for mobile Online Social Networks in Chapters **4 - 6**.

Chapter 3

Review of Related Works

In this chapter, I have presented a comprehensive review of the state-of-the-art related works on OSN attacks and their countermeasures. Several academicians and researchers have discussed different types of security threats and their defense mechanisms to protect the privacy of users and the information they have shared over OSN platforms. I have also presented an outline of related works on privacy preservation techniques for efficient location sharing in online social network (OSN) applications, detection, prevention techniques of distributed denial of service (DDoS) attacks in OSN applications, detection, and prevention techniques of phishing attacks in OSNs.

3.1 Privacy and Security Threats in Online Social Networks

In an OSN, people register with their personal credentials, which are saved in the database of the OSN service providers. A person interacts with other users, sometimes with an unknown stranger, clicks or uses third-party games, applications, advertisements, etc. Information can be leaked in any of these interactions. Using data crawling applications, users' data can be collected from social networking sites, security problems also can be arisen by linking the users' profiles from multiple OSN services. Moreover, OSN service providers sell their users' data for business profit. As a result, various types of security attacks such as Phishing, Sybil, spam, DDoS, etc. can happen or a user may be a victim of any malicious security hazard at any time.

In this section, I have mainly discussed about various types of security threats that can be incident in the OSN. I have classified all possible privacy and security issues into two main

categories – 1) Information Leakages and 2) Security Attacks, they can be further divided based on their nature which is depicted in Figure 3.1. For better readabiity, the same figure has been given again. Recently some other social threats are emerging in the OSN platform.

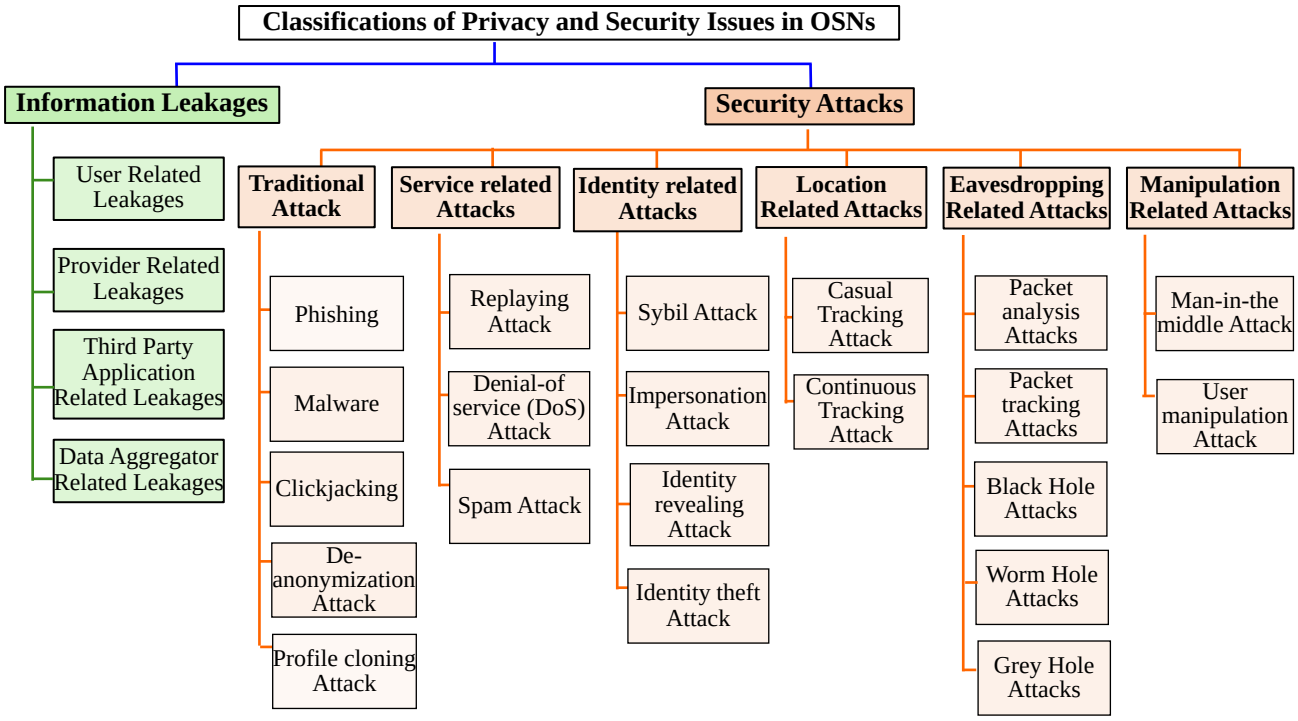

Figure 3.1: Classification of privacy and security issues in OSNs

I have categorized them as modern social threats.

3.1.1 Various types of information leakages

In this part, I have discussed the various types of information leakages that can happen in OSN. There are several types of information disclosing, I broadly categorize them into four different types.

1. **User Related Leakages:** A user can assume falsely that some information is kept private, but in reality, it is not. When people interact with other OSN users, especially with strangers or rarely associated, or sometimes users may be application robots [88] not human beings. Crowdsourcing employees interact and stroll with OSN users for malicious purposes [169]. Other OSN users can post or disclose the information of

the user. Therefore, the protection of users' information from other OSN users is a challenging task.

2. **Provider Related Leakages:** OSN service providers have total control over the information provided by the user towards the service provider. At the time of registration, people need to agree with the Terms and Conditions of Service documents if he or she wants to use the service. Most of the user either does not have any idea about the copyright form [62] or don't have any other options to use the facility. Therefore, the misuse of OSN users' personal information can be viewed as a breach of trust, and for hiding the personal information of the user several solutions have been proposed.

3. **Third-Party Application Related Leakages:** users sometimes interact with various third-party applications that are available in their OSN profiles. For smooth interaction between a user and those external applications, service providers provide an interface for accessing users' data. Sometimes, service providers provide more data than the requirement to those third-party applications. Users' privacy comes at a risk as those applications can use the users' personal information for unwanted purposes [60].

4. **Data Aggregator Related Leakages:** Several professional data crawling applications are available for OSNs. Those agencies collect and build the database from the publicly available information of users and sell those databases to researchers, background-checking organizations, insurance companies, or any other agencies [28]. People have different user profiles in several OSNs, such as social information on Facebook, professional profiles on Linkedin, etc. Those crawling data further can be merged, and complete information about a user can be accurately deducted for mischievous means.

3.1.2 Various types of security attacks

Several types of security attacks that can be possible in Online Social Networks are discussed in this sub-section. Depending on their nature of attacks or targeted sets, security attacks of OSN [61], [5] can be broadly classified into six different categories – 1) Traditional attacks, 2) Service related Attacks, 3) Identity related Attacks, 4) Location Related Attacks, 5) Eavesdropping Related Attacks, and 6) Manipulation Related Attacks as depicted in Figure 3.1.

1. **Traditional Attacks:** Some attacks are performed using classical techniques. Phishing, malware are some well-known traditional attacks for stealing users' personal information.

Obtaining those personal credentials such as user id, password, mobile number, bank details, etc., an adversary can commit several types of crime.

Though researchers have addressed [151], [192] these attacks earlier but with the development of OSNs, the effect of these attacks can propagate more easily and quickly than earlier. Some of those serious threats are discussed here.

Phishing: Phishing is one type of fake attack where the adversary obtains the user's personal data using a third-party or fake identity. For phishing attacks in OSN, attackers collect the information of the users and select the victim users from them. Attackers create a fake website that looks like the original website [25] and send a bogus message containing that phishing URL or post that link on the wall of the victim users. When a user clicks that link, the phishing website will open and the user provides its information assuming that the website is original one [11]. Attackers steal personal information of that victim user in this manner.

Malware: A malware is a malicious program code such as computer viruses, worms, etc. As OSN follows a graph-like structure where users are the nodes and connections are the links between them; a malware can easily propagate through those links [134]. Most OSNs do not have a suitable mechanism to identify whether an application or a link is malicious or not. When a user clicks or opens such a compromise link or application, it spreads the malware to steal a user's credentials from his or her computer or mobile [59],[196].

Clickjacking: Clickjacking, also known as the User Interface redress attack, is an emerging threat to OSN, where a malicious program is hidden behind the labels, pictures, or buttons in the website [153]. When an OSN user clicks on that portion of the web page, the user's information is revealed and the control of the computer goes to the attackers. There are different types of Clickjacking; most common forms are Cursorjacking and Likejacking. The most popular form is Likejacking [92], where attackers append malicious codes with the "Like" button of a Facebook page. Cursorjacking is a UI redressing technique to transfer the cursor from the location of the user control to some bogus website developed by the attackers. Using a clickjacking attack, an attacker can spy the users' activity by controlling the computer hardware of the user, such as a microphone, webcam, etc. [124].

De-anonymization Attack: Some OSNs, like Twitter, Facebook use several anonymization techniques to protect their users' identity and privacy by assigning

a fabricated ID. In a De-anonymization attack, attackers using some data-mining strategy, such as network topology, group membership of a user, re-identify a person from the set of anonymous data. A huge platform of data sharing, contact sharing, content searching is provided in OSN. As the nature of this sharing is by default public, OSN becomes a soft target for de-anonymization attacks [51]. Several techniques of de-anonymization attack have been proposed by the researchers, attackers can easily use them to perform these types of attacks in OSNs. Gulyás et al. [74] proposed a robust and precise technique to perform a de-anonymization attack on OSN data. Ghazinour et al. [69] have shown that a user's identity can be exposed easily by tracking session cookies. By proposing a new technique of de-anonymization attack, Wondracek et al. [190] demonstrate that by analyzing the group membership data, an attacker can identify a user easily in the OSN.

Profile Cloning Attack: In a profile cloning attack, attackers clone an existing user's profile either within the same OSN or across different networks using the stolen personal information of the original user. After creating that fake profile, attackers send friend requests and try to set up a trust relationship with the contacts of the original user. By collecting the sensitive information of the friends', attackers perform various types of web fraud, such as cyberstalking, cyberbullying, etc. [101]. These types of attacks are also called identity clone attacks [96].

2. **Service Related Attacks:** In some types of attacks, attackers target to hamper the service of the OSN for which it is developed.

   Replaying Attack: In replaying attacks, when a message with some authentication information is communicated between a user and a service provider, attackers listen and copy the message from the channel. When the communication is over, the attacker fraudulently re-sends the same message again to the receiver and tries to pretend itself as the original user and ask for the service. If the attacker becomes successful to commit a replaying attack, the service would be unavailable to the original user [19], [61].

   Denial-of service (DoS) Attack: Attackers launch denial-of-service (DoS) Attacks either by interrupting message routing, or blocking the file servers, or overloading the service by sending a large number of service requests so that the service provider fails to provide services [127]. Just like any other popular network, OSNs also suffer from such types of attacks.

Spam Attack: Spams are unwanted messages or contents that come to an authentic user as a wall post [67]. Spam harms resources and disturbs interactivity between users. Spams generally contain commercial advertisements or bogus links that can redirect to malware or phishing website. OSN spams are more harmful compared to earlier email spam, as that can spread more quickly through the trust path among social friends. Spam messages are mainly sent from a compromised profile or social bots [58], [72]. Most of the time, the compromised profile is generated with the name of a famous person [65]. OSN services are being disrupted by spreading malware using spam attacks.

3. **Identity Related Attacks:** The types of attacks are based on the identity misuse of authentic users.

Sybil Attack: In Sybil attack, attackers generate a large number of fake profiles to influence the outcome of any service [200]. Sybil attacks are a serious problem for OSN security. As a huge number of users are connected with a social chain, it is very easy to create and manage several fake profiles. By handling such multiple numbers of a fake account, attackers can control the result of the e-voting system[178], or can influence the search results of some media, can boost the reputation of some organization [152], etc. Sybil attacks can decrease the reputation of any organization, outvote genuine users or damage information, and illegally increase their power in the OSNs [136].

Impersonation Attack: Using an impersonation attack, an attacker can masquerade itself as a trusted entity in the OSN. At the time of registration, an attacker can capture the original identity of the user and pretend itself as a legitimate user and avail service. It is known as User Impersonation Attack [182]. During the authentication phase, an attacker can masquerade itself as a genuine service provider and respond to the login message of any real user. The user receives a message which appears the same as from a legitimate service provider, but when they reply to it with sensitive information, attackers capture and misuse that information [18].

Identity Revealing Attack: It is a different version of Impersonation attack, where after stealing the information of an individual user, attacker reveals the identity of the user [103], [123]. Due to these privacy attacks, young users face online harassment, cyberbullying, sexual solicitation, and disclosure of problematic information on social networking platforms.

Identity Theft Attack: In an identity theft attack on OSNs, attackers gather identity-based information of individual victim users and use that identity information either to achieve some benefits or hurt the victim user [81]. These attacks can be performed bBoth from inside or outside the OSN. A genuine user of the OSN maliciously behaves and launches insider attacks. Whereas, a malicious outsider or a third-party application launches outside identity theft attacks.

4. **Location Related Attacks:** Some online social networks (e.g., Facebook) provide an option to share users' current location information on their wall. Based on that information attackers [189] launch several types of location-related attacks [111].

   Casual Tracking Attack: OSN users share their real location in their profile wall; sometimes, intentionally or unintentionally, they leave that information public. An attacker generates a location profile of a victim user from his or her publicly shared location history. Attackers can launch the casual attack anytime and it stays for an indefinite period. Sometime, attackers can even harm physically [173], [203]. These types of attacks are quite common in some OSNs, such as Facebook, Renren, Foursquare, etc.

   Continuous Tracking Attack: Several OSNs provide a facility to its users to share their location indirectly [115]. Instead of disclosing the users' current GPS location, WeChat, Skout, such OSNs show the distance between two OSN friends. Attackers track and collect the continuous location of the user for the time they are using those applications. From this data, attackers can get the whereabouts of the victim user, demographic information, identify the OSN user physically [52].

5. **Eavesdropping Related Attacks:** In eavesdropping attacks, attackers take the advantage of unsecured network channels and steal the information communicated between the user and the OSN, and perform some harmful activities such as modification, rejection, etc [197].

   Packet Analysis Attacks: In OSN, attackers capture the packets between the user and the service provider. Attackers try to extract personal information by analyzing those packets and initiate some other attacks, e.g., successive-response attacks, message tampering. Sometimes, attackers can perform impersonate attacks by pretending itself as a legitimate user of the network by stealing the information of a victim user [155].

Packet Tracking Attacks: In OSN, after eavesdropping attackers trace the source and destination address of a packet, and tamper the addresses. Often that packet cannot be recovered [155]. In this type of attack, attackers also change the routing information.

Black Hole Attacks: In black hole attacks, attackers create a communication tunnel where the attackers eavesdrop on the OSN messages through that tunnel. The routing protocol would be fully under the control of the attackers when more than one attacker initiates this attack at the same time. In OSN two more varieties of these attacks are available – Wormhole attacks and Grey hole attacks.

6. **Manipulation Related Attacks:** In this type of attack, attackers psychologically manipulate some tricks so that users make some security mistakes or provide their personal information.

   Man-in-the Middle Attack: In a man-in-the-middle attack, attackers place themselves between an OSN user and OSN servers. This type of attack is mainly initiated for hacking public keys between sender and receiver. An attacker controls the communication messages; he or she intercepts the messages communicated between the user and the server. Then the attacker injects other information or steals the secure information from that message. M. Conti et. al. in their study about man-in-the-middle attack [44] have discussed, that this type of attack is vulnerable and compromises integrity, availability, and confidentiality.

   User Manipulation Attack: In a user manipulation attack, attackers investigate the OSN and select a victim user by gathering required background information. An attacker obtains a victim user's trust and sends a message having false information about some social events. When the victim user accepts the message and provides its personal information, the attacker hacks that information.

3.1.3 Modern social threats

In modern social threats, an adversary maliciously provokes and tracks the social relationships of the OSN users. Attackers specially target the young users, mostly minors of the OSN sites. They draw the attention of the teenagers by expressing care, sympathy, love, and presenting an online offer for gifts, etc. The underlying interests of the attackers are spying on cyber harassment and blackmailing.

1. **Cyberstalking Attack:** In these types of attacks, the adversary harasses a victim user by exploiting his/her personal information like contact number, address, the current location which have been shared on social networking platforms. An adversary blackmails a victim user by sending repeated messages or voice calling in OSN [56]. Adversary collects the location information which is disclosed by the user and launches the treacherous cyberstalking attack. Cyberstalking is a serious threat to the OSN platform and affects the mental health of the users.

2. **Cyberbullying and Cyber-grooming Attack:** It is a modern practice of online harassment where the adversary uses a digital platform, nowadays on OSN, to track a victim user. Cyberbullying happens when a teenager repeatedly harasses others or encourages others to do the same on social networking sites. Cyber-grooming happens when an adult establishes a relationship with a minor and sexually harasses or abuses them. Teenagers are extremely vulnerable to these types of cyber-attacks due to their immaturity [53]. Teenagers suffer from depression when an attacker traps them. According to security experts' view, thousands of students have been victimized by such attackers all over the world [57]. The most shocking cyberbullying example was the case of Megan Meier, where a teenage girl committed suicide [143].

3. **Corporate Espionage Attack:** It is a form of surveillance conducted through some automated social engineering attack for commercial purposes [104]. It is also known as Industrial espionage. Using the OSN platform, the sensitive information of any employee, such as current designation, full name, email id, etc. can be stolen and collected by executing these attacks. Attackers exploit the victim by maliciously spreading that information.

3.2 Research Issues in Mobile Online Social Networks

Over the last decade, in order to establish social relationships and share their common interests, millions of users are increasingly joining various mobile-based online social networks like Facebook, LinkedIn, Twitter etc. [29]. With the massive advancement of communication and smartphone technologies, the traditional Online Social Networks have entered into the generation of mobile-based online social networks (mOSNs). Smartphones provide the flexibility for an easy-access to mOSNs applications anytime and anywhere [91]. The presence of various user-friendly features of smartphones like cameras, sensors, GPS etc. facilitates

mOSN to become more popular than traditional OSNs.

The data available from the U.S. [30] indicates that the users have changed their connectivity habits and lean less on Wifi. The Wifi technology plays an important role in mobile connectivity in case of the data intensive activities, such as streaming video or gaming. The usage dynamic can be completely different in less-developed places, where the Wifi technology is not as readily available for home and other users who own a PC, laptop or tablet. As a result, a smartphone is more used as a go-to device for the Internet access. The data available in [34] says that as of January 2021, video apps accounted for "66.2% of the global mobile data usage every month", whereas the social networking accounted for "10.1% of the global mobile data volume. It has been observed that the users can watch the videos using both "video applications" and "social networking applications".

In general, at the time of installation, mobile applications request access permissions for sharing several personal information, such as contacts, device locations, microphone, camera, etc. However, naive mobile users rarely read the official privacy policies. OSN users often fail to understand the security vulnerabilities and adverse impacts of mobile apps. As studied by Hayes *et al.* [80], sometimes mobile apps collect a huge amount of users' information with the help of Big Data tools and technologies, especially for marketing purposes. Often, these are conducted without the consent of users, or beyond the limit of the privacy policies.

mOSNs have huge security concerns and it is more insecure compared to the situation when OSN is accessed through browsers from traditional computers like desktops, laptops [106]. Posting any sensitive information from a mobile device is more insecure. However, the incidence of various kinds of security attacks in mOSNs is growing at an alarming rate. This includes several active and passive security attacks like phishing, malware, graph-based attacks, Sybil attack etc. One of the heavily exploited security threats is DDoS attack, which is an attempt by the adversaries to overburden the server or the network through a flood of fake login requests, messages, etc. so that the service providers fail to respond to a valid user for a stipulated time interval.

The following table summarizes security and functionality features of existing schemes.

Table 3.1: Comparison of security and functionality features of existing schemes

Security attribute	C. C. Lee *et al.*[109]	X. Li *et al.*[118]	Tsai-Lo [179]	Irshad *et al.*[89]	H. Wang *et al.*[184]	Zhang *et al.*[202]	He *et al.*[82]	Qu-Tan[147]	Challa *et al.*[35]	Li *et al.*[117]	Banerjee *et al.*[17]	Saeed *et al.*[158]	Moghimi *et al.*[129]	Boojjagani *et al.*[27]	Munivel *et al.*[132]	Mustafa *et al.*[9]	Thakur *et al.* [174]	Lara *et al.*[107]
Stolen/lost device attack	✓	✓	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗
Strong replay attack	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Man-in-the-middle attack	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Phishing attack	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Password guessing attack	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Privileged insider attack	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✓
Impersonation attack	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓
Denial of service attack	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓
User anonymity provision	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓
Forward secrecy	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓
Session key security	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Session key recovery attack	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
User traceability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓
Login phase efficiency	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Supports secure Location-Sharing	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Formal security analysis	✗	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Simulation using ProVerif	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓

Note: ✗: does not support a particular feature; ✓: supports a particular feature.

3.3 Location Related Privacy & Security Issues

Mobile online social networking (mOSN) involves interactions between participants with similar interests and objectives through their mobile devices and/or tablet within virtual social networks [85]. MOSN leverages mobile communication networks and social networks, as mobile applications can use existing social networks. In mOSN, social networks can take advantage of mobile features and ubiquitous accessibility. Moreover, an mOSN can readily exploit mobile networks to support the concept of real-time web [49], which is at the forefront of the emerging trends in social networking. MSNs enhance conventional social networks with additional features, such as location-awareness, tag media [90], etc.

In the earlier days of traditional online social networks [108], users were required to share his/her current location information through the process of "check-in", where users manually input the current time and location information to the intended social networking site or mobile apps. Nowadays, the development of GPS technology empowers the mobile user to avoid this manual check-in process and facilitates the mOSN users to share, update and inform current mobile locations.

mOSNs can take advantage of the additional capabilities of modern mobile devices such as smartphones or tablets. People can access mOSNs applications in anywhere and anytime. These capabilities, such as global position system (GPS) receiver, sensing modules (cameras, sensors, etc.), and multiple radios (third/fourth generation cellular, WiFi, Bluetooth, WiFi Direct, etc.), enable mOSNs to enhance conventional social networks with additional features, such as location-awareness [199], location-based service, the ability to capture and tag media [90]. In general built-in GPS is not that much available in laptops. Moreover, it does not exploit 4G or the current standard of cellular networks. Hence, location-based services cannot be accessed using laptops[85].

Detail security analysis reveals the vulnerability of the existing related schemes against many security attacks, such as the denial-of-service (DoS) attack [48], replay attack [48], [105] and privileged insider attack [48], [105]. A recent study reveals that two attacking tricks, namely Regional Statistical Attack (RSA) [170] and Long-term Statistical Attack (LSA) [170], give more opportunity to the attackers.

3.3.1 Literature review on location sharing in mOSN

In the field of mOSNs, privacy and security issues have attracted a large number of research works. Hence, in recent years, many privacy-preserving schemes have been proposed with

their own merits and limitations. Earlier research works on privacy preserving schemes aimed at the providing information privacy [146], user anonymity [150] and protection of location privacy [110].

In order to provide location anonymity, a mobile device encrypts the current location before sending it to servers. K-anonymity for location privacy adopts the process of obfuscating the actual location of the user as proposed and used by [171] and [73]. The use of dummy location along with the real location is the next approach for location anonymity [98]. Location encryption is another very effective way to achieve location privacy protection [97]. The pseudonym methods [140], [149], mix zones [22] and the m-unobservability [39] are some well know schemes developed in the past. Rahman et al. obtained location obscurity through privacy context obfuscation based on various location parameters [148].

Location sharing while maintaining privacy protection in online social networks has been first primarily addressed in 2007 by SmokeScreen [45], which allowed sharing locations between social friends and strangers. Wei et al. enhanced this scheme and proposed Mobishare, where users' social and location information were separately stored into SNS and LBS respectively [187]. Mobishare suffers from the weakness that, in the query phrase, LBS can reveal the topology structure of social networks of a user. Recently, Li et al. [113] enhanced Mobishare to propose new privacy-protected location-sharing scheme in mOSNs, namely MobiShare+, which introduced the concept of dummy queries and private set intersection to prevent LBS from knowing social information of a user. BMobiShare is an improved version over MobiShare+ in terms of transmission efficiency, where the existing private set intersection method is replaced by Bloom Filter [163]. However, the computation cost of BMobiShare is quite high.

In 2015, in order to improve privacy-protection against the insider attack, Li et al. introduced a multiple location server based location sharing system [114]. Although it provides higher security, it is resource-demanding and time-inefficient. As these schemes rely on the third-party location server, they associate the chance of LBS to collude with SNS in order to reveal the social information. Also, they incur a high transmission and storage cost [187], [122], [121], [113], [163]. To address this issue, very recently, Xiao et al. proposed CenLocShare [194], where SNS and LBS were amalgamated into one single server. This scheme reduces communication cost, storage cost and also increases user's privacy protection.

A cryptanalysis of Xi Xiao et al.'s scheme [24] reveals that it is not suitable for practical applications due to some security flaws. The same is true for most of the schemes. Thus, design of a more secure and efficient location sharing scheme remains a challenge.

3.4 Distributed Denial-of-Service (DDoS) Attacks in mOSN

One of the heavily exploited security threats is DDoS attack, which is an attempt by the adversaries to overburden the server or the network through a flood of fake login requests, messages, etc. so that the service providers fail to respond to a valid user for a stipulated time interval.

In a DDoS attack, the attackers send a large number of service request to the service providers that can overload the server and the server deny the request for acceptance [77]. The goal is to overwhelm the network with huge traffic than the network or the server can accommodate. DDoS is a continuous threat faced by businesses of all types, regardless of target market or geographic location. The attacks are gradually becoming more complex in nature that often combines various DDoS methods to execute larger, catastrophic assault. Between January 2020 and March 2021, DDoS attacks increased by 55% and are becoming more complex, with 54% of incidents using multiple attack vectors [185]. Figure 3.2 shows a summary of Cisco's analysis, which indicates a consistent growth in the actual and predicted number of DDoS attacks (in millions) from 2018 to 2023 [12].

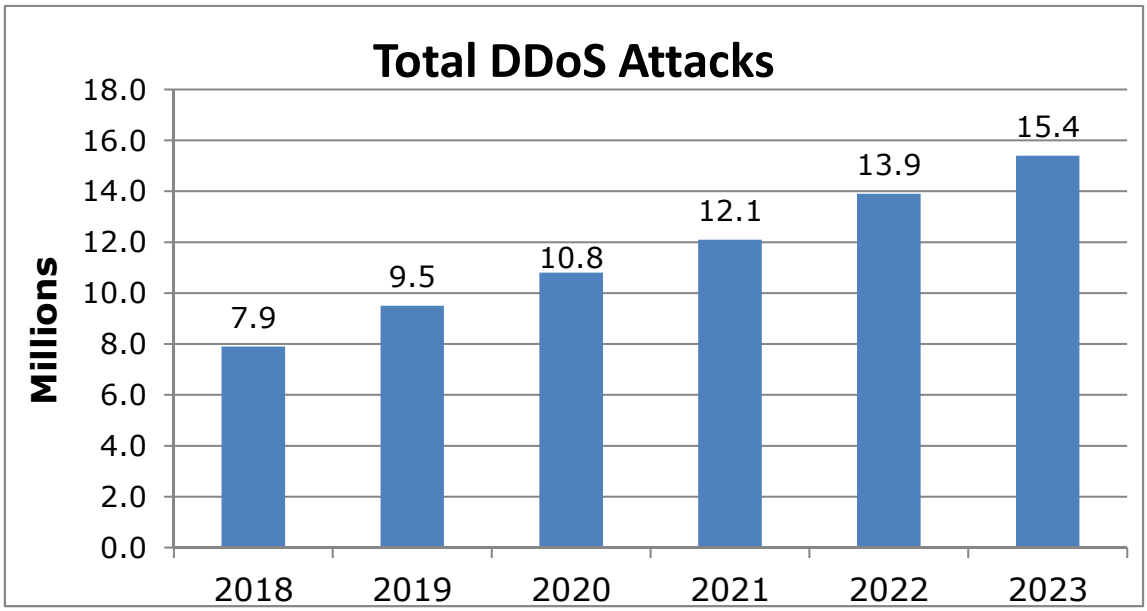

Figure 3.2: Analysis on total DDoS attacks: history and predictions (adapted from [12])

In OSN, DDoS attacks are observed to be both fast-spreading and unpredictable in nature. For launching a DDoS attack, an adversary may capture the login request message and sends

the same message to mOSN server repeatedly.

3.4.1 Literature review on DDoS attacks in mOSN

According to the survey report of NETSCOUT (formally known as Arbor Networks), internet usage has been increased massively due to the Covid-19 pandemic. At the same time, attackers have launched more than 10 million DDoS attacks on online platforms in 2020[87]. It is also to be noted there that the incidents of DDoS attacks have increased 22% in the last six months and 20% over the year 2020 [87]. The most popular of them is DDoS attacks on Amazon Web Services in February 2020. This attack lasted for three days and the company lost a huge revenue. In 2017, Google faced a problem of DDoS attacks that had lasted for six months and service was interrupted heavily due to that attack. In 2018, GitHub faced a DDoS attack, where the attack lasted for 20 minutes and the traffic load was 1.35 terabits per second [12].

In August 2009, attackers launched DDoS attacks on Twitter, YouTube, Facebook, Google's Blogger. There was an interrupted service for several hours on Twitter. Loading Facebook pages took a huge time [126]. It also lists examples of DDoS attacks in OSNs.

Athanasopoulos et al. introduced the concept of "FaceBot" where attackers launch DDoS attacks against the users of the social network [13]. They have experimentally shown how a Social Network like Facebook can be victimized by DDoS attacks. Ur and Ganapathy have shown how a social network user can be responsible for launching of DDoS attacks [181]. Creating a MySpace account, authors posted "hotlinks" of a large media file that was hosted by an ordinary web server. The "hotlinks" were shared by users and at a time a large number of visitors were trying to open the link. As a consequence, a huge number of requests were received by the victim web server resulting DDoS situation. Google Ideas and Arbor Networks together designed a website to visualize the DDoS attacks globally and the visitors also can watch the historical trends of DDoS attacks [70].

Detection of cyber-attacks is difficult. Using IP address spoofing, attackers hide their identities which makes it harder to locate the source of such attacks. Traditional tracing methods require logs from infected machines and network routers. DDoS attacks using Bot-net are more difficult to address. With the flourishing of social networks, bot-net-based DDoS attacks are increasing rapidly. Implementation of botnet based attacks are very easy to design [175], [193]. Due to the topology and social engineering technique of OSN, this type of DDoS attack have become very popular [177]. There are several types of prevention techniques against DDoS attacks.

The secure overlay is one type of DDoS attacks prevention mechanism [76], where an

overlay network needs to be created on the top of the IP network. Outsiders need to cross a distributed firewall to reach the network server. This technique ensures protection from DDoS attacks but it is not suitable for a public server. Another way is using different types of filtering techniques; the network traffic can be filtered and DDoS attacks can be prevented [95], [94]. But the failure rate and computational overhead of this technique are also high.

Another interesting prevention mechanism is the concept of Honeypots [79], [102]. Less secure mimics of the original networks called Honeypots are created to confuse the attackers. Attackers attack Honeypots, thinking that they are attacking the original network. This mechanism also suffers from major drawbacks. Honeypots neither completely disclose the identity of the attackers nor can detect the attack and it also forwards the attack packets to their destinations.

Balancing traffic load is one of the approaches to overcome DDoS attacks [20]. When a server gets overloaded, the traffic load is distributed among several servers. Hence, if a server becomes a victim of a DDoS attack, the load balancer guarantees flexibility by distributing the load to other servers. The main drawback of this technique is several numbers of data stores and replicated servers will be required to implement the system. Recently a technique is highlighted that is awareness creation about the attacks. Mobile or IoT devices have a less secure operating system. Only users' awareness can secure the system from DDoS attacks.

DDoS attacks have become a severe threat for Social network applications. At the same time, user privacy or protection of users' sensitive information on the social networking platform is also very crucial. For avoiding such problems, it is very essential to verify and validate the identity of a user before allowing him/her to access the social network service. An efficient authentication scheme can resist several issues very well. There are a huge number of researches have been done to resist DDoS attacks in Mobile Device, Internet, [68], etc. or Mobile Cloud Computing [130], etc. Yardi et al [198] first proposed an authentication system called Lineup for Facebook based on the photo where they discussed the Denial of Service (DoS). Earlier, a number of authentication schemes for social network application were was proposed. [99], [145], [144]. Recently Darwish et al. [47] proposed an authentication scheme that can resist Denial of service attacks for Cloud-based services.

Overall, many efforts have been made in designing authentication protocols for mitigating DDoS attacks. But most of these protocols are vulnerable to DDoS attacks, which occur due to the loss of synchronization between the participants. Further, to rebuild synchronization between the participants, a protocol may need to compromise un-linkability property. Therefore, designing of an efficient authentication protocol for resisting DDoS attack is still a

challenging research problem.

3.5 Phishing Attacks in mOSN

There is an immense change in our personal life with the evolution of mobile Online Social Networks (mOSNs). People can join OSN such as Facebook, Twitter, LinkedIn, etc, and share their common interests. But, in reality, mOSN has huge security concerns and it is more insecure compared to the situation when OSN is accessed through browsers from traditional computers like desktops, laptops [106]. Posting any sensitive information from a mobile device is more insecure.

The relationship between users in OSN is based on trust and using this nature, attackers can easily spread phishing attacks, spam, and malware over OSN. There are many ways of attacks like creating fake OSN profiles, using stolen credentials of OSN accounts or deploying programmatic community bots, etc. which attackers actually promote [151], [125], [40], [8].

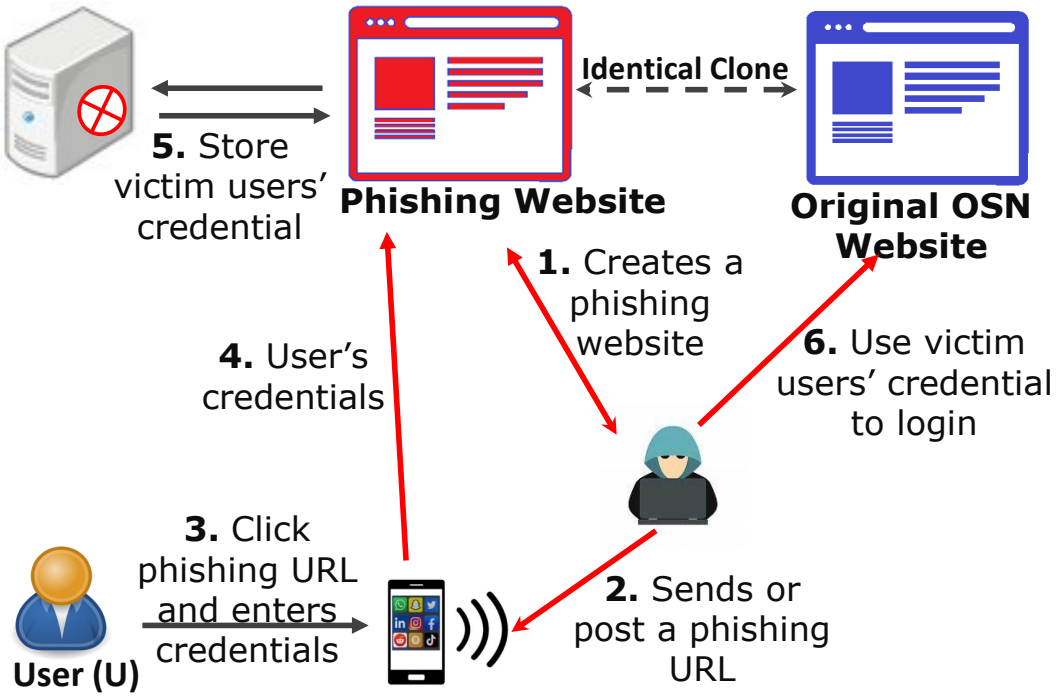

Figure 3.3: Summary of phishing attack in OSN networks (adapted from [9])

A very popular classical technique of stealing information on the web is by phishing attack which is still in practice. First, attackers develop a website that looks almost identical compare

to the original website as depicted in Figure 3.3. Next, this fake link will be posted on the OSN user's wall [180]. When an OSN user open that URL and enters its credentials, attackers will collect his/her personal or sensitive information, banking details, credit card information, etc. [162], [38]. Finally, the attacker uses those victim users' credentials to login to the original website.

The first popular phishing attack occurred in 1990 on America Online (AOL) Network Systems [176]. To create a new account and access the AOL resources, a user needs to provide his or her credit card information. Attackers had created a fake AOL website and using that had stolen the credit card information. Since then, phishers have targeted online banking services, online trading, and the most popular OSN websites.

Many studies have been published on how phishing attacks are practiced in OSN. Aggarwal et. al. [4] had shown how phishing attacks were carried out on Twitter using their PhishAri method. In that study, they also pointed out how attackers published phishing links on Facebook, Twitter, and YouTube using the growing popularity of OSN. In 2010, almost 43% of attacks were phishing attack. In 2012, Facebook users were the main victims and out of all attacks, $1/5^{th}$ were on Facebook. The trend of phishing attacks has increased day by day with the popularity of OSNs. According to some other studies, the number of phishing attacks doubled in 2012 compared to 2011 [4]. In March 2017, ten thousand employees of the US Department of Defense were targeted with some "adapted message" on Twitter [32].

According to the published report "State of the Phish Report" in Wombat, the rate of attacks has increased in 2018 compared to 2017 [154]. As per the "Internet Crime Report 2020" of the Federal Bureau of Investigation [2], phishing is the most popular crime type on the internet. It is increased by more than two times from 2019 to 2020 and the detailed comparison of the top five internet crimes is presented in Figure 3.4.

3.5.1 Literature review on phishing attack in mOSN applications

Phishing is one of the traditional security threats in OSN. This attack is usually used to hack users' private information, such as passwords, credit, or debit card details, etc. [151]. The immense development of portability and ease of smartphones and cost-effective internet connectivity make mobile phones an essential device for surfing OSN. This huge acceptance of mobile devices for online deeds makes mobile OSN, a popular attack surface for cyber phishers [4].

In mobile based authentication, security is one of the major challenges. Some current authentication approaches have used a mobile device to verify the identity of the user [119].

Various other attributes, such as mobile numbers, digital signatures, passwords, hash value, biometrics, or group keys also have been used to verify the authentication of the user [133]. But this type of authentication is not suitable for mobile OSNs. Suppose, one user has the same user identity and password for several OSNs, such as Facebook, Twitter, YouTube, etc. The credentials of all other accounts of other sites will be at risk if the confidentiality of one of these sites gets compromised. An adversary or the phisher can capture the credentials of the users if they send directly through the network. So some approaches transmit or share the value of the attributes either after hashing or encrypting [6], [141].

Any flawless authentication scheme can prevent several security attacks and make a system more protected. The phishing attack is a major security concern in the mobile environment/device [83]. Any such attack in a mobile environment compromises the user's personal or sensitive information [86]. There is an additional type of authentication scheme [195], [156], [15] using XOR operations, hash functions, and random nonce which requires low computation overhead, and those schemes are termed as lightweight.

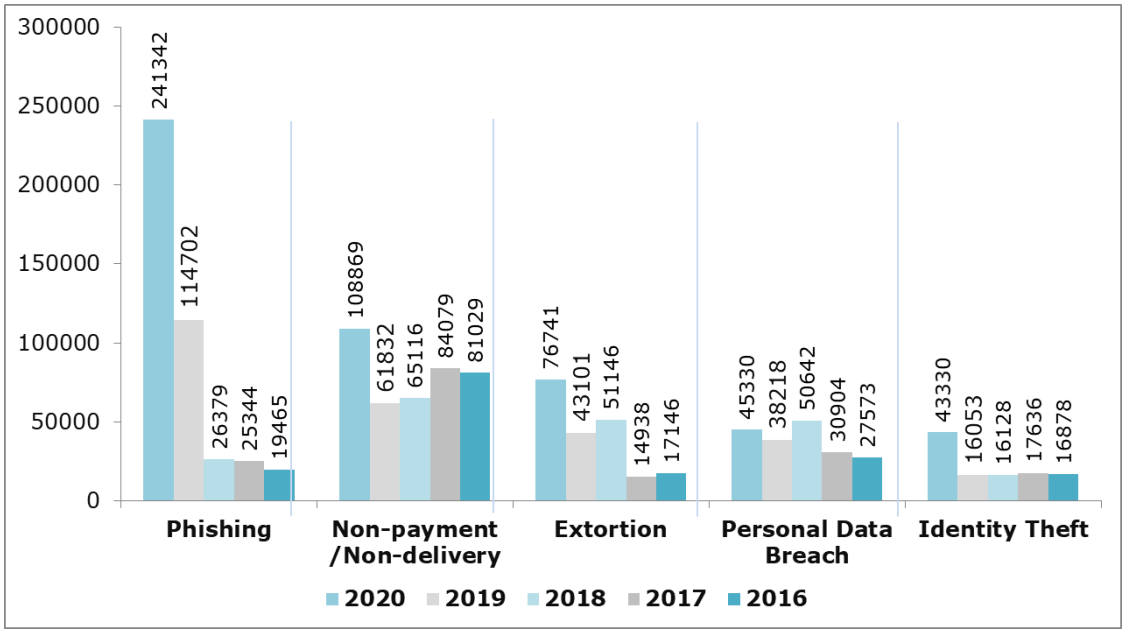

Figure 3.4: Comparison of top 5 crime type including phishing attack in last five years (adapted from [2])

Munivel *et al.* in their work [132] have provided an authentic mobile cloud environment that may be more secure against phishing attacks. Bojjagani *et al.* have proposed an authentication protocol in [27] for preventing phishing attacks in the mobile banking environment.

But these schemes are not designed to resist phishing attacks in mOSN. An anti-phishing approach based on a heuristic technique was proposed by Okunoye *et al.* [139], where the blacklist would be updated with a fishy website immediately after its detection. Similarly, the white list will be updated with the identification of a genuine website. But this scheme fails to provide user confidentiality, security against replay, insider attacks, cyber-attacks, etc. and the required time complexity is more.

In [14], Awan provided several types of phishing attacks and defense mechanisms against those attacks. Alabdan [7] emphasized a comprehensive analysis of several methods used for traditional and recent phishing attacks and their technical features. Chaudhry *et al.* [37] discussed different techniques used for phishing attacks and provided prevention mechanisms of the server and the client. Alzuwaini and Yassin [9] proposed a digital-signature-based verification scheme for preventing phishing attacks. But several rounds of message communication are required to authenticate a user, which is a very time-consuming procedure. Thakur and Yoshiura [174] introduced an anti-phishing method in the mobile banking system. As all original user credentials were communicated through the mobile app, the scheme suffers from man-in-middle attacks, replay attacks and the mobile app itself is a privacy snatcher of a mobile user.

3.6 Summary

In this chapter, I have discussed an outline of state-of-art of related works in the areas of secure location sharing, detection, and prevention techniques of distributed denial of service (DDoS) attacks and phishing attacks in online social network applications. From the literature survey, it can be concluded that location sharing in OSN suffers from security and privacy issues. Similarly, (DDoS) attacks and phishing attacks become severe threats for OSN users. It is also observed that most of the existing security protocols did not fulfill the requirements. The merits and demerits of existing protocols motivate me to design more secure, efficient, and lightweight authentication schemes for OSN applications.

Chapter 4

Privacy-Preserving Efficient Location-Sharing Scheme for mOSN

As discussed in the literature, the existing location sharing schemes of mOSNs suffer from several security drawbacks. In general, three major security challenges are primarily faced by existing location sharing schemes designed for mOSN applications. First, all location-based services must be privacy-preserving. An attacker or malicious user must not be able to access and/or modify the personal information of a user. Second, to ensure user location privacy, a location-based social network server should store various fake or dummy identities of a user. Finally, a physical distance threshold between a user and his/her friend or stranger must be registered. A location query about a user's friends or strangers is processed only if their current physical distance is within that predefined distance threshold.

In this chapter, I have aimed to address the above security drawbacks of the existing location-based features provided by popular OSNs. According to our proposed scheme, a user and a location-based social network server at the beginning, separately establish a shared symmetric session key with a cellular tower. All location updates and friend's location query messages will be encrypted with the session key before transmission. Because of this end-to-end encryption, an adversary will have little chance to reveal the location information of users. Furthermore, unlike the location-based services of existing OSNs, our proposed scheme allows a user to decide a distance threshold, up to which he/she wants to make himself/herself visible to its social friends. This imposes a much better user-controlled restriction on location sharing, as unrestricted location sharing can lead to security vulnerabilities.

4.1 Research Contributions

The following contributions are made in this chapter:

- The location sharing scheme of the proposed scheme does not depend on any third-party location-based server. This eliminates the possibility of LBS to reveal the social network topology structure of a social user.
- The proposed scheme integrates LBS and SNS into a set of single entity servers, thereby reducing their internal communication overhead.
- The proposed scheme has the ability to resist various active and passive security attacks which are present in the existing schemes.
- The location sharing mechanism is efficient, lightweight and secure. I have avoided computation costly operations like bilinear pairing, elliptic curve cryptography, public key infrastructure (PKI), public key cryptography.
- On top of informal security analysis, I have validated security of the proposed scheme through formal security verification using random oracle, and through security simulation using ProVerif 1.93.

4.2 The Threat Model and System Model

This section briefly describes the basic attack model or adversary model applicable for our proposed scheme. Moreover, I depict the outline of the system model adopted for our proposed location sharing scheme for the online social network.

4.2.1 The threat model

I primarily assume that cellular tower (CT) is a trusted body and define the threat model concerning the location-sharing social network servers ($LSSNSs$) and the user (U). I define the model below:

- Registered entities like U, $LSSNS$ and CT communicate through a public insecure wireless channel. The proposed scheme adopts the widely-accepted Dolev-Yao threat model (DY model) [55]. An attacker or a malicious user has all the capabilities of executing all potential attacks defined in the classical DY model.

- A registered or authorized user or a privileged insider of the system may turn into a malicious user, who illegally intends to access various location or social information of other genuine users.
- $LSSNS$s exhibit an 'honest but curious' nature. They alone, or after colluding with other servers, try to retrieve the social network topology or location information of other registered users.
- Our proposed scheme assumes CT to be a trusted entity.

4.2.2 The system model

Figure 4.1 shows the basic system model of the proposed scheme. Here, I have defined the basic entities, which are described as follows:

- **Mobile User** (MU): Sends and responds to three types of request queries. These include sharing of location information to other social friends and strangers, updating own location information and querying a friend's location information.
- **Location Sharing Social Network Servers** ($LSSNS$): Responsible for storing, updating and informing various location information of U.
- **Cellular Tower** (CT): It is a trusted entity, which receives, processes and forwards various messages of U and $LSSNS$. All messages communicated between U and $LSSNS$ are communicated via CT.

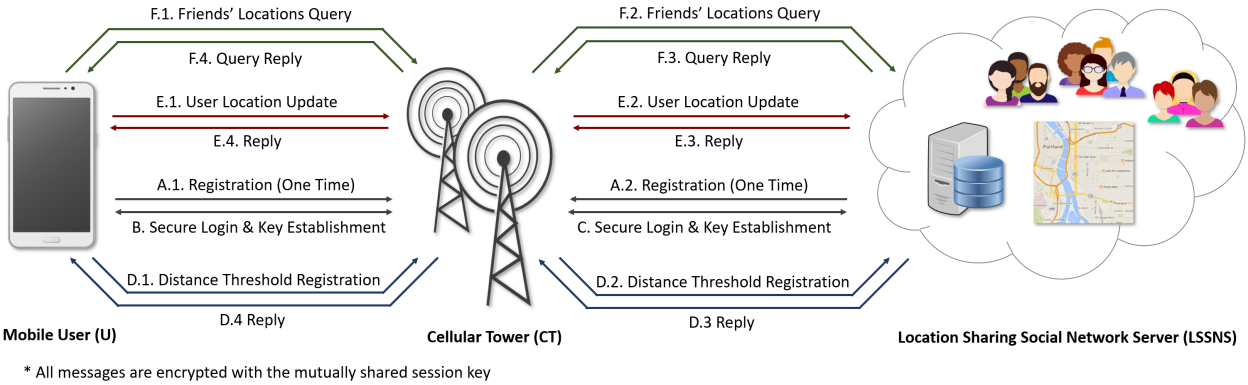

Figure 4.1: The architecture for location sharing in the mOSN through multiserver system

The overall flow of the model is shown in Figure 4.1. First, the mobile user MU and the Location Sharing Social Network Server $LSSNS$ register to a cellular tower CT (process A). This is a one-time operation and is executed through a secure channel. Next, the mobile user MU and $LSSNS$ make a secure login to the registered CT and establish a shared session key (processes B and C respectively). Thereafter, MU registers a distance threshold to $LSSNS$ via CT, in which corresponding social friends can be searched (process D). When required, the mobile user updates his/her current location to $LSSNS$ through the cellular tower (process E). Finally, the mobile user obtains his/her social friends' identity and location, for those who are willing to share their information from $LSSNS$ through the cellular tower (process F).

Table 4.1: Symbols and notations used in the proposed scheme

Symbol	Description	Symbol	Description
MU_i	i^{th} mOSN user	ID_i	The identity of the i^{th} user
CT	Cellular Tower	ID_{CT}	The identity of the CT
$LSSNS_j$	j^{th} Location Sharing Social Network Server	ID_{S_j}	The identity of the $LSSNS_j$
PW_i	The login password of MU_i	PW_{S_j}	The login password of $LSSNS_j$
$\mathcal{B}_i$	User biometric of MU_i	$MPWB_i$	Biometric embedded password of MU_i
TID_i	Temporary identity of MU_i	RID_i	The pseudo-identity of MU_i
RID_Γ	The pseudo-identity of CT	Ω_{u_i}	128-bits Random Variable
RN_{u_i}	128-bit random number chosen by MU_i	RN_{ct}	128-bit random number chosen by CT
X_Γ	1024-bit master secret key chosen by CT	$x_{CT_{U_i}}$	1024-bit secret key chosen by CT
$A_{U_i CT}$	Temporary Variable	$V_{CT_{U_i}}$	Temporary Variable
P_i^1	Temporary Variable used by MU_i	P_i^2	Temporary Variable used by MU_i
RPW_{S_j}	Masked password of $LSSNS_j$	C_j	Temporary masked password of $LSSNS_j$
E_1, E_2	Temporary Variables used by $LSSNS_j$	f_{S_j}	Temporary Variable used by $LSSNS_j$
PID_{S_j}	pseudo-identity of $LSSNS_j$	SN_j	Serial number of $LSSNS_j$
(x_{u_i}, y_{u_i})	The real location coordinate of MU_i	$Index_{u_i}$	The encrypted real index of user's location
SK	Symmetric key	$\mathcal{K}_{MU_i,\mathcal{F}}$	SK shared between MU_i and its friends $\mathcal{F}$
$\mathcal{SK}_{MU_i,CT}$	SK shared between MU_i and CT	$\mathcal{SK}_{CT,S_j}$	SK shared between CT and $LSSNS_j$
Θ	The set of social friends of MU_i	$\delta(\cdot)$	The euclidean distance function among users
$\mathcal{D}_{f_{u_i}}$	The registered distance threshold of MU_i	ds_{U_i}	Distance threshold for strangers' location query
$\mathcal{T}_{u_i}$	Timestamp generated by MU_i	$\mathcal{T}_{ct}$	Timestamp generated by CT
$H(\cdot)$	One way cryptographic hash function	$E_k(\cdot)/D_k(\cdot)$	Symmetric encryption/decryption using key k
$\|\|, \oplus$	Concatenation, bitwise XOR operations	$A \rightarrow B : \langle M \rangle$	A sends message M to B via public channel
$\triangle T$	Maximum transmission delay	qf_{u_i}	Friend location query distance limit

4.3 The Proposed Scheme

In order to design the proposed scheme, various symbols are used. The symbols and notations are tabulated in Table 4.1.

4.3.1 The registration phase

This phase involves two distinct registration processes, namely, (a) the registration of a mOSN user (MU_i) to a cellular tower, and (b) the registration of a $LSSNS_j$ to a cellular tower. The registration process is a one-time operation that is executed through a secure channel, the message communications of this phase is shown in Figure 4.2.

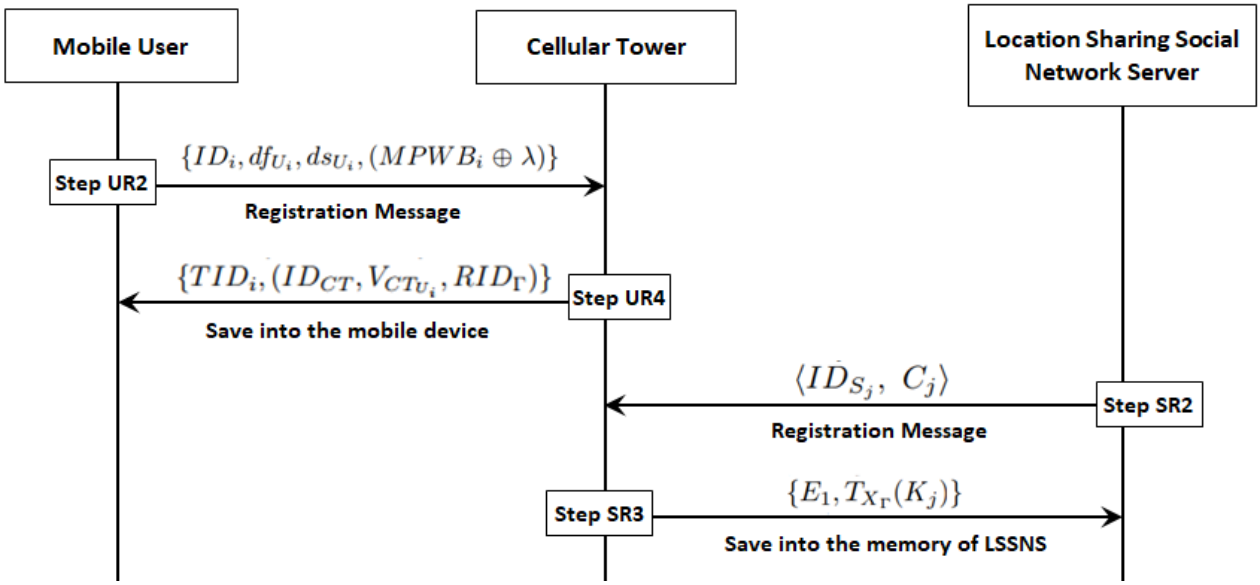

Figure 4.2: The message communication in registration phase

Mobile user registration phase:

In this phase, a series of steps are executed for the registration of a mobile user MU_i to the CT. These steps are as follows.

Step UR1:

1. MU_i selects own identity, password, and biometrics as ID_i, PW_i, $\mathcal{B}_i$ respectively.

2. MU_i selects parameters n and λ, which are two 128-bit random numbers.

Step UR2:

1. MU_i uses the fuzzy extractor $(\cdot)$ function to produce $(\eta_i, \mu_i) = Generation(\mathcal{B}_i)$ and computes the biometric embedded password $MPWB_i = H(ID_i||\ H(PW_i||\ \eta_i\ ||n))$.

2. Through a secure channel, MU_i delivers its registration message $\{ID_i, df_{U_i}, ds_{U_i}, (MPWB_i \oplus \lambda)\}$ to the CT.

   Note that ID_i and PW_i is randomized with the concatenation of 128-bit (16-byte) random numbers[36], [10], [156]. I mask the user id and password as $MPWB_i = H(ID_i|| H(PW_i|| \eta_i ||n))$. Thus, guessing of ID_i and PW_i from $MPWB_i$ is infeasible, as it computationally hard to guess three secrets simultaneously. An 128-bit random number can generate 10^{38} possible values (as $2^{128} \approx 10^{38}$). So the guessing possibility is only $\approx \frac{1}{10^{38}}$[66], [10].

Step UR3:

1. CT randomly selects its own 1024-bit master secret key X_Γ.
2. For each $\langle CT \leftrightarrow MU_i \rangle$ pair, CT randomly selects a 1024-bit secret key $x_{CT_{U_i}}$.
3. CT computes $A_{U_{iCT}} = H(H(ID_i \oplus x_{CT_{U_i}}) ||X_\Gamma)$, $V_{CT_{U_i}} = A_{U_{iCT}} \oplus MPWB_i$.
4. CT chooses its pseudo-identity as $RID_\Gamma = H(ID_{CT}||X_\Gamma)$.

Step UR4:

1. CT provides an anonymous temporary identity for each mOSN user MU_i. This is done by selecting a random but temporary identity TID_i for each user MU_i.
2. CT saves m $\langle CT \leftrightarrow MU_i \rangle$ key-plus-id combinations $\{TID_i, (ID_{CT}, V_{CT_{U_i}}, RID_\Gamma) \mid 1 \leq j \leq m\}$ in mobile device of MU_i.

   Note that for all MU_is, the CT saves the record $\{ID_i, TID_i, x_{CT_{U_i}}\}$ in own database.

Step UR5:

1. MU_i computes $P_i^1 = H(PW_i|| \eta_i) \oplus n$ and $P_i^2 = H(ID_i ||PW_i ||\eta_i ||n)$.
2. MU_i modifies $V_{CT_{U_i}}$ as $V'_{CT_{U_i}} = V_{CT_{U_i}} \oplus \lambda$, $RID_i = TID_i \oplus H(ID_i ||V'_{CT_{U_i}})$ and $RID'_\Gamma = RID_\Gamma \oplus H(\eta_i||n)$ for all $1 \leq j \leq m$.
3. MU_i stores parameters $\langle \mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}\text{s}, RID_i\text{s}$ and $RID'_\Gamma \rangle$ and removes $V_{CT_{U_i}}\text{s}$, RID_Γ and TID_is from own mobile device.

The location sharing social network server registration phase:

Each location sharing social network server $LSSNS_j$ registers to the cellular tower CT through the following steps:

Step SR1:

1. $LSSNS_j$ chooses own id and password as ID_{S_j} and PW_{S_j}.
2. It selects one random number b of 128-bit long.

Step SR2:

1. $LSSNS_j$ computes masked password $RPW_{S_j} = H(ID_{S_j} \,||PW_{S_j})$ and $C_j = H(ID_{S_j} \,||PW_{S_j} \,||b)$.
2. $LSSNS_j$ submits $\langle ID_{S_j}, C_j \rangle$ to CT via a secure channel.

Step SR3:

1. CT uses its master secret key X_Γ and one random number r (128-bit) to compute $K_j = H(H(ID_{S_j}||X_\Gamma) \oplus r)$, and $E_1 = K_j \oplus C_j = K_j \oplus H(ID_{S_j} \,||PW_{S_j} \,||b)$.
2. CT embeds the parameters $\{E_1, T_{X_\Gamma}(K_j)\}$ in memory of each $LSSNS_j$.
3. CT saves pair $\langle ID_{S_j}, SN_j, r \rangle$ into its database, where SN_j is the identity or serial number of the server $LSSNS_j$.

Step SR4:

1. $LSSNS_j$ computes $E_2 = RPW_{S_j} \oplus b$ and $f_{S_j} = H(RPW_{S_j}||b)$.
2. $LSSNS_j$ stores E_2, f_{S_j}, $H(\cdot)$ into its own memory.

The Summary of registration process of MU_i and $LSSNS_j$ to CT is shown in Figure 4.3.

4.3.2 The mOSN user login, authentication and key establishment phase

The mOSN user MU_i makes a secure login to the registered CT and establishes a shared session key through the following steps:

mOSN user (MU_i)	Cellular tower (CT)
Input ID_i, PW_i and $\mathcal{B}_i$. Select 128-bit numbers n and λ. Compute $(\eta_i, \mu_i) = Generation(\mathcal{B}_i)$, $MPWB_i = H(ID_i \|\| H(PW_i \|\| \eta_i \|\| n))$. $\xrightarrow[\textbf{Secure channel}]{\{ID_i, (MPWB_i \oplus \lambda)\}}$	Select own master secret key X_Γ. Select $x_{CT_{U_i}}$ for each $\{CT \leftrightarrow MU_i\}$ pair. Compute $A_{U_i CT} = H(H(ID_i \oplus x_{CT_{U_i}}) \|\| X_\Gamma)$, $V_{CT_{U_i}} = A_{U_i CT} \oplus MPWB_i$, $RID_\Gamma = H(ID_{CT} \|\| X_\Gamma)$. Select temporary identity TID_i for each MU_i. For all $\{CT \leftrightarrow MU_i\}$ pairs, store $\{TID_i, (ID_{CT}, V_{CT_{U_i}}, RID_\Gamma)\}$ in memory of MU_i. Store record $\{ID_i, TID_i, x_{CT_{U_i}}\}$ in CT. $\xleftarrow[\textbf{Secure channel}]{\{Mobile\ Device\}}$
Compute $P_i^1 = H(PW_i \|\| \eta_i) \oplus n$, $P_i^2 = H(ID_i \|\| PW_i \|\| \eta_i \|\| n)$, $V'_{CT_{U_i}} = V_{CT_{U_i}} \oplus \lambda$ Compute $RID_i = TID_i \oplus H(ID_i \|\| V'_{CT_{U_i}})$, $RID'_\Gamma = RID_\Gamma \oplus H(\eta_i \|\| n)$. Store $\{\mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}\text{s}, RID_i\text{s}, RID'_\Gamma\}$ into own device memory. Delete $V_{CT_{U_i}}\text{s}$, RID_Γ and $TID_i\text{s}$ from MU_i mobile device memory.	

Location Sharing Social Network Server ($LSSNS_j$)	Cellular tower (CT)
Selects id, password as ID_{S_j} and PW_{S_j} and 128-bit number b randomly. $RPW_{S_j} = H(ID_{S_j} \|\| PW_{S_j})$, $C_j = H(ID_{S_j} \|\| PW_{S_j} \|\| b)$. $\xrightarrow[\textbf{secure channel}]{\langle ID_{S_j}, C_j \rangle}$	
	Generate 128-bit number r randomly. Using the master secret key X_Γ, Computes $K_j = H(H(ID_{S_j} \|\| X_\Gamma) \oplus r)$, $T_{X_\Gamma}(K_j)$, $E_1 = K_j \oplus C_j$. $\xleftarrow[\textbf{secure channel}]{\textbf{server memory}\{E_1, T_{X_\Gamma}(K_j)\}}$
Computes $E_2 = RPW_{S_j} \oplus b$, $f_{S_j} = H(RPW_{S_j} \|\| b)$. Store E_2, f_{S_j}, $H(\cdot)$, into server memory.	Stores $\langle ID_{S_j}, SN_j, r \rangle$ into own database.

Figure 4.3: The registration phases of MU_i and $LSSNS_j$ in the proposed scheme

Step ULA1:

1. MU_i inputs own identity, password, and bio-metrics (noisy) as ID_i, PW_i, and $\mathcal{B}'_i$ re-

spectively.

2. Using stored μ_i and P_i^1, MU_i computes $\eta_i = Reproduction(\mathcal{B}_i', \mu_i)$ and generates $n' = P_i^1 \oplus H(PW_i \,||\eta_i)$.

3. MU_i calculates $H(ID_i \,||PW_i \,||n' \,||\eta_i)$ and compares with stored P_i^2.

4. If the verification succeeds, go to *Step ULA2*, else, *exit*.

Step ULA2:

1. MU_i randomly generates Ω_{u_i} (128-bit number).

2. Using stored paratemer $V'_{CT_{U_i}}$, MU_i computes :

   (a) $MPWB_i = H(ID_i|| \; H(PW_i|| \; n' \; ||\eta_i))$.

   (b) $A_{U_{i}CT} = V'_{CT_{U_i}} \oplus \; MPWB_i$.

   (c) $\mathcal{M}_1 = A_{U_{i}CT} \oplus \; \Omega_{u_i} \oplus \; \mathcal{T}_{u_i} \oplus \; H(ID_{CT})$.

   (d) $TID_i = RID_i \oplus H(ID_i|| \; V'_{CT_{U_i}})$.

   (e) $TID_i^* = TID_i \oplus \; H(ID_{CT}|| \; \mathcal{T}_{u_i})$

3. MU_i uses the current login timestamp $\mathcal{T}_{u_i}$ and computes a hash value $H_1 = H(ID_i|| \; \mathcal{M}_1|| \; \Omega_{u_i}|| \; \mathcal{T}_{u_i})$.

4. Through a public channel, MU_i sends $\{TID_i^*, \; \mathcal{M}_1, \; H_1, \; \mathcal{T}_{u_i}\}$ to CT.

Step ULA3:

1. CT verifies if $|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}| \stackrel{?}{\leq} \triangle T$. If verification holds go to step 2, else *exit*.

2. CT calculates $TID_i = TID_i^* \oplus \; H(ID_{CT} \; ||\mathcal{T}_{u_i})$.

3. Corresponding to calculated TID_i, CT finds the record $\langle\{ID_i, x_{CT_{U_i}}\}\rangle$ from own database.

4. CT computes $B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})||X_\Gamma)$.

5. CT computes $\mathcal{P}_1 = \mathcal{M}_1 \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}) \oplus B_{CT_{U_i}} = \Omega_{u_i}$.

   Note that CT obtaines $\mathcal{P}_1$ of step (5), as $A_{U_{i}CT} = B_{CT_{U_i}} = H(H(ID_i \oplus \; x_{CT_{U_i}})|| \; X_\Gamma)$.

Step ULA4:

1. CT uses received parameters to prepares a hash value $H_2 = H(ID_i \,||\mathcal{M}_1 \,||\mathcal{P}_1 \,||\mathcal{T}_{u_i})$.
2. CT verifies if $H_2 \stackrel{?}{=} H_1$. If verification holds go to step 3, else *exit*.
3. CT saves the record $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i} \rangle$ in its database.
4. CT generates a 128-bit random number Ω_{ct}.
5. CT computes $\mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$. Here $\mathcal{T}_{ct}$ is the current timestamp of CT.
6. CT computes shared session key $SK_{CT,MU_i} = H(ID_i \,||ID_{CT} \,||B_{CT_{U_i}} \,||\mathcal{P}_1 \,||\Omega_{ct} \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct})$.
7. CT prepares hash value $H_3 = H(ID_i \,||\mathcal{P}_1 \,||\Omega_{ct} \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct} \,||SK_{CT,MU_i})$.
8. Through public channel, CT sends authentication response message $\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i.

Step ULA5:

1. MU_i receives an authentication response message from step 8 of $ULA4$.
2. MU_i verifies the transmission delay by comparing received and current timestamps. Go to step 3, if verification holds, else *exit*.
3. MU_i computes $\mathcal{P}_2 = \mathcal{M}_2 \oplus \mathcal{T}_{ct} \oplus ID_i \oplus A_{U_{iCT}} = \Omega_{ct}$.

   Note that I obtain $\mathcal{P}_2$ of step (3), as $A_{U_{iCT}} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \,||X_\Gamma)$.

Step ULA6

1. MU_i generates a session key (mutually shared with CT) as $SK_{MU_i,CT} = H(ID_i \,||ID_{CT} \,||A_{U_{iCT}} \,||\Omega_{u_i} \,||\mathcal{P}_2 \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct})$.
2. MU_i computes final hash value $H_4 = H(ID_i \,||\Omega_{u_i} \,||\mathcal{P}_2 \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct} \,||SK_{MU_i,CT})$.
3. If $H_4 \stackrel{?}{=} H_3$, then MU_i confirms that the session key $SK_{MU_i,CT}$ $(= SK_{CT,MU_i})$ is mutually verified and established. Else, MU_i discards the session key and terminates the process.

For all message communicaions in the current session, MU_i and CT use this key for message encryption. The Summary of mOSN user login, authentication and key establishment phase with CT is shown in Figure 4.5 and the message communications of Login, Authentication and Key Establishment Phase is shown in Figure 4.4.

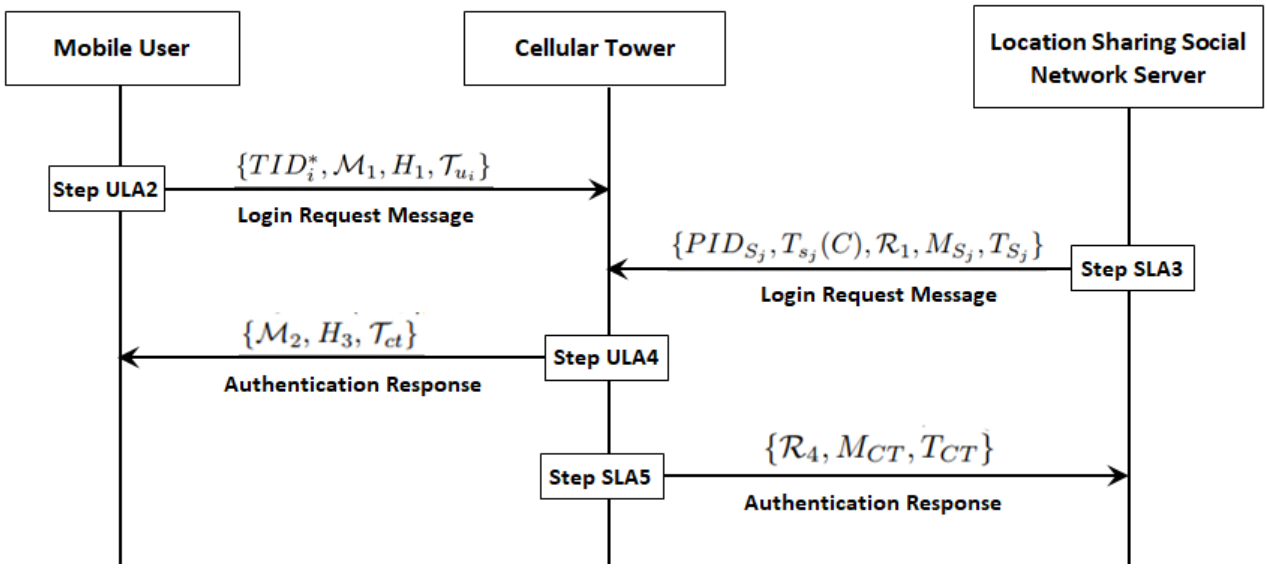

Figure 4.4: The message communication in login, authentication and key establishment phase

4.3.3 The $LSSNS_j$ login, authentication and key establishment phase

The $LSSNS_j$ makes a secure login to the registered cellular tower CT and establishes a shared session key through the following steps:

Step SLA1:

1. $LSSNS_j$ inputs own id ID_{S_j} and password PW_{S_j}.
2. $LSSNS_j$ generates $RPW_1 = H(ID_{S_j}||PW_{S_j})$ and $b_1 = E_2 \oplus PWB_1$.
3. $LSSNS_j$ uses generated b_1 and computes $f'_{S_j} = H(RPW_1 \,||b_1)$.
4. $LSSNS_j$ verifies if $f_{S_j} \stackrel{?}{=} f'_{S_j}$. If verification holds, go to step *SLA2*, else *exit*.

Step SLA2:

1. $LSSNS_j$ computes $C = E_1 \oplus H(ID_{S_j}||\, PW_{S_j} \,||b_1)$.
2. $LSSNS_j$ generates random number s_j.
3. $LSSNS_j$ computes $T_{s_j}(C)$.
4. $LSSNS_j$ computes $K_1 = T_{s_j}(T_{X_\Gamma}(C))$.

mOSN user (MU_i)	Cellular Tower (CT)
Login phase	
Input ID_i, PW_i, and $\mathcal{B}_i'$.	
Compute $\eta_i = Reproduction(\mathcal{B}_i', \mu_i)$, $n' = P_i^1 \oplus H(PW_i \,||\eta_i)$.	
Verifies if stored $P_i^2 = H(ID_i \,||PW_i \,||\text{n'} \,||\eta_i)$?	
If verification holds, generate 128-bit number Ω_{u_i}	
Compute $MPWB_i = H(ID_i\|\| \, H(PW_i\|\| \, n' \,||\eta_i))$	
$A_{U_iCT} = V'_{CT_{U_i}} \oplus MPWB_i$	
$\mathcal{M}_1 = A_{U_iCT} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$	
$TID_i = RID_i \oplus H(ID_i\|\| \, V'_{CT_{U_i}})$	
$TID_i^* = TID_i \oplus H(ID_{CT}\|\| \, \mathcal{T}_{u_i})$	
$H_1 = H(ID_i\|\| \, \mathcal{M}_1\|\| \, \Omega_{u_i}\|\| \, \mathcal{T}_{u_i})$	
$\xrightarrow{\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}}$ **(public channel)**	
	Authentication phase
	Verify if $\|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}\| \leq \triangle T$?
	Compute $TID_i = TID_i^* \oplus H(ID_{CT} \,||\mathcal{T}_{u_i})$,
	Corresponding to TID_i, find the record $\langle\{ID_i, x_{CT_{U_i}}\}\rangle$ from own database.
	Compute $B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})\|\|X_\Gamma)$.
	$\mathcal{P}_1 = \mathcal{M}_1 \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}) \oplus B_{CT_{U_i}} = \Omega_{u_i}$,
	as $A_{U_iCT} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})\|\|X_\Gamma)$,
	$H_2 = H(ID_i \,||\mathcal{M}_1 \,||\mathcal{P}_1 \,||\mathcal{T}_{u_i})$, and verify if $H_2 = H_1$?
	If verification holds, accepts the user login request.
	Save the record $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i}\rangle$ in its database.
	Generate 128-bit random number Ω_{ct}
	Compute $\mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$,
	Compute session key as
	$SK_{CT,MU_i} = H(ID_i \,||ID_{CT} \,||B_{CT_{U_i}} \,||\mathcal{P}_1 \,||\Omega_{ct} \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct})$,
	Compute hash value $H_3 = H(ID_i \,||\mathcal{P}_1 \,||\Omega_{ct} \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct} \,||SK_{CT,MU_i})$.
	$\xleftarrow{\{\mathcal{M}_2,\, H_3,\, \mathcal{T}_{ct}\}}$ **(public channel)**
Verify if $\|\mathcal{T}_{ct}^* - \mathcal{T}_{ct}\| \leq \triangle T$?	
Compute $\mathcal{P}_2 = \mathcal{M}_2 \oplus \mathcal{T}_{ct} \oplus ID_i \oplus A_{U_iCT} = \Omega_{ct}$,	
as $A_{U_iCT} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \,||X_\Gamma)$.	
Compute session key as	
$SK_{MU_i,CT} = H(ID_i \,||ID_{CT} \,||A_{U_iCT} \,||\Omega_{u_i} \,||\mathcal{P}_2 \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct})$	
Compute hash value	
$H_4 = H(ID_i \,||\Omega_{u_i} \,||\mathcal{P}_2 \,||\mathcal{T}_{u_i} \,||\mathcal{T}_{ct} \,||SK_{MU_i,CT})$.	
Verify if $H_4 = H_3$? if verification holds,	
store session key $SK_{MU_i,CT}$ $(= SK_{CT,MU_i})$**.**	**Store session key** SK_{CT,MU_i} $(= SK_{MU_i,CT})$**.**

Figure 4.5: User login, authentication and key establishment phase in the proposed scheme

Step SLA3:

1. $LSSNS_j$ generates 128-bit random number Ω_{s_j}.
2. $LSSNS_j$ computes $\mathcal{R}_1 = C \oplus \Omega_{s_j} \oplus T_{s_j}(C) \oplus T_{S_j}$. Here, T_{S_j} is the current timestamp of $LSSNS_j$.

3. $LSSNS_j$ generates its pseudo identity $PID_{S_j} = ID_{S_j} \oplus H(K_1)$.
4. $LSSNS_j$ computes $M_{S_j} = H(ID_{S_j}||\ C\ ||K_1\ ||\Omega_{s_j}||\ T_{S_j})$.
5. Finally, through a public channel, $LSSNS_j$ sends its login request $\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}$ to the cellular tower CT.

Step SLA4:

1. CT receives login message and verifies if $|T^*_{S_j} - T_{S_j}| \stackrel{?}{\leq} \triangle T$. If verification holds go to step 2, else *exit*. Here, $T^*_{S_j}$ is the current timestamp.
2. CT calculates $K'_1 = T_{X_\Gamma}(T_{s_j}(C))$.
3. CT calculates $ID'_{S_j} = PID_{S_j} \oplus H(K'_1) = ID_{S_j} \oplus H(K_1) \oplus H(K'_1) = ID_{S_j}$.
4. CT verifies if $K'_1 \stackrel{?}{=} K_1$. If verification holds, CT ensures that $ID'_{S_j} = ID_{S_j}$ and go to step 5.
5. CT finds the record $\langle ID_{S_j}, SN_j, r\rangle$ in the database.
6. CT further computes $C' = H(H(ID'_{S_j}||X_\Gamma) \oplus r)$.
7. CT computes $\mathcal{R}_2 = \mathcal{R}_1 \oplus T_{S_j} \oplus C' \oplus T_{s_j}(C) = H(H(ID_{S_j}\ ||X_\Gamma) \oplus r) \oplus \Omega_{s_j} \oplus T_{S_j} \oplus T_{s_j}(C)) \oplus T_{S_j} \oplus H(H(ID_{S_j}\ ||X_\Gamma) \oplus r) \oplus T_{s_j}(C) = \Omega_{s_j}$.
8. CT uses the received parameters T_{S_j} and calculates $\mathcal{R}_3 = H(ID'_{S_j}||C'||K'_1||\mathcal{R}_2||T_{S_j})$.
9. CT verify whether $\mathcal{R}_3 \stackrel{?}{=} \mathcal{R}_2$.
10. On the successful verification, CT accepts the login request and considers the Location Server $LSSNS_j$ as authentic. Otherwise, CT terminates the session and *exit*.

Step SLA5:

1. CT selects a 128-bit random number Ω_{CT}.
2. CT computes $\mathcal{R}_4 = C' \oplus \Omega_{CT} \oplus T_{CT} = H(H(ID_{S_j}\ ||X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT}$. Here, T_{CT} is the current timestamp of CT.
3. CT computes the mutually shared session key $SK_{CT,S_j} = H(C'||\ K'_1||\ T_{S_j}||\ T_{CT}||\ \mathcal{R}_2||\ \Omega_{CT})$.

4. CT computes $M_{CT} = H(ID_{S_j}|| SK_{CT,S_j}|| \mathcal{R}_2|| \Omega_{CT}|| T_{S_j}|| T_{CT})$.

5. Through a public channel, CT sends the authentication response message $\{\mathcal{R}_4, M_{CT}, T_{CT}\}$ to $LSSNS_j$.

Step SLA6:

1. $LSSNS_j$ Receives the authentication response message from CT.

2. $LSSNS_j$ Verifies the transmission delay $|T^*_{CT} - T_{CT}| \stackrel{?}{\leq} \triangle T$, where T^*_{CT} is the current timestamp. If verification holds, go to step 3, else *exit*.

3. $LSSNS_j$ computes $\mathcal{R}_5 = C' \oplus \mathcal{R}_4 \oplus T_{CT} = H(H(ID'_{S_j} ||X_\Gamma) \oplus r) \oplus H(H(ID_{S_j} ||X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT} \oplus T_{CT} = \Omega_{CT}$.

Step SLA7:

1. $LSSNS_j$ generates the session key mutually shared with CT as $SK_{S_j,CT} = H(C ||K_1 ||T_{S_j} ||T_{CT} ||\Omega_{CT} ||\mathcal{R}_5)$.

2. $LSSNS_j$ verifies $M_{CT} \stackrel{?}{=} H(ID_{S_j} ||SK_{S_j,CT} ||\Omega_{CT} ||\mathcal{R}_5 ||T_{S_j} ||T_{CT})$.

3. If verification succeeds, then $LSSNS_j$ confirms that the cellular tower CT is authentic and the current session key $SK_{S_j,CT}$ $(= SK_{CT,S_j})$ is mutually verified and established. Otherwise, discard the session key and *exit*.

The summary of the $LSSNS_j$ login, authentication and key establishment phase is shown in Figure 4.6.

4.3.4 The distance threshold registration phase

Every registered mOSN user MU_i needs to register a distance threshold to $LSSNS_j$ in which corresponding social friends can be searched and the message communications of Distance Threshold Registration Phase is shown in Figure 4.7.

Step DR1:

1. MU_i decides a distance threshold $\mathcal{D}_{f_{u_i}}$ beyond which MU_i does not allow his/her social friends to find himself in a friends' location query.

2. MU_i sends encrypted distance registration message $Msg^1_{dreg} = \langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i|| \mathcal{D}_{f_{u_i}}|| RN_{u_i}|| TS_{u_i}|| \mathcal{R}_{flag} = 1), H(ID_i|| RN_{u_i}|| TS_{u_i}), TS_{u_i}\rangle$.

Location Sharing Social Network Server ($LSSNS_j$)	Cellular tower (CT)
Login phase	
Input ID_{S_j} and PW_{S_j} Compute $RPW_1 = H(ID_{S_j}\|\|PW_{S_j})$ and $b_1 = E_2 \oplus PWB_1$ Verifies if stored $f_{S_j} = H(RPW_1\|\|b_1)$? If verification holds, Generate 128-bit number s_j and Ω_{s_j} Compute $C = E_1 \oplus H(ID_{S_j}\|\| PW_{S_j} \|\|b_1)$ $K_1 = T_{s_j}(T_{X_\Gamma}(C))$, and $\mathcal{R}_1 = C \oplus \Omega_{s_j} \oplus T_{s_j}(C) \oplus T_{S_j}$ Generate pseudo-identity, $PID_{S_j} = ID_{S_j} \oplus H(K_1)$ Compute $M_{S_j} = H(ID_{S_j}\|\| C \|\|K_1 \|\|\Omega_{s_j}\|\| T_{S_j})$ $\xrightarrow[\textbf{(public channel)}]{\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}}$	
Authentication phase	
	Verify if $\|T^*_{S_j} - T_{S_j}\| \leq \triangle T$? Compute $K'_1 = T_{X_\Gamma}(T_{s_j}(C))$, $ID'_{S_j} = PID_{S_j} \oplus H(K'_1) = ID_{S_j}$, Corresponding to ID_{S_j}, find the record $\langle ID_{S_j}, SN_j, r\rangle$ from database. Compute $C' = H(H(ID'_{S_j}\|\|X_\Gamma) \oplus r)$, $\mathcal{R}_2 = \mathcal{R}_1 \oplus T_{S_j} \oplus C' \oplus T_{s_j}(C)$, $= H(H(ID_{S_j}\|\|X_\Gamma) \oplus r) \oplus \Omega_{s_j} \oplus T_{S_j} \oplus T_{s_j}(C)) \oplus T_{S_j} \oplus$ $H(H(ID_{S_j}\|\|X_\Gamma) \oplus r) \oplus T_{s_j}(C) = \Omega_{s_j}$, Using T_{S_j}, Compute $\mathcal{R}_3 = H(ID'_{S_j}\|\|C'\|\|K'_1\|\|\mathcal{R}_2\|\|T_{S_j})$ Verify if $\mathcal{R}_3 \stackrel{?}{=} \mathcal{R}_2$ If verification holds, CT **accepts the login request.** Generate 128-bit random number Ω_{CT} Compute $\mathcal{R}_4 = C' \oplus \Omega_{CT} \oplus T_{CT}$, $= H(H(ID_{S_j}\|\|X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT}$ Compute session key $SK_{CT,S_j} = H(C'\|\| K'_1\|\| T_{S_j}\|\| T_{CT}\|\| \mathcal{R}_2\|\| \Omega_{CT})$, Compute hash value $M_{CT} = H(ID_{S_j}\|\| SK_{CT,S_j}\|\| \mathcal{R}_2\|\| \Omega_{CT}\|\| T_{S_j}\|\| T_{CT})$. $\xleftarrow[\textbf{(public channel)}]{\{\mathcal{R}_4, M_{CT}, T_{CT}\}}$
Verify if $\|T^*_{CT} - T_{CT}\| \leq \triangle T$? Compute $\mathcal{R}_5 = C' \oplus \mathcal{R}_4 \oplus T_{CT}$ $= H(H(ID'_{S_j}\|\|X_\Gamma) \oplus r) \oplus H(H(ID_{S_j}\|\|X_\Gamma) \oplus r)$ $\oplus \Omega_{CT} \oplus T_{CT} \oplus T_{CT} = \Omega_{CT}$, Compute session key as $SK_{S_j,CT} = H(C\|\|K_1\|\|T_{S_j}\|\|T_{CT}\|\|\Omega_{CT}\|\|\mathcal{R}_5)$ Compute and verify hash value $M_{CT} \stackrel{?}{=} H(ID_{S_j}\|\|SK_{S_j,CT}\|\|\Omega_{CT}\|\|\mathcal{R}_5\|\|T_{S_j}\|\|TS_2)$.	
If verification holds, **store session key** $SK_{S_j,CT} (= SK_{CT,S_j})$	**Store session key** SK_{CT,S_j} $(= SK_{S_j,CT})$**.**

Figure 4.6: The $LSSNS_j$ login, authentication and key establishment phases

Here, RN_{u_i}, TS_{u_i}, $H(\cdot)$ and $E(\cdot)$ convey their meaning as tabulated in Table 4.1. $\mathcal{R}_{flag} = 1$ indicates that this message in intended for the distance threshold registration.

Step DR2:

1. CT uses session key $\mathcal{SK}_{CT,MU_i}$ and decrypts $D_{\mathcal{SK}_{CT,MU_i}}(Msg^1_{dreg})$.

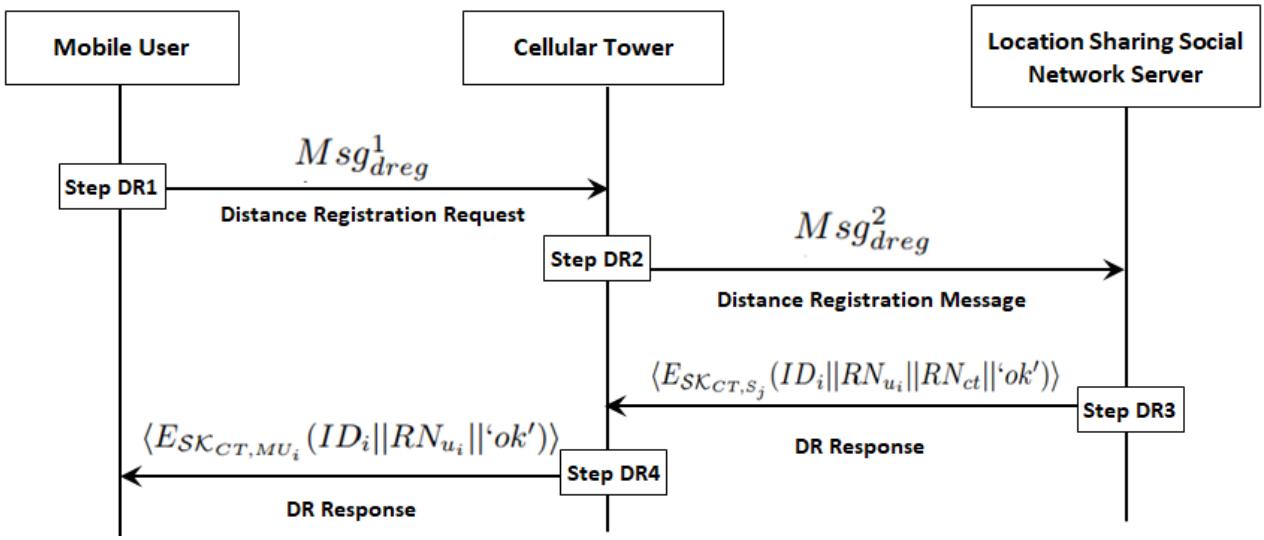

Figure 4.7: The message communication in distance threshold registration phase

2. CT verifies if $|TS^*_{u_i} - TS_{u_i}| \stackrel{?}{\leq} \triangle T$, where $TS^*_{u_i}$ is the current timestamp. If verification holds, go to step 3, else terminate and *exit*.

3. CT computes the hash value $H(ID_i||\ RN_{u_i}||\ TS_{u_i})$. If the computed and received hash values are same, then go to step 4, else discards the message and *exit*.

4. CT makes a login to $LSSNS_j$ and establishes the shared session key $\mathcal{SK}_{CT,S_j}$ as explained in subsection 4.3.3.

5. CT encrypts and sends the distance registration message as $Msg^2_{dreg} = \langle E_{\mathcal{SK}_{CT,S_j}}(ID_i||\ \mathcal{D}_{f_{u_i}}||\ RN_{u_i}||\ RN_{ct}||\ TS_{ct}||\ \mathcal{R}_{flag} = 1),\ H(ID_i||\ RN_{ct}||\ TS_{ct}),\ TS_{ct}\rangle$ to $LSSNS_j$.

Step DR3:

1. $LSSNS_j$ decrypts Msg^2_{dreg} using the session key $\mathcal{SK}_{S_j,CT}$.

2. $LSSNS_j$ verifies communication delay using the received and current timestamp values.

3. verifies message integrity and authenticity by computing and comparing hash values with decrypted parameters.

4. If verifications of step (2) and (3) are successful, then go to Step 5, else terminate the session and *exit*.

5. $LSSNS_j$ saves record $\{ID_i, \mathcal{D}_{f_{u_i}}\}$ and sends response message $Msg^1_{resp} = E_{\mathcal{SK}_{CT,S_j}}(ID_i \; ||RN_{u_i} \; ||RN_{ct} \; ||\text{`}ok\text{'})$ to CT.

Step DR4:

1. CT decrypts Msg^1_{resp} using the shared session key.
2. CT verifies the received random number RN_{ct} and sends $Msg^2_{resp} = E_{\mathcal{SK}_{CT,MU_i}}(ID_i||RN_{u_i}||\text{`}ok\text{'})$ to MU_i.
3. MU_i decrypts the message Msg^2_{resp} using the shared session key $\mathcal{SK}_{MU_i,CT}$
4. MU_i verifies the random number RN_{u_i}. If these verification holds, go to step 5.Otherwise, termiate the session and *exit*.
5. MU_i reads '*ok*' message and distance registration process successfully terminates.

4.3.5 The user location update phase

In this subsection, I have described how mOSN user MU_i updates his current location to the Location Sharing Social Network Server $LSSNS_j$ and the message communications of User Location Update Phase is shown in Figure 4.8. The location updation is done through the cellular tower CT, following the steps as mentioned in subsection 4.3.2, MU_i makes a secure login to CT and mutually estabsishes a shared session key $\mathcal{SK}_{MU_i,CT}(= \mathcal{SK}_{CT,MU_i})$. The user location update phase is summarized in Figure 4.9. Next, it executes the following steps:

Step LU1:

1. MU_i selects a one-time 128-bit random number RN_{u_i}.
2. MU_i uses the shared session key $\mathcal{SK}_{MU_i,CT}$ and sends an encrypted message $Msg_1 = E_{\mathcal{SK}_{MU_i,CT}}(ID_i||\; x_{u_i}||\; y_{u_i}||\; E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\; y_{u_i})||\; RN_{u_i}||\; TS_{u_i})$ to CT.
3. MU_i sends a hash value $H_1 = H(ID_i||\; x_{u_i}||\; y_{u_i}||\; E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\; y_{u_i})||\; RN_{u_i}||\; TS_{u_i})$ to CT.

   Note that, (x_{u_i}, y_{u_i}) is the current location of MU_i, and $E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\; y_{u_i})$ is the current location of MU_i encrypted with the symmetric key $\{\mathcal{K}_{MU_i,\mathcal{F}}\}$, mutually shared between MU_i and all trusted friend $\mathcal{F}$.

Step LU2:

1. CT receives location update message $\{Msg_1, H_1\}$ from MU_i.

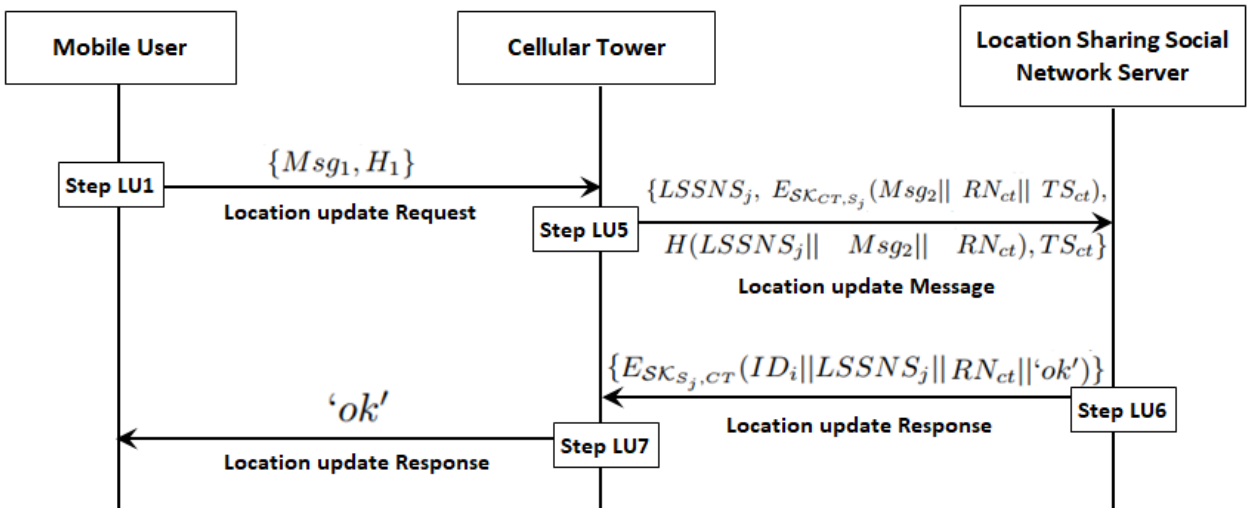

Figure 4.8: The message communication in user location update phase

2. CT uses $\mathcal{SK}_{CT,MU_i}$ and decrypts Msg_1 as $D_{\mathcal{SK}_{CT,MU_i}}(Msg_1)$.

3. CT retrieves parameters ID_i, (x_{u_i}, y_{u_i}), $E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\ y_{u_i})$, RN_{u_i} and TS_{u_i} respectively (from step 2).

4. CT verifies if $|TS^*_{u_i} - TS_{u_i}| \stackrel{?}{\leq} \triangle T$, where $TS^*_{u_i}$ is the current timestamp. If the verification holds then go to step $LU3$, else discards the received message and $exit$.

Step LU3:

1. CT uses the decrypted parameter and computes a hash value $H_2 = H(ID_i||\ x_{u_i}||\ y_{u_i}||\ E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\ y_{u_i})||\ RN_{u_i}||\ TS_{u_i})$.

2. CT verifies if $H_2 \stackrel{?}{=} H_1$. If the verification holds then go to step 3, else terminate the session and $exit$.

3. CT confirms the authenticity and integrity of the message and makes a login to $LSSNS_j$.

4. CT and $LSSNS_j$ establishes a mutually shared session key $\mathcal{SK}_{CT,S_j}$ as mentioned in subsection 4.3.3.

Step LU4:

1. CT generates $\mathcal{L}-1$ dummy locations and $\mathcal{L}-1$ dummy encrypted string chosen randomly as $\{x^*_i, y^*_i, enc^*_i\}_{i=1...\mathcal{L}-1}$.

Mobile user (MU_i)	Cellular Tower (CT)	Location Storing Social Network Server ($LSSNS_j$)
$\xrightarrow{\{\text{Login request to CT}\}}$ **(public channel)**		
	Establish **session key** $\mathcal{SK}_{CT,MU_i}$ $\xleftarrow[\textbf{(public channel)}]{\{\text{Authentication response to } MU_i\}}$	
Establish **session key** $\mathcal{SK}_{MU_i,CT}$ Compute $Msg_1 = E_{\mathcal{SK}_{MU_i,CT}}(ID_i \| x_{u_i} \| \| y_{u_i} \| \| E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}) \| RN_{u_i} \| TS_{u_i})$ $H_1 = H(ID_i \| x_{u_i} \| y_{u_i} \| E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}) \| RN_{u_i} \| TS_{u_i})$ $\xrightarrow[\textbf{(public channel)}]{\{Msg_1, H_1\}}$		
	Decrypts Msg_1 as $D_{\mathcal{SK}_{CT,MU_i}}(Msg_1)$ Verify if $\|TS^*_{u_i} - TS_{u_i}\| \leq \triangle T$? Compute $H_2 = H(ID_i \| x_{u_i} \| y_{u_i} \| E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}) \| RN_{u_i} \| TS_{u_i})$ Verify if $H_2 = H_1$? **[accept/reject]** $\xrightarrow[\textbf{(public channel)}]{\{\text{Login request to } LSSNS_j\}}$	
		Establish **session key** $\mathcal{SK}_{S_j,CT}$ $\xleftarrow[\textbf{(public channel)}]{\{\text{Authentication response}\}}$
	Establish **session key** SK_{CT,S_j} Randomly generate $\{x^*_i, y^*_i, enc^*_i\}_{i=1\ldots\mathcal{L}-1}$ Set real location at n^{th} place $(1 \leq n \leq \mathcal{L})$ Compute $Index_{u_i} = E_{X_\Gamma}[Seq(x_{u_i} \| y_{u_i} \| E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}))]$ $Msg_2 = \{ID_i, (x^*_1, y^*_1, enc^*_1), \ldots, (x_{u_i}, y_{u_i}, E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i})), \ldots, (x^*_{\mathcal{L}}, y^*_{\mathcal{L}}, enc^*_{\mathcal{L}}), Index_{u_i}\}$ $H_3 = H(LSSNS_j \| Msg_2 \| RN_{ct})$ $\xrightarrow[\textbf{(public channel)}]{\{E_{\mathcal{SK}_{CT,S_j}}(Msg_2 \| RN_{ct} \| TS_{ct}), H_3, TS_{ct}\}}$	
		Decrypts $D_{\mathcal{SK}_{S_j,CT}}(Msg_2)$ Verify if $\|TS^*_{ct} - TS_{ct}\| \leq \triangle T$? Compute and verify hash value **[accept/reject]** Update the user location. Compute $Msg_3 = E_{\mathcal{SK}_{S_j,CT}}(ID_i \| LSSNS_j \| \| RN_{ct} \| \| \text{`}ok\text{'})$ $\xleftarrow[\textbf{(public channel)}]{\{Msg_3\}}$
	Decrypts $D_{\mathcal{SK}_{CT,S_j}}(Msg_3)$ Verify correctness of RN_{ct} **[accept/reject]**	

Figure 4.9: User location update phase of the proposed scheme

2. CT randomly put MU_i's real updated location information string at the n^{th} place among the dummy information set, $(1 \leq n \leq \mathcal{L})$.

3. The sequence number of MU_i's real location update information is encrypted by CT with its own master secret key X_Γ, i.e., $Index_{u_i} = E_{X_\Gamma}[Sequence(x_{u_i}|| \ y_{u_i}|| \ E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}))]$.

Step LU5:

1. CT prepares the message $Msg_2 = \{ID_i, (x_1^*, y_1^*, enc_1^*), \ldots, (x_{u_i}, y_{u_i}, E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i})), \ldots, (x_{\mathcal{L}}^*, y_{\mathcal{L}}^*, enc_{\mathcal{L}}^*), Index_{u_i}\}$.

2. CT generates a 128-bit random number RN_{ct}.

3. sends $\{LSSNS_j, E_{\mathcal{SK}_{CT,S_j}}(Msg_2|| \ RN_{ct}|| \ TS_{ct}), H(LSSNS_j|| \ Msg_2|| \ RN_{ct}), TS_{ct}\}$ to server $LSSNS_j$.

Step LU6:

1. $LSSNS_j$ uses its session key $\mathcal{SK}_{S_j,CT}$ (shared with CT) and decrypts the message Msg_2, random number RN_{ct}, and timestamp TS_{ct}.

2. $LSSNS_j$ checks the transmission delay using the received and current timestamps.

3. $LSSNS_j$ checks message integrity by checking computing a fresh hash value from the decrypted parameters.

4. $LSSNS_j$ updates the user location and sends $Msg_3 = E_{\mathcal{SK}_{S_j,CT}}(ID_i||LSSNS_j|| \ RN_{ct}||`ok')$ to CT.

Step LU7:

1. CT uses the session key $\mathcal{SK}_{CT,S_j}$ and decrypts Msg_3.

2. CT verifies the correctness of RN_{ct}. If it is correct, go to step 3, else terminate the session and *exit*.

3. CT forwards 'ok' to MU_i.

Remark 4.1. *When the user reaches a new place, he/she updates his/her location in the LSSNS's database to ensure that LSSNS knows the user's real-time location. MU_i executes the user location update phase and sends the current location coordinate (x_{u_i}, y_{u_i}) (obtained by GPS) to LSSNS. As the user location update phase of our proposed scheme is based only on the private key encryption, cryptographic hash function and xor operation, it is both secure and lightweight. Table 4 reveals that considering all the entities, this location update process takes only 0.0721 second. Hence, LSSNS can update the current location of MU_i very quickly.*

*Depending on the population density, potential users, etc., the LTE technology nowadays requires cellular towers (or BTSs) to be spaced in the range of 2km to 5km [188]. When the mOSN user MU_i moves to a new cellular tower zone, he/she needs to register to CT. Once the registration is complete, MU_i provides his/her location update to LSSNS and access other location-based services from LSSNS. This mobile user registration process is a one-time task and incurs very small computation cost on a mobile device. As presented in Figure 2 and Table 3, MU_i registration has the computation cost of 5*T_H + 5*T_X + T_{FE}, which essentially takes only 0.0656 second. This evidences that the user registration process is very efficient.*

4.3.6 The friends' locations query phase

In this subsection, I have described how MU_i achieves his/her social friends' identity and location, who are willing to share their information and the message communications of Friends' Locations Query Phase is shown in Figure 4.10.

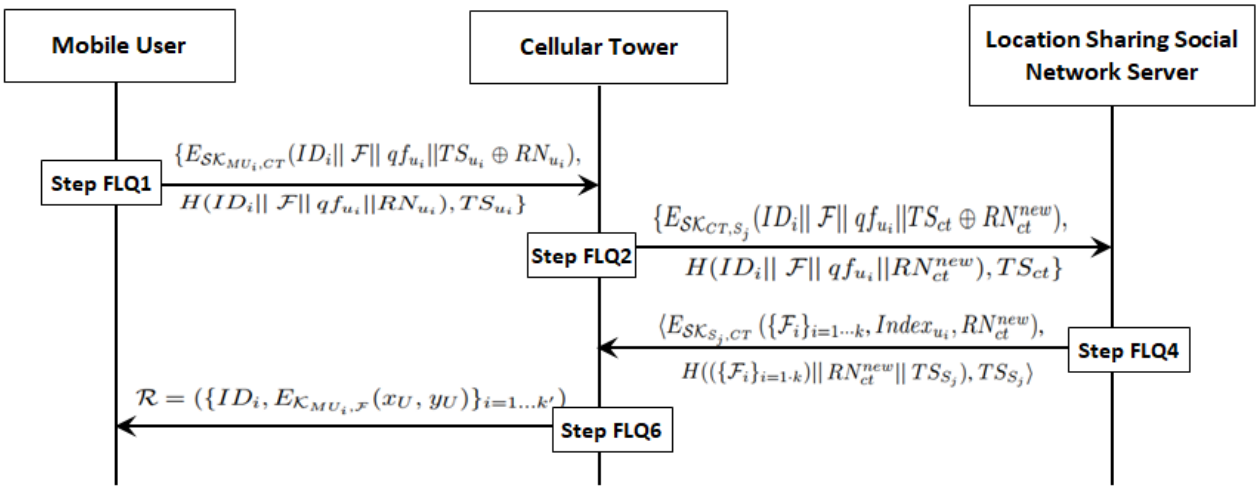

Figure 4.10: The Message Communication in Friends' Locations Query Phase.

Step FLQ1:

1. MU_i makes a secure login to CT and mutually establishes a shared session key $\mathcal{SK}_{MU_i,CT}(=\mathcal{SK}_{CT,MU_i})$ (As explained in subsection 4.3.2).

2. MU_i sends $\{E_{\mathcal{SK}_{MU_i,CT}}(ID_i||\ \mathcal{F}||\ qf_{u_i}||TS_{u_i} \oplus RN_{u_i}),\ H(ID_i||\ \mathcal{F}||\ qf_{u_i}||RN_{u_i}), TS_{u_i}\}$ to CT.

   Note that the message $\mathcal{F}$ is a request to find 'friends'. TS_{u_i}, RN_{u_i}, $E(\cdot)$ and $H(\cdot)$ convey their usual meanings as explained in Table 4.1.

Step FLQ2:

1. CT receives request from MU_i, and checks if $|TS^*_{u_i} - TS_{u_i}| \stackrel{?}{\leq} \triangle T$. Here, $TS^*_{u_i}$ is the current timestamp. If the verification holds, go to step 2, else *exit*.

2. CT uses its session key $\mathcal{SK}_{CT,MU_i}$ (shared with MU_i) to decrypt the encrypted user message.

3. CT uses received timestamp TS_{u_i} and parameter $TS_{u_i} \oplus RN_{u_i}$ to retrieve the random number as $RN'_{u_i} = TS_{u_i} \oplus RN_{u_i} \oplus TS_{u_i}$.

4. CT computes hash value $H_3 = H(ID_i||\ \mathcal{F}||\ qf_{u_i}||RN'_{u_i})$. If the computed H_3 and the received hash value does not match, then CT rejects the request immediately. Otherwise, go to step 5.

5. CT logins to the server $LSSNS_j$ and creates a shared session key $\mathcal{SK}_{CT,S_j}$, as explained in subsection 4.3.3.

6. Through a public channel, CT forwards $\{E_{\mathcal{SK}_{CT,S_j}}(ID_i\ ||\mathcal{F}||\ qf_{u_i}||TS_{ct} \oplus RN^{new}_{ct}),\ H(ID_i||\ \mathcal{F}||\ qf_{u_i}||\ RN^{new}_{ct}),\ TS_{ct}\}$ to $LSSNS_j$.

Step FLQ3:

1. $LSSNS_j$ receives the message from CT and decrypts the message using its session key $\mathcal{SK}_{S_j,CT}$ (shared with CT).

2. $LSSNS_j$ and checks the communication delay using current timestamp TS^*_{ct} and the received timestamp TS_{ct}. if verification holds, go to step 3, else terminate the session and *exit*.

3. $LSSNS_j$ retrieves RN_{ct}^{new} and computes fresh hash value with the decrypted parameter and compares with the received hash value.

4. If verification holds, go to step *FLQ4*, else terminate the session and *exit*.

Step FLQ4:

1. $LSSNS_j$ finds the set Θ containing a database entry for all friends of MU_i.

2. $LSSNS_j$ finds whether $\delta((x_p, y_p), (x_{u_{i_t}}, y_{u_{i_t}})) \leq min(qf_{u_i}, df_s)_{s\in\Theta}, p = 1, \cdots, k$, and $t = 1, \cdots, k$, where $\delta(\cdot)$ is the distance function and $(x_{u_{i_t}}, y_{u_{i_t}})$ are one real and $k-1$ fake locations of MU_i. Here, database entry of ID_i is excluded.

3. For all friends $\alpha \in \Theta$, $LSSNS_j$ includes record $(\alpha, (p, enc_p^*), Index_\alpha)$ in the result set if the coordinate $(x_{\alpha_{i_t}}, y_{\alpha_{i_t}})$ meets the distance requirement.

4. Corresponding to k coordinate entries of MU_i $(x_{u_i}, y_{u_i})_{i=1\cdots k}$, $LSSNS_j$ prepares k subsets $\{\mathcal{F}_i\}_{i=1\cdots k}$ and adds them to result set.

5. $LSSNS_j$ uses RN_{ct}^{new} (the random number sent by CT), TS_{S_j} (the current timestamp) and encrypts the result set using the shared session key $\mathcal{SK}_{S_j,CT}$.

6. Through public channel, $LSSNS_j$ forwards message $\langle E_{\mathcal{SK}_{S_j,CT}}$ $(\{\mathcal{F}_i\}_{i=1\cdots k}, Index_{u_i}, RN_{ct}^{new}), H((\{\mathcal{F}_i\}_{i=1\cdots k})||\ RN_{ct}^{new}||\ TS_{S_j}), TS_{S_j}\rangle$ to CT.

Step FLQ5:

1. CT receives the encrypted result set from $LSSNS_j$.

2. CT decrypts it using own shared session key $\mathcal{SK}_{CT,S_j}$ to obtain $\{\mathcal{F}_i\}_{i=1\cdots k}$, $Index_{u_i}$, and RN_{ct}^{new}.

3. CT verifies transmission delay by comparing the current timestamp $TS_{S_j}^*$ and the received timestamp TS_{S_j}.

4. CT verifies the value of received RN_{ct}^{new} with stored RN_{ct}^{new}.

5. If both verification of step 3 and 4 is successful, then go to step $FLQ6$, else *exit*.

Step FLQ6:

1. CT decrypts $Index_{u_i}$ as $D_{X_\Gamma}(Index_{u_i}) = D_{X_\Gamma}\ (E_{X_\Gamma}(Seq(x_{u_i}||\ y_{u_i}||\ E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},\ y_{u_i}))))$ and retrieves the real sequence number γ. CT uses its master secret key for the decryption.

2. CT discurds all records $\{\mathcal{F}_i\}_{i\neq\gamma}$ and accepts only $\mathcal{F}_\gamma$.

3. CT finds every present user U in the dataset $\mathcal{F}_\gamma$.

4. CT decrypts $Index_U$ and finds its real center point location γ_U.

5. CT retrieves $enc_{\gamma_U} = E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_U,\ y_U)$ and prepares answer set $\mathcal{R}$.

6. CT sends the friend set $\mathcal{R} = (\{ID_{U_i}, E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_U,\ y_U)\}_{i=1\ldots k'})$ to MU_i.

Remark 4.2. *The existing location sharing schemes for OSN suffer from multiple security drawbacks. The purpose of our research is to design a secure and efficient location sharing scheme for OSN. User location updates and friend's location queries are two essential operations for location sharing services. As mentioned in existing location sharing schemes [187], [113], [194], group key-establishment among a user and its trusted social friends is an intrinsic requirements. In the literature, several group key distribution and key-establishment schemes among social friends have been proposed in distributed online social networks [93], [78], [183], [75].*

Unlike those schemes, our proposed one is not designed for purpose group key distribution and key-establishment among social friends. That said, the key-establishment process among social friends advocated by Y. Jung et al. [93] and L. Guo et al. [75] can be adapted to work with my proposed scheme.

4.4 Security Analysis

In this section, I provide the detail security analysis of the proposed scheme. This is done in two ways. First, I present the authentication proof Using Burrows-Abadi-Needham (BAN) logic. Second, I present an informal security analysis to logically explain how and why the proposed scheme resists various security attacks.

4.4.1 Authentication proof using BAN logic

BAN logic is used to analyze the security of any authentication scheme to verify the secure transmission between two communicating parties of that network [31]. In this section, I use BAN logic to show that the proposed scheme actually achieves the authentication goals. The basic syntax and semantics of BAN logic are explained in Table 4.2.

Table 4.2: Notations and their descriptions used in BAN logic

Symbol	Description
$Q \mid\equiv S$	Q believes that the statement S is true
$Q \lhd S$	Q can see the statement S
$\#(S)$	Formula S is considered as fresh
$Q \mid\sim S$	Q said the statement S once
$Q \Rightarrow S$	Q keeps jurisdiction over the statement S
$\langle S \rangle_T$	Formula S is combined with the formula T
$Q \overset{K}{\longleftrightarrow} R$	Only Q and R know the value of the key K and it is used for communication between them
$Q \overset{S}{\rightleftharpoons} R$	Only Q and R know the secret statement S. Principals trusted by Q & R may know S
SK	Current session key

The main logical postulates of the BAN logic are defined by a set of laws or rules as listed below [31], [172].

- Law 1 (Message Meaning Law (MML)).
  $\frac{Q \mid\equiv R \overset{K}{\rightleftharpoons} Q, Q \lhd \langle S \rangle_K}{Q \mid\equiv R \mid\sim S}$.
- Law 2 (Nonce Verification Law (NVL)).
  $\frac{Q \mid\equiv \#(S), Q \mid\equiv R \mid\sim S}{Q \mid\equiv R \mid\equiv S}$.
- Law 3 (Freshness Conjuncatenation Law (FCL)).
  $\frac{Q \mid\equiv \#(S)}{Q \mid\equiv \#(S,T)}$.
- Law 4 (Jurisdiction Law (JL)).
  $\frac{Q \mid\equiv R \Rightarrow S, Q \mid\equiv R \mid\equiv S}{Q \mid\equiv S}$.

- Law 5 (Additional Laws (AL)). $\frac{Q|\equiv(S,T)}{Q|\equiv S}$, $\frac{Q\lhd(S,T)}{Q\lhd S}$, $\frac{Q|\equiv R\sim(S,T)}{Q|\equiv R\sim S}$.

In order to show that the proposed scheme ensures authentication, two goals, as mentioned in the following, must be achieved.

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

In the proposed scheme, there will be two basic types of messages as follows:

- **Message 1.** $MU_i \rightarrow CT$: $\{ TID_i^*, H(H(ID_i \oplus x_{CT_{u_i}} || x_\Gamma)) \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}), \mathcal{T}_{u_i}, H_1\}$.
- **Message 2.** $CT \rightarrow MU_i$: $\{B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i, \mathcal{T}_{ct}, H_3\}$.

The above generic messages have to be converted to idealized messages. These idealized messages are as follows.

- **Message 1.** $MU_i \rightarrow CT$: $\{TID_i, \mathcal{T}_{u_i}, \langle ID_i, x_{CT_{U_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma, H_1\}$.
- **Message 2.** $CT \rightarrow A : \{\mathcal{T}_{ct}, \langle \Omega_{ct}\ \mathcal{T}_{ct}, ID_i, \rangle x_\Gamma, H_3\}$.

With the following assumptions, the authentication proof of our proposed scheme is presented as follows:

- A.1: $MU_i \mid\equiv \#(T_{ct})$;
- A.2: $CT \mid\equiv \#(T_{u_i})$;
- A.3: $MU_i \mid\equiv (MU_i \overset{A_{u_i CT}}{\rightleftharpoons} CT)$;
- A.4: $CT \mid\equiv (MU_i \overset{A_{u_i CT}}{\rightleftharpoons} CT)$;
- A.5: $MU_i \mid\equiv CT \Rightarrow (ID_{ct}, \Omega_{ct}, \mathcal{T}_{ct})$;
- A.6: $CT \mid\equiv MU_i \Rightarrow (ID_i, \Omega_{u_i}, T_{u_i})$;
- A.7: $MU_i \mid\equiv T_{u_i}$;
- A.8: $MU_i \mid\equiv \Omega_{u_i}$;

- A.9: $MU_i \mid\equiv ID_i$;
- A.10: $MU_i \mid\equiv ID_{CT}$;
- A.11: $CT \mid\equiv T_{ct}$;
- A.12: $CT \mid\equiv \Omega_{ct}$;
- A.13: $CT \mid\equiv ID_{CT}$.

Next, I shall show that two goals mentioned earlier can be achieved using the assumptions, idealized messages and Basic BAN logic laws.
From the first message, I may obtain the following.

- S_1: $CT \lhd \{ID_i, \mathcal{T}_{u_i}, \langle ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma, H_1\}$.
- S_2: Using AL, I can derive: $CT \lhd \langle ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma$.
- S_3: According to A.4 and MML, I obtain, $CT \mid\equiv MU_i \mid\sim (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_4: According to A.2 and FCL, I get, $CT \mid\equiv \#(ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_5: According to NVL, I have, $CT \mid\equiv MU_i \mid\equiv (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_6: Using A.6 and JL, I get, $CT \mid\equiv (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_7: From S_6 and AL, I obtain, $CT \mid\equiv \Omega_{u_i}$, $CT \mid\equiv \mathcal{T}_{u_i}$, $CT \mid\equiv ID_i$.
- S_8: According to A.11, A.12, A.13, I get, $CT \mid\equiv ID_{CT}$, $CT \mid\equiv \mathcal{T}_{ct}$ and $CT \mid\equiv \Omega_{ct}$.
- S_9: Since $SK_{CT,MU_i} = H(ID_i \mid\mid ID_{CT} \mid\mid B_{CT_{u_i}} \mid\mid \mathcal{P}1 \mid\mid \Omega_{ct} \mid\mid \mathcal{T}_{u_i} \mid\mid \mathcal{T}_{ct})$ and the results in Steps S_7 and S_8 give $CT \mid\equiv (MU_i \overset{SK_{CT,MU_i}}{\longleftrightarrow} CT)$. (**Goal 2**)
- S_{10}: Using the message 2 and AL, I obtain, $MU_i \lhd \langle\, \Omega_{ct}, \mathcal{T}_{ct}\rangle_{x_\Gamma}$.
- S_{11}: According to A.3 and MML, I get, $MU_i \mid\equiv CT \mid\sim (\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{12}: Using A.1 and FCL, I obtain, $MU_i \mid\equiv \#(\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{13}: Using NVL, I obtain, $MU_i \mid\equiv CT \mid\equiv (\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{14}: A.5 and JL give $MU_i \mid\equiv (\Omega_{ct}, \mathcal{T}_{ct})$.

- S_{15}: According to S_{14} and AL, I have, $MU_i \mid\equiv \Omega_{ct}$, $MU_i \mid\equiv \mathcal{T}_{ct}$.
- S_{16}: According to A.7-A.10, I obtain, $MU_i \mid\equiv ID_i$, $MU_i \mid\equiv ID_{CT}$, $MU_i \mid\equiv \mathcal{T}_{u_i}$, $MU_i \mid\equiv \Omega_{ct}$.
- S_{17}: The results of Steps S_{15} and S_{16} give $MU_i \mid\equiv (MU_i \overset{SK_{CT,MU_i}}{\longleftrightarrow} CT)$. (**Goal 1**)

Consequently, both the goals are achieved to ensure that mutual authentication between MU_i and CT is established.

4.4.2 Informal security analysis

In this section, I present an informal analysis of the security of the proposed scheme. This analysis aims to logically show that our scheme can successfully defend against the following known attacks.

The Replay Attack

In the proposed scheme, two message communications are needed by the login phase and the authentication phase. In the process of login, MU_i sends $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to CT, whereas in authentication phase, CT sends $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i. CT does not accept Msg_1 if $|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}| \geq \triangle T$. Additionally, CT computes $H_2 = H(ID_i \,||\mathcal{M}_1\, ||\mathcal{P}_1 \,||\mathcal{T}_{u_i})$ and checks whether H_2=H_1 or not. This computation is crucial to prevent a replay attack. The cellular tower CT rejects any request for log-in if this checking does not succeed. I have explained in *Step LA*5 in Section 4.3.2, of the mOSN user login, authentication and key establishment phase how an attacker cannot succeed in replaying the authentication message Msg_2. Moreover, CT also stores parameters $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i}\rangle$ in its repository. In case CT receives another login request message, say $Msg_1^n = \{TID_n^*, \mathcal{M}_1^n, H_1^n, \mathcal{T}_{u_n}\}$, it first checks whether T_{u_n} is valid or not. If it is found to be valid, CT gies on to check whether the extracted $TID_n^* = TID_n \oplus H(ID_{CT} \,||\mathcal{T}_{u_n})$ is the same as the TID_n stored in its repository for the same ID_n. If they are the same, Msg_1^n is considered being a replay message. Thus, our proposed scheme is capable of resisting a strong replay attack with the help of current timestamp and a random nonce.

The Man-in-the-middle Attack

An adversary $\mathcal{A}$ may attempt to modify login or authentication message through a man-in-the-middle attack. In order to execute this attack, $\mathcal{A}$ set up an independent parallel connection

with both MU_i and CT for a specific session. Additionally, to invalidate the login request of an authorized user, the attacker may modify some parameters from the request message. In the proposed scheme, the credentials of both login and authentication message, such as ID_{CT}, RID_{Γ}, $A_{U_{iCT}}$, etc. are generated with fuzzy extractor, hash function, bitwise XOR and random nonce. This makes adversary $\mathcal{A}$ very difficult to regenerate and modify. As a consequence, the proposed scheme can resist the man-in-the-middle attack.

The Stolen/Lost Mobile Device Attack

Suppose the mobile device of the user MU_i has been stolen or lost, an adversary can easily find P_i^1 and P_i^2, which are stored in the memory of the device. However, ID_i, PW_i, and biometric η_i are not stored directly in the device. From stored $P_i^1 = H(PW_i|| \ \eta_i) \oplus n$ and $P_i^2 = H(ID_i \ ||PW_i \ ||\eta_i \ ||n)$, it is computationally infeasible to identify or predict all these parameters. Furthermore, P_i^1 and P_i^2 are masked with a random number n and the collision-resistant hash function $H(\cdot)$. This makes it a computationally infeasible problem to predict all the credentials in polynomial time. Therefore, the proposed scheme resists this type of attacks.

The Offline Password Guessing Attack

As describe in Section 4.3.2, a mobile user MU_i needs the identity ID_i and password PW_i for its login. An adversary can obtain the P_i^1 and P_i^2 from the lost or stolen mobile device, but it cannot guess and compute identity ID_i, password PW_i, and biometric η_i at the same time as it is computationally infeasible. Hence, this scheme can prevent the offline password guessing attack.

Known Key Secrecy/Forward Secrecy

An adversary may obtain the current session key, but with that compromised session key, it cannot compute previous session keys. As per the proposed scheme, the session key is computed as $SK_{CT,MU_i} = SK_{MU_i,CT} = H(ID_i \ ||ID_{CT} \ ||B_{CT_{U_i}} \ ||\mathcal{P}_1 \ ||\Omega_{ct} \ ||\mathcal{T}_{u_i} \ ||\mathcal{T}_{ct})$ where $A_{U_{iCT}} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \ ||X_{\Gamma})$. With the use of Ω_i, T_{u_i}, Ω_j, and T_{u_j}, a new login key for each session, $SK_{CT,MU_i} = SK_{MU_i,CT}$ is generated freshly and uniquely. So, the key cannot be used further in future. Moreover, before establishing a session key, both MU_i and CT mutually validated each other. Hence, the proposed scheme confirms that the leakage of

temporal information does not break the secrecy of the session key and it provides the session key security.

User Anonymity

In this proposed scheme, the anonymity property of any mobile user is maintained. An adversary may eavesdrop a login or authentication message communicated between MU_i and CT, but adversary cannot get the original ID_i from those messages. At the time of login MU_i send $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to cellular tower CT. Instead of its original identity MU_i send its temporary identity TID_i embedded in $TID_i^* = TID_i \oplus H(ID_{CT} || \mathcal{T}_{u_i})$, which is valid for only one session. Furthermore, it is not possible to compute ID_i from $\mathcal{M}_1 = A_{U_{iCT}} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$ and $H_1 = H(ID_i || \mathcal{M}_1 || \Omega_{u_i} || \mathcal{T}_{u_i})$. At the time of authentication, CT transmits back a authentication response message $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i where $\mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$ and $H_3 = H(ID_i || \mathcal{P}_1 || \Omega_{ct} || \mathcal{T}_{u_i} || \mathcal{T}_{ct} || SK_{CT,MU_i})$. So, from any intrude message, it is not feasible to figure out the original ID_i by an adversary. Thus, the proposed scheme can preserve the anonymity property of any user.

The Parallel Session and Reflection Attack

In the proposed scheme, an adversary cannot start a new session with CT using any fake identity, obtaining from any eavesdropped messages $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$. As described in section 4.3.2, an adversary cannot obtain the correct identity ID_i, password PW_i or the biometric key η_i of any legal user MU_i with an offline password guessing attack. Hence, from any eavesdropped message, an attacker cannot create a valid login request message Msg_1, so a new session with CT as a legal user not possible. Thus, our proposed scheme can protect the parallel session and reflection attacks.

The Privileged-insider Attack

This kind of attacks is launched by an internal user who may be authorized to use the system that is attacked. Suppose that an adversary, who is an internal user also, obtains the registration credentials ID_i, $(MPWB_i \oplus \lambda)$ from the mobile registration request Msg_1. However, as discussed in section 4.3.2, it is not feasible to compute the PW_i and the biometric key $\eta_i =$ even if the adversary has that lost or stolen mobile device. Without the knowledge of λ, it also not possible to calculate $MPWB_i$ from $(MPWB_i \oplus \lambda)$. So, our scheme can resist this type of attacks.

Session Key Security

For establishing a new session, a mutually computed session key $SK_{MU_i,CT}$ $(= SK_{CT,MU_i})$ is shared between MU_i and CT. The session key is computed as follows:

$$\begin{aligned} SK_{CT,MU_i} &= H(ID_i||ID_{CT}||B_{CT_{U_i}}||\mathcal{P}_1||\Omega_{ct}||\mathcal{T}_{u_i}||\mathcal{T}_{ct}) \\ &= H(ID_i||ID_{CT}||A_{U_{iCT}}||\mathcal{P}_1||\Omega_{ct}||\mathcal{T}_{u_i}||\mathcal{T}_{ct}) \\ &= H(ID_i||ID_{CT}||A_{U_{iCT}}||\Omega_{u_i}||\Omega_{ct}||\mathcal{T}_{u_i}||\mathcal{T}_{ct}) \\ &= H(ID_i||ID_{CT}||A_{U_{iCT}}||\Omega_{u_i}||\mathcal{P}_2||\mathcal{T}_{u_i}||\mathcal{T}_{ct}) \\ &= SK_{MU_i,CT} \end{aligned}$$

Both MU_i and CT authenticate each other to compute the mutually shared session key. Moreover, an adversary needs the credentials ID_i, ID_{CT}, $B_{CT_{U_i}} = (A_{U_{iCT}})$ for computing session keys. Therefore, the session keys are fully secured in our proposed scheme.

```
(* ———— channels ————*)
free pch: channel. (* public channel *)
free sch: channel [private]. (* private channel *)
(* ———— shared keys ———— *)
free SKmuct:bitstring [private].(* the session key of user *)
free SKctmu:bitstring [private]. (* the session key of server *)
(* ———— servers secret key ———— *)
free Xtau:bitstring [private].
free XctU:bitstring [private].
(* ———— constants ———— *)
free IDCT:bitstring [private].
free IDmu:bitstring [private].
free PW:bitstring [private].
const Bi:bitstring [private].
const dfu:bitstring [private].
const dsu:bitstring [private].
```

Figure 4.11: Code for channel declarations, keys, and constants

The Ephemeral Secret Leakage Attack

An adversary may obtain the temporary (ephemeral) secrets (e.g., random variable) of any session from a compromised mobile device if those are not deleted properly. In this kind of attacks, with the mentioned information, an attacker can initiate an ephemeral secret leakage attack. As per our proposed scheme, our session key is generated as follows:
$SK_{MU_i,CT} = H(ID_i \ ||ID_{CT} \ ||A_{U_{i}CT} \ ||\Omega_{u_i} \ ||\mathcal{P}_2 \ ||\mathcal{T}_{u_i} \ ||\mathcal{T}_{ct})$ where $A_{U_{i}CT} = V'_{CT_{U_i}} \oplus (H(ID_i||$ $H(PW_i||\ n' \ ||\eta_i)))$. Ω_{u_i} is a 128-bit random number there. With this single random number, an attacker cannot regenerate the session key $SK_{MU_i,CT}$, as it requires some other credentials, such as ID_i, ID_{CT}, PW_i etc. Thus, our scheme can defend the ephemeral secret leakage attack.

```
(* ————— functions and equations ————— *)
fun H(bitstring):bitstring. (* hash function *)
fun Generation(bitstring):bitstring. (* Fuzzy extractor function *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring. (* string concatenation *)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(* ————— aims for verification ————— *)
query attacker(SKmuct).
query attacker(SKctmu).
query id:bitstring; inj-event(UserAuth(id)) ==⟩ inj-event(UserStart(id)).
(* ————— event ————— *)
event UserStart(bitstring). (* User starts authentication *)
event UserAuth(bitstring). (* User is authenticated *)
(*—event CTReg(bitstring) —- Cellular Tower starts Registration *)
```

Figure 4.12: Code for declarations of functions, equations, queries and events

The User Impersonation Attack

In the user impersonation attack, an adversary pretends itself as an authorized user to the cellular tower. So, for login, an adversary needs the credentials value of ID_i, PW_i, $\mathcal{B}'_i$. As it is already discussed that in our proposed scheme, these credentials are not sent directly through

```
(*————user starts———*)
let MUser=
new n:bitstring;
new lamda:bitstring;
let meu = Generation(Bi) in
let MPWB = H(con(IDmu,H(con(PW,con(meu,n))))) in
out(sch,(IDmu,xor(MPWB,lamda)));
in(sch,(TIDMU:bitstring,VctU:bitstring,cRID:bitstring));
let P1 = xor(H(con(PW,meu)),n) in
let P2 = H(con(IDmu,con(PW,con(meu,n)))) in
let VctU1 = xor(VctU,lamda) in
let RIDi = xor(TIDMU,H(con(IDmu,VctU1))) in
let RID1 = xor(cRID,H(con(meu,n))) in
```

```
!( event UserStart(IDmu);
let n1 = xor(P1,H(con(PW,meu))) in
let P21 = H(con(IDmu,con(PW,con(meu,n1)))) in
if P2 = P21 then
new Omega:bitstring;
new TUi:bitstring; (*—Current Timestamp—*)
let MPWB1 = H(con(IDmu,H(con(PW,con(meu,n1))))) in
let mAuct = xor(VctU1,MPWB1) in
let M1 = xor(mAuct,xor(Omega,xor(TUi,H(IDCT)))) in
let TID1 = xor(RIDi,H(con(IDmu,VctU1))) in
let TIDi = xor(TID1,H(con(IDCT,TUi))) in
let H1 = H(con(IDmu,con(M1,con(Omega,TUi)))) in
out(pch,(TIDi,M1,H1,TUi));
in(pch,(M2:bitstring,xH3:bitstring,xTct:bitstring)); (*—received after authentication—*)
let P22 = xor(M2,xor(xTct,xor(IDmu,mAuct))) in
let SKmuct =
H(con(IDmu,con(IDCT,con(mAuct,con(Omega,con(P22,con(TUi,xTct))))))) in
let H4 = H(con(IDmu,con(Omega,con(P22,con(TUi,con(xTct, SKmuct)))))) in
if H4 = xH3 then
0 ).
```

Figure 4.13: Code in ProVerif for the process of MU_i, the i^{th} mobile user

the public channel or saved in the device memory, or it is computationally infeasible to obtain them from the easily available information. If an adversary wants to send a login message $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to CT, it needs to compute TID_i^* $(= TID_i \oplus H(ID_{CT}|| \mathcal{T}_{u_i})$ and $\mathcal{M}_1 = A_{U_iCT} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$. After receiving the login request, with the help of timestamp value $\mathcal{T}_{u_i}$ and the random variable A_{U_iCT}, CT can determine if the received message Msg_1 is original or replayed. Therefore, with invalid ID_i, PW_i, and $\mathcal{B}_i'$, it is not possible to generate or modify Msg_1. Thus, our proposed scheme can defend the user impersonation attack.

The Server Impersonation Attack

In the server impersonation attack, an attacker may pretend itself as a genuine server. In our proposed scheme, after receiving valid login request message, a cellular tower CT replies back with an authorization message $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i. For calculating $\mathcal{M}_2 (= B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i)$ and hash value $H_3 (= H(ID_i \ ||\mathcal{P}_1 \ ||\Omega_{ct} \ ||\mathcal{T}_{u_i} \ ||\mathcal{T}_{ct} \ ||SK_{CT,MU_i}))$, an attacker needs the secret key, X_Γ of the cellular tower and the random number $x_{CT_{U_i}}$ as $B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \ ||X_\Gamma)$. Hence, our proposed scheme can resist the server impersonation attack.

4.5 Formal Security Verification Using ProVerif

In this section, I present the formal security verification of the proposed scheme using based ProVerif simulation tool [1]. This tool is based on applied pi calculus and can be used to verify whether an attacker can attack the session key [156]. I have modelled the proposed scheme in ProVerif and corresponding the source codes have been presented in Figure 4.11, Figure 4.13, Figure 4.14, and Figure 4.15.

In Figure 4.11 and 4.12 the code for channel declarations is presented along with the definition of constants, free variables,functions, equations, queries and events, which are needed to model the proposed scheme. Figure 4.13 depicts the ProVerif code for mobile user MU registration, login, authentication and key-establishment process with CT. Cellular tower registration process (CTReg) and authentication process (CTAuth) have been presented as a parallel composition in Figure 4.14.

Finally, I execute the codes given in the previous three Figures in the latest version (1.93) of ProVerif simulation tool. The results of session key secrecy (from the user as well as cellular

```
let CTReg =
in(sch,(xIDmu:bitstring,uMPWB:bitstring));
new TIDMU:bitstring;
let RID = H(con(IDCT,Xtau)) in
let Auct = H(con(H(xor(xIDmu,XctU)),Xtau)) in
let VctU = xor(Auct,uMPWB) in
out(sch,(TIDMU,VctU,RID)).
```

```
(*————————————————-*)
let CTAuth =
in(pch,(mTIDi:bitstring,mM1:bitstring,mH1:bitstring,mTUi:bitstring));
new TSi:bitstring; (*—Received Timestamp—*)
let TID2 = xor(mTIDi,H(con(IDCT,mTUi))) in
let Bctu = H(con(H(xor(IDmu,XctU)),Xtau)) in
let P1 = xor(mM1,xor(mTUi,xor(H(IDCT),Bctu))) in
let H2 = H(con(IDmu,con(mM1,con(P1,mTUi)))) in
if H2 = mH1 then
event UserAuth(IDmu);
new OmegaCT:bitstring;
new Tct:bitstring;
let M2 = xor(Bctu,xor(OmegaCT,xor(Tct,IDmu))) in
let SKctmu =H(con(IDmu,con(IDCT,con(Bctu,con(P1,con(OmegaCT,con(mTUi,Tct)))))))) in
let H3 = H(con(IDmu,con(P1,con(OmegaCT,con(mTUi,con(Tct,SKctmu)))))) in
out(pch,(M2,H3,Tct)).
let CT = CTReg | CTAuth.
process !MUser | !CT
```

Figure 4.14: Code in ProVerif for the process of *CT*

– Query not attacker(SKmuct[])
Completing...
Starting query not attacker(SKmuct[])
RESULT not attacker(SKmuct[]) is true.
– Query not attacker(SKctmu[])
Completing...
Starting query not attacker(SKctmu[])
RESULT not attacker(SKctmu[]) is true.
– Query inj-event(UserAuth(id)) ==⟩ inj-event(UserStart(id))
Completing...
200 rules inserted. The rule base contains 200 rules. 10 rules in the queue.
Starting query inj-event(UserAuth(id)) ==⟩ inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==⟩ inj-event(UserStart(id)) is true.

Figure 4.15: Results of the ProVerif simulation and their analysis

tower) and authentication are presented in Figure 4.15. The following observations can be drawn from the results.

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

From the result set mentioned above, It can be concluded that the proposed scheme passes the required security verification.

4.6 Performance Analysis

In this section, I present the computation and communication cost of our proposed scheme. It is to be noted that the proposed scheme avoids cryptographic operations such as bilinear pairing, elliptic curve point multiplication operation etc., as they incur high computation overhead.

Table 4.3: Various notations used and their time complexity

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

Table 4.4: Computation cost of the proposed scheme

Phase/Entity	Mobile User (MU_i)	Cellular Tower (CT)	Location Server ($LSSNS_j$)	Total execution time (in ms)
Mobile User Login and Authentication	$10{*}T_H + 10{*}T_X + T_{FE}$	$7{*}T_H + 8{*}T_X$	–	71.58
Location Server Login and Authentication	–	$6{*}T_X + T_{CH}$	$7{*}T_H + 8{*}T_X + T_{CH}$	45.54
Distance Threshold Registration	$T_H + 2{*}T_{sym}$	$2{*}T_H + 4{*}T_{sym}$	$2{*}T_{sym}$	71.10
User Location Update	$T_H + 2{*}T_{sym}$	$2{*}T_H + 4{*}T_{sym}$	$2{*}T_{sym}$	72.10
Friends' Location Query	$T_H + 2{*}T_{sym}$	$3{*}T_H + 4{*}T_{sym}$	$2{*}T_H + 2{*}T_{sym}$	72.6

4.6.1 Computation cost analysis

Table 4.3 shows various cryptographic operations, corresponding notations and their execution time on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, as performed in [161], [100]. Due to the fuzzy extractor $Rep(\cdot)$ function for extracting the biometric key α_i, I require $T_{FE} \approx T_M$ [84]. Symmetric encryption/decryption has been given for a AES-128 symmetric cryptosystem. The mobile user registration and $LSSNS$ Registration mechanism is a one-time process. As a result, I have not considered the computation cost of the registration

Table 4.5: Communication cost of the proposed scheme

Phase	Entity	Communicated message	Size (bits)
Mobile user login and authentication phase	MU_i side	$\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$	512
	CT side	$\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$	352
Location server login and authentication phase	$LSSNS_j$ side	$\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}$	640
	CT side	$\{\mathcal{R}_4, M_{CT}, T_{CT}\}$	352
Distance threshold registration phase	MU_i side	$\langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|\| \mathcal{D}_{f_{u_i}}\|\| RN_{u_i}\|\| TS_{u_i}\|\| \mathcal{R}_{flag}=1), H(ID_i\|\| RN_{u_i}\|\| TS_{u_i}), TS_{u_i}\rangle$	320
	CT to $LSSNS_j$	$\langle E_{\mathcal{SK}_{CT,S_j}}(ID_i\|\| \mathcal{D}_{f_{u_i}}\|\| RN_{u_i}\|\| RN_{ct}\|\| TS_{ct}\|\| \mathcal{R}_{flag}=1), H(ID_i\|\| RN_{ct}\|\| TS_{ct}), TS_{ct}\rangle$	320
	CT to MU_i	$E_{\mathcal{SK}_{CT,MU_i}}(ID_i\|\|RN_{u_i}\|\|`ok')$	128
	$LSSNS_j$ side	$\langle E_{\mathcal{SK}_{CT,S_j}}(ID_i\|\|RN_{u_i}\|\|RN_{ct}\|\|`ok')\rangle$	128
User location update phase	MU_i side	$\langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|\|x_{u_i}\|\|y_{u_i}\|\|E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},y_{u_i})\|\|RN_{u_i}\|\|TS_{u_i}), H_1\rangle$	288
	CT to $LSSNS_j$	$\{ID_{S_j}, E_{\mathcal{SK}_{CT,S_j}}(Msg_2\|\| RN_{ct}\|\| TS_{ct}), H(LSSNS_j\|\| Msg_2\|\| RN_{ct}), TS_{ct}\}$	480
	CT to MU_i	$E_{\mathcal{SK}_{CT,MU_i}}(`ok')$	128
	$LSSNS_j$ side	$E_{\mathcal{SK}_{S_j,CT}}(ID_i\|\|LSSNS_j\|\|RN_{ct}\|\|`ok')$	128
Friends' Location Query	MU_i side	$\{E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|\| \mathcal{F}\|\| qf_{u_i}\|\|TS_{u_i} \oplus RN_{u_i}), H(ID_i\|\| \mathcal{F}\|\| qf_{u_i}\|\|RN_{u_i}), TS_{u_i}\}$	320
	CT to $LSSNS_j$	$\{E_{\mathcal{SK}_{CT,S_j}}(ID_i\|\| \mathcal{F}\|\| qf_{u_i}\|\|TS_{ct} \oplus RN_{ct}^{new}), H(ID_i\|\| \mathcal{F}\|\| qf_{u_i}\|\|RN_{ct}^{new}), TS_{ct}\}$	320
	CT to MU_i	$(\{ID_{U_i}, E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_U, y_U)\}_{i=1\ldots k'})$	288
	$LSSNS_j$ side	$\langle E_{\mathcal{SK}_{S_j,CT}}(\{\mathcal{F}_i\}_{i=1\cdots k}, Index_{u_i}, RN_{ct}^{new}), H((\{\mathcal{F}_i\}_{i=1\cdot k})\|\| RN_{ct}^{new}\|\| TS_{S_j}), TS_{S_j}\rangle$	320

phases.

In Table 4.4, I have tabulated the computational overhead for the main three entities of the scheme MU_i, CT and $LSSNS_j$. For MU_i, during login phase overhead is $10^*T_H + 10^*T_X + T_{FE}$. Since bitwise XOR operation, T_X time is negligible, the overhead will be $10 * T_H + T_{FE}$. For the authentication, required overhead of CT will be $7 * T_H + 8 * T_X \approx 7 * T_H$. Hence, overall computation cost of mobile user login and authentication phase is $17 * T_H + T_{FE} = 17{*}0.5{+}1{*}63.08 = 71.58$ms. Following the same procedure, I have calculated the computation cost and the exact execution time of all other remaining phases of the proposed scheme and tabulate them in Table 4.4.

4.6.2 Communication cost analysis

In order to calculate the overall communication overhead of my proposed scheme, I have assumed standard bit sizes of various parameters and cryptographic function outputs. As an example, the bit size of used identity, random numbers and timestamp are 160, 128 and

32 bits respectively. The size of output of hash function $H(\cdot)$ is 160 bits, (if I use SHA-1 hash function [63]) and output of symmetric encryption/decryption (for example, Advanced Encryption Standard or AES-128 [64]) is 128 bits and the prime number is 160 bits. For mobile user login and authentication in our proposed scheme, two message communications are required. In step ULA2 of Section 4.3.2, CT receives the login request message from mobile user MU_i. In step ULA4, CT sends one authentication response message to the MU_i. The communication cost for transmission of the MU_i login message $\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ requires $(160+160+160+32)$=512 bits and authentication response message $\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ requires (160+160+32)= 352 bits. In the same fashion, I calculate the communication cost of messages communicated in various other phases of the proposed scheme. Table 4.5 shows the detailed communication cost for different phases.

Table 4.6: Storage analysis of the proposed scheme.

Entity	Parameters	Total size
MU_i	μ_i, P_i^1, P_i^2, $V'_{CT_{U_i}}$, RID_i, RID'_Γ, $SK_{MU_i,CT}$	1088 bit
CT	X_Γ, ID_{CT}, ID_i, TID_i, $x_{CT_{U_i}}$, ID_{S_j}, SN_j, r, SK_{CT,MU_i}, SK_{CT,S_j})	3296 bit
$LSSNS_j$	E_1, $T_{X_\Gamma}(K_j)$, E_2, f_{S_j}, $SK_{S_j,CT}$	800 bit

4.6.3 Storage overhead analysis

I have three different entities in my scheme - mobile device (MU_i), cellular tower(CT) and location sharing social network server($LSSNS_j$). I have calculated the storage requirement for each of them separately. The lengths of some important parameters that are needs to calculate the storage space are followings:
Device identity or serial number :: 160 bit
Output of a secured one way hash function $H(\cdot)$ :: 160 bit
Session key :: 160 bit
One random number, r :: 128 bit
Master secret key, X_Γ:: 1024-bit
Secret key, $x_{CT_{U_i}}$:: 1024-bit

Fuzzy Extractor, μ_i :: 128 bit

According to our proposed scheme, a mobile device MU_i mandatory needs to store μ_i, P_i^1, P_i^2, $V'_{CT_{U_i}}$, RID_i, RID'_Γ, $SK_{MU_i,CT}$. Hence, the required storage space of MU_i is = 128 + 160 + 160 + 160 + 160 + 160 +160 = 1088 bit. A cellular tower, CT needs minimum { X_Γ + ID_{CT} + ID_i + TID_i + $x_{CT_{U_i}}$ + ID_{S_j} + SN_j + r + SK_{CT,MU_i} + SK_{CT,S_j}) = 1024 +160 + 160+ 160+ 1024+ 160+ 160 + 128+160 + 160} = 3296 bit storage space to complete its processing. $LSSNS_j$ requires { E_1 + $T_{X_\Gamma}(K_j)$ + E_2 + f_{S_j} + $SK_{S_j,CT}$} = 160 +160+ 160 +160 +160 = 800 bit. Table 4.6 shows the storage analysis of the proposed scheme.

Table 4.7: Security and functionality comparison

Security attribute /Scheme	C. C. Lee *et al.* [109]	X. Li *et al.* [118]	Tsai-Lo [179]	Irshad *et al.* [89]	H. Wang *et al.* [184]	Our
Stolen smart card attack	✓	✓	X	X	✓	✓
Supports three-factor authentication	X	X	X	X	✓	✓
Off-line password guessing attack	✓	✓	X	✓	✓	✓
On-line password guessing attack	✓	✓	✓	✓	✓	✓
Strong replay attack	✓	✓	✓	✓	✓	✓
Privileged insider attack	✓	✓	X	✓	✓	✓
User impersonation attack	✓	✓	✓	✓	✓	✓
Server impersonation attack	X	✓	✓	✓	✓	✓
Denial of service attack	✓	✓	✓	✓	✓	✓
Known session key secrecy	✓	✓	✓	✓	✓	✓
User anonymity provision	✓	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓	✓	✓
Session key security	✓	✓	X	✓	✓	✓
Session key recovery attack	✓	✓	✓	X	✓	✓
Login phase efficiency	X	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓
Supports Location-Sharing	X	X	X	X	X	✓
Supports Friends' Locations Query	X	X	X	X	X	✓
Formal security analysis	X	✓	X	X	✓	✓
Simulation using AVISPA/ProVerif	X	X	X	X	X	✓

Note: X: does not support a particular feature; ✓: supports a particular feature.

4.7 Performance and Comparative Study

In this section, I have presented a comparative study of my proposed scheme with some recent chaotic-map based user authentication schemes under multi-server environment, such as schemes proposed by C. C. Lee *et al.* [109], X. Li *et al.* [118], Tsai-Lo [179], Irshad *et al.* [89] and H. Wang *et al.* [184]. The comparative study includes detail analysis and comparison in terms of security and functionality features, computation overheads and communication overheads.

In Table 4.7, I have tabulated an overall security and functionality features comparison among our proposed scheme and other related authentication and key-establishment schemes. It is seen that a large number of the recent schemes do not support three-factor authentication, as they do not include user biometrics [36]. The tabulation result reveals that the existing schemes suffer from various security attacks like stolen smart card attack [179], [89], server impersonation attack [109], session key recovery attack [89] and login phase inefficiency [109]. Moreover, it is observed that these chaotic-map based authentication schemes can not support proper location-sharing and friends' locations query feature. It is clear from Table 4.7 that the proposed scheme overcomes such security and functionality weaknesses of the existing schemes.

Table 4.8: Comparison of computational costs among related schemes.

Phase	Entity	C. C. Lee *et al.* [109]	X. Li *et al.* [118]	Tsai-Lo [179]	Irshad *et al.* [89]	H. Wang *et al.* [184]	Our
Mobile user and server login and authentication phase	Mobile user	$5T_H + 3T_{CH}$ $\approx$ 65.56 ms	$7T_H + 3T_{CH}$ $\approx$ 66.56 ms	$8T_H + 2T_{CH}$ $\approx$ 46.04 ms	$11T_H + 3T_{CH}$ $\approx$ 68.56 ms	$11T_H + 2T_{CH}$ $\approx$ 47.54 ms	$10T_H + T_{FE}$ $\approx$ 68.08 ms
	Server	$6T_H + 3T_{CH}$ $\approx$ 66.06 ms	$4T_H + 2T_{CH}$ $\approx$ 44.04 ms	$5T_H + 2T_{CH}$ $\approx$ 44.54 ms	$7T_H + 2T_{CH}$ $\approx$ 45.54 ms	$8T_H + 2T_{CH}$ $\approx$ 46.04 ms	$7T_H + T_{CH}$ $\approx$ 24.52 ms
	RC/ *CT*	–	$8T_H + T_{CH}$ $\approx$ 25.02 ms	$10T_H$ $\approx$ 5 ms	$11T_H + T_{CH}$ $\approx$ 26.52 ms	$8T_H + 3T_{CH}$ $+2T_{Sym}$ $\approx$ 84.46 ms	$14T_H + T_{CH}$ $\approx$ 24.52 ms
Total computation cost		$\approx$ 131.62 ms	$\approx$ 135.62 ms	$\approx$ 95.58 ms	$\approx$ 140.62 ms	$\approx$ 178.04 ms	$\approx$ 117.122 ms

In Table 4.8, I tabulate and compare the computation overheads of the proposed scheme with the relevant schemes [109], [118], [179], [89], [184]. The mobile user registration phase and the location sharing social network server registration phase are an one-time process

Table 4.9: Comparison of communication costs

Scheme	Communication rounds	No. of bits
C. C. Lee *et al.* [109]	3	1088
X. Li *et al.* [118]	5	2592
Tsai-Lo [179]	5	2560
Irshad *et al.* [89]	5	3072
H. Wang *et al.* [184]	5	3200
Our	4	1856

only. Hence, for calculation as well as comparison of communication cost, I consider only user and server login, authentication and key-establishment phases for the proposed and related schemes. Table 4.3 shows various cryptographic operations, corresponding notations and their execution time on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, as performed in [161], [100]. For all the given schemes, I separately tabulated computation for the user, server and the registration center or the cellular tower. Also, in Table 4.8, I mention and compare total computation cost for each relevant scheme.

It is observed that the total computation cost of the proposed scheme is $\approx$ 117.122 ms only, whereas computation cost of C. C. Lee *et al.*'s scheme [109] is $\approx$ 131.62 ms, X. Li *et al.*'s scheme [118] is $\approx$ 135.62 ms, Tsai-Lo's scheme [179] is $\approx$ 95.58 ms, Irshad *et al.*'s scheme [89] is $\approx$ 140.62 ms and H. Wang *et al.*'s scheme [184] is $\approx$ 178.04 ms. It is to be noted that, except Tsai-Lo's scheme, I have the lowest computation cost. The reason behind such low computation cost of our proposed schemes is that, I use only two chaotic map operations for authentication and key-establishment purpose, which is the minimum among other related existing schemes.

Table 4.9 shows and compares message communication rounds and communication cost (in bits) of the proposed scheme with related schemes [109], [118], [179], [89], [184]. Since the user and server registration phasesare executed only once, I consider only user and server login, authentication & key-establishment phases for calculation of communication cost for the proposed scheme and other schemes. In our proposed scheme, mOSN user and location server login and authentication phase needs 864 bits and 992 bits of message communication respectively, with a total communication cost of 1856 bits. From Table 4.9, it is clear that, compared to all related scheme, except C. C. Lee *et al.*'s scheme [109], the proposed scheme

has the minimum communication cost. Unfortunately, as shown in Table 4.7, C. C. Lee *et al.*'s scheme is vulnerable to some serious security attacks. Overall, the proposed scheme is both efficient and provides much greater security and functionality features for the smart devices as compared to all existing compared schemes.

4.8 Summary

This chapter presents an efficient location sharing scheme for mOSNs and shows the ability to resist various active and passive security attacks that are present in the existing schemes. The proposed scheme integrates LBS and SNS into a set of single entity servers, thereby reducing their internal communication overhead. Our location sharing scheme for mOSNs shows both efficiency and flexibility in location update, sharing, and query of social friends and social strangers. Formal security verification, authentication proof and simulation results prove the security strength of the proposed scheme.

Chapter 5

DDoS Attack Resisting Authentication Protocol for mOSN Applications

As discussed in the literature, in distributed denial-of-service (DDoS) attacks, an adversary aims to overwhelm the normal traffic of a targeted server with a flood of fake login messages so that the associated Internet service or website turns inoperable. All popular OSN platforms such as Facebook, Twitter, YouTube, Google's Blogger, etc. are facing DDoS attacks regularly. The existing DDoS resisting schemes of mOSNs suffer from several security drawbacks and most of them are vulnerable to DDoS attacks, which are occurred due to the loss of synchronization between the participants. To rebuild synchronization between the participants, a protocol may need to compromise the un-linkability property. Therefore, the design of an efficient authentication protocol for resistance to DDoS attacks is essential.

In this chapter, I have aimed to propose an authentication protocol to resist DDoS in mobile-based OSN applications. I have provided here a multi-faceted solution towards the remedy of DDoS attacks in the OSN environment. According to our proposed scheme, a user and an online social network (OSN) server at the beginning, separately establish a shared symmetric session key through a cellular tower. If an OSN server receives more than one login request from a specific user, after a certain threshold, the scheme discards further user login attempts and blocks that user treating as an adversary who intends to overload the network server. I also use the pre-loaded shadow identity and emergency key pairs, and a key-refilling strategy that rebuilds the essential synchronization between a blocked naive user and the OSN server. This technique also restores the intended un-linkability property of the protocol.

5.1 Research Contributions

The main research contributions of this chapter are summarized below:

- I propose a secure and lightweight DDoS attack resisting authentication scheme (*PRDoS*) for mOSN applications. *PRDoS* blocks DDoS attackers who intend to overwhelm the mOSN server by using redundant login requests. Through the emergency key refilling phase, resynchronization can be built between user and OSN server using preloaded shadow identity and emergency key pairs.

- The proposed scheme is very lightweight and making it an easy-to-implement choice for any practical scenario for all battery limited user devices. The computation and communication costs analysis shows that it outperforms other existing competing schemes.

- Using NS3 simulation, I study the impact of DDoS attacker on network throughput and network delay. The simulation results show that the proposed authentication scheme does not allow the DDoS attacker to keep negative impact on both network throughput and delay.

- I further validate and compare the proposed scheme against state-of-the-art solutions using real attack and benign datasets, like Canadian Institute for Cybersecurity (CIC) DoS dataset, 2017. I have used the machine learning algorithms, like KNN, Gaussian Naive Bayes and Multilayer Perceptron (MLP) to demonstrate the performance of the proposed scheme in a practical attack detection scenario.

5.2 Threat Model

According to the system model, two entities are communicating through an insecure public channel. As a consequence, an adversary may eavesdrop on the communicated message and may modify, replay or delete the content of the message. In this chapter, I have considered the Dolev-Yao threat model (DY model) [55]. Besides launching several types of attacks, an adversary can extract the users' sensitive information by stealing and executing power analysis of the mobile device.

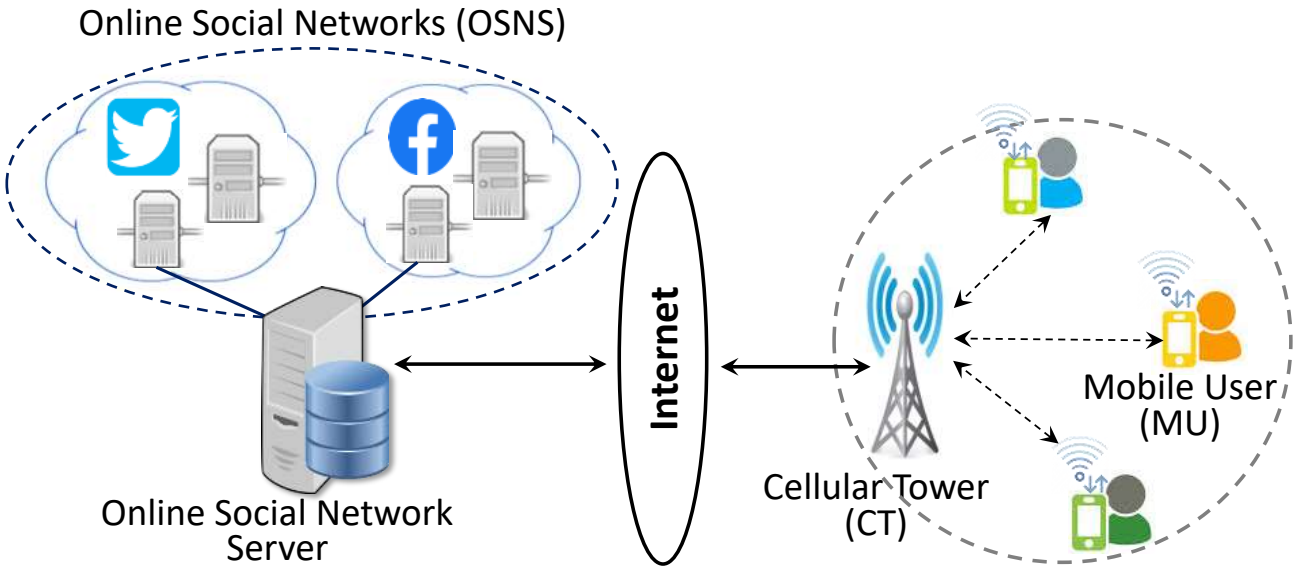

Figure 5.1: Simple architecture of the proposed scheme

5.3 The Proposed Scheme

In this section, I describe various phases of the proposed DDoS attacks resisting user authentication protocol for mobile-based OSN Applications ($PRDoS$). The proposed scheme is composed of five phases, namely, 1) mOSN user and server registration, 2) mOSN and server login, 3) authentication and key establishment, 4) DDoS attack remedy, and 5) emergency key-refilling phase.

The first three phases are used to set up the initial authentication process between OSN user and server. The next phase, i.e., the DDoS attack remedy phase protects the OSN server from DDoS attackers. Here, if an adversary eavesdrops or generates a login message Msg_u and sends multiple copies of it to overload the server SNS_j, the scheme will block that sender and resist DoS attacks in the mOSN environment. Now, if a registered genuine user MU_i forgets its own login credential and makes a redundant login attempt, then, like a DDoS attacker, MU_i will be blocked too. In this case, to be able to login again, that blocked MU_i makes one-time use of preloaded emergency numbers and shadow identity pair of (en_k, sid_k) which are stored in list L_{em_i}. With this, from the blocking stage, MU_i re-synchronizes again with the system. In the emergency key-refilling phase, I have described that process.

The proposed scheme requires three main entities of any social network architecture:

- **Online social network server** (SNS): It provides a social networking interface and stores users' data. One of its important tasks is to authenticate the registered users who

can have an access to online social networking applications.

- **Mobile OSN user (MU)**: User registers himself/herself as a genuine user for the first time, and then login to the mOSN server to access the social networking interfaces.
- **Cellular tower (CT)**: It works as a coordinator between MU and SNS. All the messages communicated between MU and SNS are received, processed, and forwarded via CT. I assume that, it is a trusted entity.

Table 5.1: Symbols and notations used in the proposed scheme.

Symbol	Description
MU_i	i^{th} mOSN user
ID_{mu_i}	The identity of the i^{th} user
SNS_j	j^{th} Online Social Network Server
ID_{ss}	The identity of the SNS_j
CT	Cellular Tower
PW_{mu_i}	The login password of MU_i
TID_{mu_i}	Temporary identity of MU_i
PID_{mu_i}	The pseudo-identity of MU_i
S_j	1024-bit master secret key of SNS_j
Z_{us}	1024-bit secret number
R_u	128-bit random number chosen by MU_i
SN_u	Serial number of MU_i's mobile device
$SKEY$	Symmetric key
$E_{sk}(\cdot)/D_{sk}(\cdot)$	Encryption/decryption using key k
$H(\cdot)$	Cryptographic hash function
B_{mu_i}	User biometric of MU_i
$\oplus, \|\|$	Bitwise XOR, Concatenation operations
$P \xrightarrow{Msg} Q$	P sends Msg to Q via public channel
$\triangle T$	Maximum transmission delay
T_u	Timestamp generated by MU_i

Figure 5.1 is used to describe the overall architecture of the system. At the very beginning,

both mobile OSN user (MU) and online social network server (SNS) need to register to the cellular tower (CT). This is a one-time process and is executed through a secure channel. Next, in a normal scenario, MU will send a login message to SNS via CT through a public channel. CT will verify whether the MU and SNS both are registered entities or not. After the successful verification, CT forwards that login message to SNS. After receiving the login message, SNS verifies both authenticity and integrity of MU and its message. A mutually confirmed session key will be established after successful authentication and all the messages of that session will be encrypted with the session key.

All the notations used in this scheme and their descriptions are listed in Table 5.1.

5.3.1 The registration phase

To access OSN services, a new mobile user MU_i needs to register himself/herself to the social network server (SNS_j). This registration is done through the cellular tower (CT). The OSN server SNS_j also registers itself to CT. These are independent, one-time processes and are executed through a secure private channel.

OSN server registration phase

When a new OSN server SNS_j has to join the network, it needs to register to CT. The following steps will get executed at the time of server registration:

- **Step SR1:** SNS_j selects an unique identity ID_{ss} and password PW_{ss} for its own. SNS_j computes hashed password, $H(PW_{ss})$.
- **Step SR2:** Next, SNS_j sends a message $Mreg_s = \{H(PW_{ss}) \oplus ID_{ss}, H(PW_{ss})\}$ for its registration to CT through a secure channel.
- **Step SR3:** When CT receives $Mreg_s$, it extracts the unique id $ID_{ss} = (H(PW_{ss}) \oplus ID_{ss}) \oplus H(PW_{ss})$ from that message. CT provides a unique and 1024-bit number S_j as the master secret key of SNS_j.
  CT calculates the pseudo-identity $PID_{ss} = H(ID_{ss}||S_j)$ for SNS_j. CT provides the sensible credentials $\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us})\}$ of all OSN users securely in the database of SNS_j.
- **Step SR4:** SNS_j saves $\{ID_{ss}, S_j\}$ into its own database.
- **Step SR5:** CT also saves $\{ID_{ss}, S_j\}$ into its database.



Mobile User (MU_i)	Cellular Tower (CT)	OSN Server (SNS_j)
Input ID_{mu_i}, PW_{mu_i} and B_{mu_i}. Compute $(\gamma_i, \delta_i) = Gen(B_{mu_i})$, Select random numbers a and b. $BPW_i = (H(ID_{mu_i}\|\| H(PW_{mu_i}\|\| \gamma_i \|\|a)))$. $\xrightarrow[\textbf{Secure channel}]{\{ID_{mu_i}, BPW_i \oplus b\}}$		
	Selects random number Z_{us} for each (MU_i, SNS_j) pair. Computes $B_{us} = H(H(ID_{mu_i} \oplus Z_{us}) \|\|S_j)$, $E_{us} = B_{us} \oplus BPW_i$, Select temporary identity TID_{mu_i} for MU_i. Fetches the serial number SN_u of MU_i. $\xrightarrow[\textbf{Secure channel}]{\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us})\}}$	
		Generates a list $L_{em_i} = \{(en_1, sid_1),$ $(en_2, sid_2), (en_3, sid_3), \ldots, (en_n, sid_n)\}$. Save $\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us}), L_{em_i}\}$ in its memory. $\xleftarrow[\textbf{Secure channel}]{\{L_{em_i}\}}$
	For all SNS_js, store $\{TID_{mu_i}, (ID_{ss_j}, PID_{ss_j}, E_{us_j}) \| 1 \le j \le n\}$ and L_{em_i} in mobile device memory of MU_i. Stores (ID_{mu_i}, SN_u) into its own memory. $\xleftarrow[\textbf{Secure channel}]{\{Mobile\ Device\}}$	
Compute $F_i^1 = H(PW_{mu_i}\|\| \gamma_i) \oplus a$, $F_i^2 = H(ID_{mu_i} \|\|PW_{mu_i}\|\| \gamma_i \|\|a)$, $E'_{us} = E_{us} \oplus b$ $= B_{us} \oplus H(ID_{mu_i}\|\|H(PW_{mu_i}\|\|\gamma_i\|\|a))$. For all OSN servers SNS_j, MU_i compute: $PID_{mu_i} = TID_{mu_i} \oplus H(ID_{mu_i} \|\|E'_{us})$, $PID'_{ss} = PID_{ss} \oplus H(\gamma_i\|\|a)$. Store $\{\delta_i, F_i^1, F_i^2, E'_{us}, PID_{mu_i}, PID'_{ss}\}$ into the mobile device memory Delete E_{us}s, TID_{mu_i} and PID_{ss}s from its memory.		

Figure 5.2: Registration phases for mobile user of mOSN

Mobile user registration phase

When a new mobile user MU_i wishes to join in a specific online social network, the following steps will be required to execute for MU_i registration to the CT and SNS_j. The steps require

to communicate in this phase are presented in Figure 5.2.

- **Step MU1:** MU_i selects an unique identity ID_{mu_i}, password PW_{mu_i}, and biometric B_{mu_i}. Using the fuzzy extractor $(\cdot)$ function MU_i generates $(\gamma_i, \delta_i) = Gen(B_{mu_i})$ and selects two 128-bit numbers a and b randomly.

- **Step MU2:** Next, MU_i produces a biometric embedded password $BPW_i = H(ID_{mu_i}|| H(PW_{mu_i}|| \gamma_i ||a))$ and sends a registration request message $Mreg_u = \{ID_{mu_i}, (BPW_i \oplus b)\}$ to the CT.

- **Step MU3:** When CT receives a message $Mreg_u$ from a new user MU_i, it selects a 1024-bit random number Z_{us} for each SNS_j and MU_i pair. CT also computes some parameters, such as $B_{us} = H(H(ID_{mu_i} \oplus Z_{us}) ||S_j)$, where S_j is the master secret key of the server SNS_j and $E_{us} = B_{us} \oplus BPW_i$. CT uses the process of identity anonymization, it provides a temporary but unique identity TID_{mu_i} to MU_i instead of its original identity ID_{mu_i}.

- **Step MU4:** CT fetches the serial number SN_u from MU_i's mobile device and forwards $\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us})\}$ to SNS_j through a secure channel.

- **Step MU5:** After receiving the registration information from a new user MU_i, SNS_j generates a list of N emergency numbers and shadow identity pairs $L_{em_i} = \{(en_1, sid_1), (en_2, sid_2), (en_3, sid_3), \ldots, (en_n, sid_n)\}$. This list will be required in future for resolving the issue of DoS attacks. SNS_j saves $\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us}), L_{em_i}\}$ into its database and sends L_{em_i} to CT through a secure channel.

- **Step MU6:** When CT receives the list of emergency numbers and shadow identity pairs L_{em_i} from SNS_j, CT saves a list of n servers information $\{TID_{mu_i}, (ID_{ss_j}, E_{us_j}, PID_{ss_j}) \mid 1 \leq j \leq n\}$ and $\{L_{em_i}\}$ in the memory of the mobile device of MU_i securely. CT stores the record of (ID_{mu_i}, SN_u) into its own database.

- **Step MU7:** After receiving the parameter from CT, MU_i calculates the following parameters.

  $F_i^1 = H(PW_{mu_i}|| \gamma_i) \oplus a$,
  $F_i^2 = H(ID_{mu_i} ||PW_{mu_i} ||\gamma_i ||a)$,
  $E'_{us} = E_{us} \oplus b = B_{us} \oplus H(ID_{mu_i}|| H(PW_{mu_i}|| \gamma_i ||a))$,
  $PID_{mu_i} = TID_{mu_i} \oplus H(ID_{mu_i} ||E'_{us})$
  $PID'_{ss} = PID_{ss} \oplus H(\gamma_i||a)$ for all servers.

- **Step MU8:** MU_i stores (δ_i, F_i^1, F_i^2, E'_{us}, PID_{mu_i}, PID'_{ss}), and removes E_{us}, TID_{mu_i} and PID_{ss} such original parameters from the memory of its mobile device.

5.3.2 Mobile user login phase

For accessing OSN services, registered mobile user MU_i needs to login into the server SNS_j. MU_i sends a login request message to SNS_j via CT. Figure 5.3 summarizes the steps required for the login, authentication, and key-establishment process. The following steps are essential to complete the login phase:

- **Step UL1:** MU_i enters its user id ID_{mu_i}, password PW_{mu_i} and bio-metrics B'_{mu_i} at the time of login. From the saved parameter δ_i and the fuzzy extractor reproduction method, MU_i extracts $\gamma_i = Rep(B'_{mu_i}, \delta_i)$. MU_i also regenerates $a = F_i^1 \oplus H(PW_{mu_i} ||\delta_i)$ using the saved parameters F_i^1.

- **Step UL2:** After calculating $H(ID_{mu_i} ||PW_{mu_i} ||\gamma_i ||a)$, MU_i checks if $F_i^2 = H(ID_{mu_i}||PW_{mu_i}||\gamma_i||a)$ is holding or not. Execution terminates in case of verification fails. MU_i computes $BPW_i = H(ID_{mu_i}|| H(PW_{mu_i}|| \gamma_i ||a))$ and using the saved parameter E'_{us} computes $B_{us} = E'_{us} \oplus BPW_i$.

- **Step UL3:** After selecting a 128-bit random number R_u and current timestamp TS_u, MU_i calculates:
  $X_{i_1} = B_{us} \oplus R_u \oplus TS_u \oplus H(ID_{ss})$
  $= H(H(ID_{mu_i} \oplus Z_{us})||S_j) \oplus R_u \oplus TS_u \oplus H(ID_{ss})$,
  $H_{i_1} = H(ID_{mu_i}||X_{i_1}||R_u||TS_u)$,
  $TID_{mu_i} = PID_{mu_i} \oplus H(ID_{mu_i}||E'_{us})$,
  $PID_{ss} = PID'_{ss} \oplus H(\delta_i||a)$,
  $TID^*_{mu_i} = TID_{mu_i} \oplus H(PID_{ss}||TS_u)$.

- **Step UL4:** Next, MU_i sends $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ to CT for login through a public channel.

- **Step UL5:** When CT receives message Msg_u from MU_i, it computes $H(PID_{ss}||TS_u)$, and extracts $TID_{mu_i} = TID^*_{mu_i} \oplus H(PID_{ss}||TS_u)$. CT checks whether MU_i is a genuine user or not by searching the records of its database.

- **Step UL6:** If MU_i is a genuine user, CT forwards Msg_u to SNS_j.

Mobile User (MU_i)	Cellular Tower (CT)	OSN Server (SNS_j)
	Login phase	
Enters ID_{mu_i}, PW_{mu_i} and B'_{mu_i}. Computes $\gamma_i = Rep(B'_{mu_i}, \delta_i)$, $a = F_i^1 \oplus H(PW_{mu_i}\|\|\gamma_i)$. Verifies if $F_i^2 = H(ID_{mu_i}\|\|PW_{mu_i}\|\|\gamma_i\|\|a)$? After successful verification, Computes $BPW_i = H(ID_{mu_i}\|\| H(PW_{mu_i}\|\| \gamma_i \|\|a))$, $B_{us} = E'_{us} \oplus BPW_i$. Generates random number R_u. Computes $X_{u_1} = B_{us} \oplus R_u \oplus TS_u \oplus H(ID_{ss})$, $H_{i_1} = H(ID_{mu_i}\|\|X_{u_1}\|\|R_u\|\|TS_u)$, $TID_{mu_i} = PID_{mu_i} \oplus H(ID_{mu_i}\|\|E'_{us})$, $PID_{ss} = PID'_{ss} \oplus H(\gamma_i\|\|a)$, $TID^*_{mu_i} = TID_{mu_i} \oplus H(PID_{ss}\|\|TS_u)$. $\xrightarrow[\text{public channel}]{Msg_u=\{TID^*_{mu_i}, X_{u_1}, H_{i_1}, TS_u\}}$	Computes $TID_{mu_i} =$ $TID^*_{mu_i} \oplus H(PID_{ss}\|\|TS_u)$. Checks MU_i register user or not If verification succeeded, $\xrightarrow[\text{public channel}]{Msg_u}$	
	Authentication phase	
		Checks if $\|TS^*_u - TS_u\| \leq \triangle T$? Regenerates $PID_{ss} = H(ID_{ss}\|\|S_j)$, $TID_{mu_i} = TID^*_{mu_i} \oplus H(PID_{ss}\|\|TS_u)$. Searches the record $\langle ID_{mu_i}, TID_{mu_i}, Z_{us}\rangle$ from its database. Compute $D_{su} = H(H(ID_{mu_i} \oplus Z_{us})\|\|S_j)$. $P_1 = X_{u_1} \oplus TS_u \oplus H(ID_{ss}) \oplus D_{su} = R_u$, as $B_{us} = D_{su} = H(H(ID_{mu_i} \oplus Z_{us})\|\|S_j)$, $H_{j_1} = H(ID_{mu_i} \|\|X_{u_1} \|\|P_1 \|\|TS_u)$. Verify if $H_{j_1} = H_{i_1}$? After successful verification, generates R_s. Calculates $X_{j_1} = D_{su} \oplus R_s \oplus TS_s \oplus ID_{mu_i}$, $SKEY_{ji} = H(ID_{mu_i}\|\|ID_{ss}\|\|D_{su}\|\|P_1\|\|R_s\|\|TS_u\|\|TS_s)$, $H_{j_2} = H(ID_{mu_i}\|\|P_1\|\|R_s\|\|TS_u\|\|TS_s\|\|SKEY_{ji})$. $\xleftarrow[\text{public channel}]{Msg_s=\{X_{j_1}, H_{j_2}, TS_s\}}$
	Checks SNS_j is registered or not if verification holds $\xleftarrow[\text{public channel}]{Msg_s}$	
Verify if $\|TS^*_s - TS_s\| \leq \triangle T$? After successful verification, Computes $P_2 = X_{j_1} \oplus TS_s \oplus ID_{mu_i} \oplus B_{us} = R_s$, as $B_{us} = D_{su} = H(H(ID_{mu_i} \oplus Z_{us})\|\|S_j)$. $SKEY_{ij} = H(ID_{mu_i}\|\|ID_{ss}\|\|B_{us}\|\|R_u\|\|P_2\|\|TS_u\|\|TS_s)$, $H_{i_2} = H(ID_{mu_i} \|\|R_u \|\|P_2 \|\|TS_u \|\|TS_s \|\|SKEY_{ij})$. Verify if $H_{i_2} = H_{j_2}$? After successful verification, store session key $SKEY_{ij}(= SKEY_{ji})$.		Store session key $SKEY_{ji}(= SKEY_{ij})$.

Figure 5.3: Login and authentication phases of proposed scheme

5.3.3 User authentication and key-establishment phase

After receiving login request message Msg_u from MU_i, SNS_j checks the both authenticity of the user and the validity of the message Msg_u. Upon successful login, SNS_j and MU_i mutually establish shared secret session key for future message communication. This phase involves the following steps:

- **Step SA1:** When SNS_j receives Msg_u from MU_i, it checks the timestamp received with Msg_u and current timestamp. SNS_j discards the login request, if $|TS_u^* - TS_u| \leq \triangle T$ does not hold, where $\triangle T$ is the predefine maximum transmission delay. SNS_j regenerates its pseudo-identity $PID_{ss} = H(ID_{ss}||S_j)$ and extracts $TID_{mu_i} = TID_{mu_i}^* \oplus H(PID_{ss}\ ||TS_u)$.

- **Step SA2:** SNS_j searches its database to find the record $\langle ID_{mu_i}, Z_{us}\rangle$ associated with TID_{mu_i}. Next, SNS_j computes some other parameters with its id ID_{ss} and master key S_j,

  $D_{su} = H(H(ID_{mu_i} \oplus Z_{us})||S_j)$,

  $P_1 = X_{i_1} \oplus TS_u \oplus H(ID_{ss}) \oplus D_{su}$

  $= B_{us} \oplus R_u \oplus TS_u \oplus H(ID_{ss}) \oplus TS_u \oplus H(ID_{ss}) \oplus D_{su}$

  $= R_u$,

  Now, $B_{us} = D_{su} = H(H(ID_{mu_i} \oplus Z_{us})||S_j)$ is confirmed.

- **Step SA3:** Now, with the help of X_{i_1}, P_1, and TS_u , SNS_j computes the hash value $H_{j_1} = H(ID_{mu_i}\ ||X_{i_1}\ ||P_1\ ||TS_u)$ and match it with received hashed value H_{i_1} if $H_{j_1} \stackrel{?}{=} H_{i_1}$ is hold or not. With successful verification, SNS_j accepts MU_i's request and the executes further otherwise execution terminates.

- **Step SA4:** SNS_j stores the record $\langle ID_{mu_i}, R_u, TS_u\rangle$ into the database to resist DoS attacks and strong replay attacks. If an adversary eavesdrops a login request and sends the same message again and again within a pre-define short span of time for launching the denial-of-service attack, after three consecutive login requests from the same user MU_i, SNS_j blocks the user. Execution will stop there and no authentication message will communicate to the user MU_i.

- **Step SA5:** Otherwise, SNS_j selects a random number R_s (128-bit) and using its current timestamp TS_s calculates $X_{j_1} = D_{su} \oplus R_s \oplus TS_s \oplus ID_{mu_i}$. SNS_j also computes the mutually shared session key $SKEY_{ji} = H(ID_{mu_i}\ ||ID_{ss}\ ||D_{su}\ ||P_1\ ||R_s\ ||TS_u\ ||TS_s)$.

This key will be used for message encryption for MU_i in future. Moreover, SNS_j computes a new hash value $H_{j_2} = H(ID_{mu_i} ||P_1 ||R_s ||TS_u ||TS_s ||SKEY_{ji})$.

- **Step SA6:** Finally, SNS_j sends the authentication reply message $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$ to CT via a public channel.

- **Step SA7:** When CT receives a message from SNS_j, first it verify whether the server is a register entity or not. CT discards the message if the verification fails. Otherwise, CT forwards Msg_s to MU_i .

- **Step SA8:** When MU_i receives the response message, it verifies the time delay and checks whether $|TS_s^* - TS_s| \leq \triangle T$ is or not. Execution stops if verification fails, otherwise it proceeds to the next step.

- **Step SA9:** MU_i then computes the following steps:

$$\begin{aligned} P_2 &= X_{j_1} \oplus TS_s \oplus ID_{mu_i} \oplus B_{us} \\ &= D_{su} \oplus R_s \oplus TS_s \oplus ID_{mu_i} \oplus TS_s \oplus ID_{mu_i} \oplus B_{us} \\ &= R_s \end{aligned}$$

as $B_{us} = D_{su} = H(H(ID_{mu_i} \oplus Z_{us})||S_j)$.

With computed variable P_2 and received timestamp value TS_s, MU_i computes mutually shared key $SKEY_{ij} = H(ID_{mu_i} ||ID_{ss} ||B_{us} ||R_u ||P_2 ||TS_u ||TS_s)$ for that session. At the end, by calculating a hash value $H_{i_2} = H(ID_{mu_i} ||R_u ||P_2 ||TS_u ||TS_s ||SKEY_{ij})$, MU_i checks whether $H_{i_2} \stackrel{?}{=} H_{j_2}$ is hold or not.

- **Step SA10:** In successful verification, the computed secret session key $SKEY_{ij}$ $(= SKEY_{ji})$ is mutually verified and confirmed. This key will be used for message transmission in the future of the existing session.

5.3.4 DDoS attack remedy phase

In this phase, the remedy for DoS attacks in the mOSN environments has been discussed. Figure 5.4 will describe the required steps in this phase and the detailed approach is as follows:

- **Step DS1:** If an OSN server SNS_j receives the same login request message from the same user MU_i, three times within a predefined short period T_α, SNS_j blocks the user.

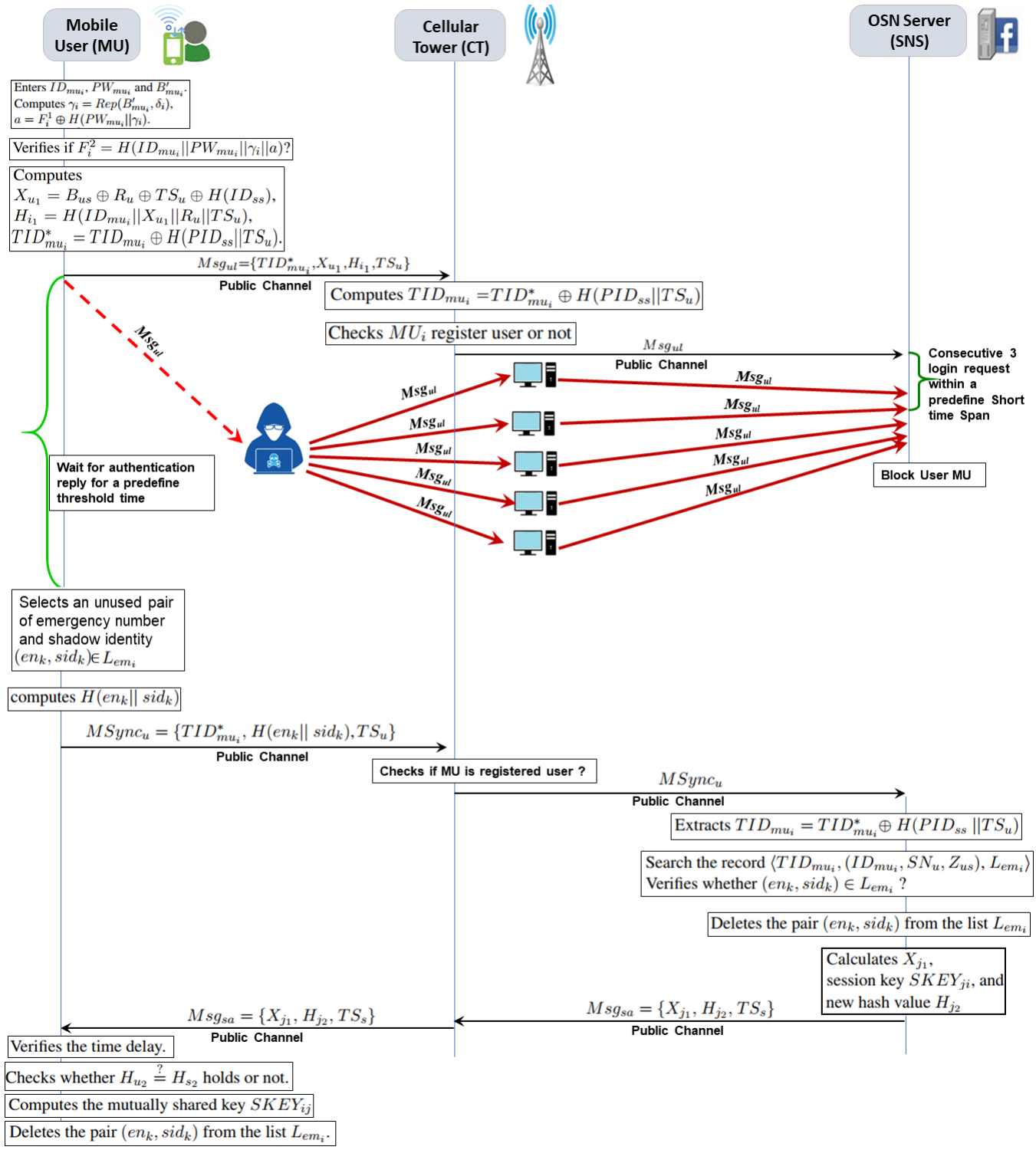

Figure 5.4: DDoS attack resisting phase of the proposed scheme

Execution will stop there and no authentication message will further communicate to the user MU_i.

- Therefore, if an adversary eavesdrops on a login request message and sends it again and again for launching the denial-of-service (DoS) attack, it cannot receive any authentication message for accessing or misuse the OSN services.

- **Step DS2:** There will be a discontinuity of synchronization between MU_i and SNS_j.

- For solving this issue, MU_i waits for the authentication reply for a predefined threshold period T_p.

- **Step DS3:** After that, he/she selects an un-used pair of emergency number and shadow identity (en_k, sid_k) from the emergency key list L_{em_i} and computes $H(en_k||\ sid_k)$.

- **Step DS4:** Next, selecting current timestamp TS_u, and with the other parameters H_{i_1}, MU_i sends the synchronization request $MSync_u = \{TID^*_{mu_i},\ H(en_k||\ sid_k),\ TS_u\}$ to SNS_j via CT.

- **Step DS5:** After receiving the message $MSync_u$ from a registered user MU_i, CT just forwards it to SNS_j.

- When SNS_j receives a synchronization request $MSync_u$, it verifies if $|TS^*_u - TS_u| \leq \triangle T$ or not. On successful verification, it extracts $TID_{mu_i} = TID^*_{mu_i} \oplus H(PID_{ss}\ ||TS_u)$ and search the record $\{TID_{mu_i}, (ID_{mu_i}, SN_u, Z_{us}),\ L_{em_i}\}$ saved for MU_i in its database and verifies whether $(en_k, sid_k) \in L_{em_i}$ or not.

- **Step DS6:** On successful verification, SNS_j deletes the pair (en_k, sid_k) from the list L_{em_i} and selects a random number R_s (128-bit) and using its current timestamp TS_s calculates X_{j_1}, the mutually shared session key $SKEY_{ji}$, and a new hash value H_{j_2} as described in Step **SA5** of Section 5.3.3.

- **Step DS6:** Finally, SNS_j sends the authentication reply message $Msg_s = \{X_{j_1},\ H_{j_2},\ TS_s\}$ to MU_i via CT through a public channel.

- **Step DS7:** When MU_i receives the authentication response message, he or she proceeds further normally as described in **Step SA8** of Section 5.3.3.

- MU_i verifies the time delay, computes the mutually shared key $SKEY_{ij}$, and calculating the hash value H_{u_2} checks whether $H_{u_2} \stackrel{?}{=} H_{s_2}$ holds or not and concludes the computed secret session key $SKEY_{ij}$ $(= SKEY_{ji})$ is mutually verified and confirmed and uses that key for message transmission in the future.

- **Step DS8:** Finally, MU_i also deletes the pair (en_k, sid_k) from the list L_{em_i} saved on its mobile device.

5.3.5 Emergency key-refilling phase

From the blocking stage, if an user MU_i try to re-synchronize with the system again, it needs a pair of emergency numbers and shadow identity, (en_k, sid_k) and these keys can be used only once. This key must be deleted from both lists, saved in the server-side and user-side after using it. Hence, after some time, a user requires to refill the list L_{em_i} with a new set of emergency key and shadow identity pairs before it goes running out. Therefore, MU_i sends a key-refilling request message and after verification the server SNS_j sends the new key list $L_{em_i}^{new}$. The detailed steps are as follows:

- **Step ER1:** In this scheme, I have suggested that MU_i can use m numbers of pairs among n numbers of (en_n, sid_n) pairs, where $(m < n)$.

- For generating a refill request message, MU_i selects $(m+1)^{th}$ pair from the list L_{em_i} and selects a random number N_{U_i} and current timestamp TS_u also.

- **Step ER2:** Next, MU_i calculates $Rfill_{reg_i} = H(en_{m+1} || \ sid_{m+1} || \ N_{U_i})$.

- MU_i sends a refill request message $Mrfill_u = \{E_{SKey_{ij}} \langle TID_{mu_i}, \ N_{U_i}, \ Rfill_{reg_i}, \ TS_u \rangle\}$ to SNS_j.

- **Step ER3:** After receiving the message $Mrfill_u$, SNS_j uses the session key $SKey_{ji}$ $(= SKey_{ij})$ and decrypts the message $Mrfill_u$ as $D_{SKey_{ji}} \langle Mrfill_u \rangle$.

- Next, SNS_j verifies the receiving delay comparing current timestamp TS_s^* and received timestamp TS_s, if $|TS_s^* - TS_s| \leq \triangle T$ or not.

- **Step ER4:** After successful verification, SNS_j extracts TID_{mu_i} and fetches the record corresponding to it.

- Next, SNS_j calculates $Rfill'_{reg_i} = H(en_{m+1}|| \ sid_{m+1}|| \ N_{U_i})$ and verify whether $Rfill'_{reg_i} \stackrel{?}{=} Rfill_{reg_i}$ or not. The execution terminates if the verification fails.
- Else, SNS_j produces a new list $L^{new}_{em_i}$ with n numbers of emergency number and shadow identity pair $\{(en_1^{new}, sid_1^{new}), (en_2^{new}, {sid_2}^{new}), ({en_3}^{new}, {sid_3}^{new}),\ldots, (en_n^{new}, sid_n^{new})\}$.
- **Step ER5:** SNS_j selects a random number N_{sj} and computes a hashed value $H_r = H(L^{new}_{em_i}|| \ N_{sj}|| \ en_{m+1}|| \ sid_{m+1})$.
- SNS_j fetches the current timestamp TS_s and sends the refilling response message $Mrfill_{res} = \{E_{SKey_{ji}}\langle TID_{mu_i}, L^{new}_{em_i}, H_r, N_{sj}, TS_s\rangle$ to MU_i via CT.
- **Step ER6:** CT only checks whether SNS_j is a registered server or not, for a registered server CT only forwards the message to MU_i.
- **Step ER7:** Upon receiving the message $Mrfill_{res}$, MU_i verifies the timestamp, if $|TS^*_s - TS_s| \leq \triangle T$ or not.
- **Step ER8:** After successful verification, MU_i decrypts the message $D_{SKey_{ij}}(Mrfill_{res})$, and extracts $L^{new}_{em_i}$. MU_i computes $H'_r = H(L^{new}_{em_i}|| \ N_{sj}|| \ en_{m+1}|| \ sid_{m+1})$ and verifies if $H'_r \stackrel{?}{=} H_r$ or not.
- **Step ER9:** If the verification succeeds, MU_i refills the list of emergency numbers and shadow identity pair with $L^{new}_{em_i}$. Otherwise, the execution of the refilling process will terminate and restarts again from the beginning.

5.4 Security Analysis

This section presents detail security analysis of the proposed $PRDoS$ scheme from several angles. First, I present the formal security of $PRDoS$ using random oracle based proof. Second, using informal security analysis, I discuss on how $PRDoS$ withstands several security attacks. Finally, I provide the Burrows–Abadi–Needham logic (also known as the BAN logic) based authentication proof of the proposed $PRDoS$ scheme.

5.4.1 Formal security using ROR model

In this subsection, I present a formal security model proposed $PRDoS$ scheme using widely-accepted Real-Or-Random (ROR) model [3], [186]. I define the security of the protocol, say

$\mathscr{P}$ by a set of games played between an participant $\mathcal{P}$ and an adversary $\mathcal{A}$.

Table 5.2: Notations used for the formal security analysis using ROR model

Notation	Description
q_h	Total number of executed hash H oracle queries
q_e	Total number of executed *Execute* oracle queries
q_s	Total number of executed *Send* oracle queries
l_h	The string length of hash results
l_b	The extracted string length of user biometrics
l_r	The string length of random numbers
$\mathcal{D}$	A finite Password space dictionary with size $\|\mathcal{D}\|$
ε_{bm}	The probability of false positive in biometrics
L_h	List of stored output of hash H oracle queries
L_r	List of stored outputs of random oracle queries
L_m	List of stored records transcripts between MU_i and SNS_j

- **Participants.** Let $\mathcal{P}^l$ is the l^{th} instance of the participants and $\mathcal{P}^l \in \{MU_i, SNS_j\}$.
- **Adversary.** An adversary $\mathcal{A}$ can have the complete control over the all transmissions according to the ROR model.

  For breaking the security of the protocol, $\mathcal{A}$ can execute various types of real attacks. The possible oracle queries that $\mathcal{A}$ can execute to simulate the model are described as follows.

  - ***Send***$(\mathcal{P}^l, m)$: When $\mathcal{A}$ executes the send query with a request message m to $\mathcal{P}^l$ and receives a reply message according to the rule of the protocol.
  - ***Execute***(MU_i, SNS_j): Using this query $\mathcal{A}$ executes a passive attack by eavesdropping on the messages communicated between two honest participants, SNS_j and MU_i.
  - ***Test***$(\mathcal{P}^l)$: Invoking this query, $\mathcal{A}$ can ask the current session key $SKEY$ from a participant $\mathcal{P}^l$. A *null* value will be returned if the session key is not established. Otherwise, if a fresh key is generated, $\mathcal{P}^l$ replies according to the result of an unbiased flipped coin c. $\mathcal{P}^l$ replies current session key $SKEY$ if $c = 1$ and a random number of same length to $\mathcal{A}$ if $C = 0$.

Table 5.3: Simulation of Send and Execute oracle queries

Simulation of *Send* oracle query of the proposed PRDoS protocol: **(a)** MU_i responses the $Send(MU_i, \textbf{start})$ query as follows. Compute $TID^*_{mu_i}$, X_{i_1}, H_{i_1}, TS_u as in Figure 5.3 and sends $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$. **(b)** $SNCS_j$ responds $Send(SNS_j, \langle TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\rangle)$ query as follows. Verify if $\|TS^*_u - TS_u\| \leq \triangle T$ and then computes D_{su} and P_1. Further checks the parameter H_{i_1}, if verification fails, terminate the current session. Moreover, SNS_j generates X_{j_1}, $SKEY_{ji}$ and H_{j_2}, and output $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$. **(c)** MU_i replies $Send(MU_i, \langle X_{j_1}, H_{j_2}, TS_s\rangle)$ query as follows. Verifies if $\|TS^*_s - TS_s\| \leq \triangle T$. If verification succeeded, calculates P_2 and $SKEY_{ij}$. Finally, checks H_{j_2}, and terminate the current session, if verification fails. Otherwise, accepts $SKEY_{ij}$ as the mutually authenticated session key as depicted in Figure 5.3. After establishment of the mutually authenticated session key, both MU_i & SNS_j terminate the session.
Simulation of *Execute*(MU_i, SNS_j) query take place in succession of *Send* queries as follows: MU_i Computes X_{i_1} and H_{i_1} as depicted in Figure 5.3 and sends the message $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ to SNS_j. SNS_j Calculates X_{j_1}, H_{i_2} and $SKEY_{ji}$ as given in Figure 5.3 and sends the authentication response message $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$ to MU_i. Note that $\langle TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\rangle \leftarrow Send(MU_i, \textbf{start})$, $\langle X_{j_1}, H_{j_2}, TS_s\rangle \leftarrow Send(SNS_j, \langle TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\rangle)$. Finally, Msg_u and Msg_s are returned.

  - ***Corrupt***(MU_i, α): executing this query $\mathcal{A}$ can acquire the secret credentials such as password, biometric, etc. of an honest user MU_i depending the value of α.
  - ***Reveal***$(\mathcal{P}^l)$: Through this query $\mathcal{A}$ can reveal the current session key $SKEY$ generated between two honest participants SNS_j and MU_i.

- **Freshness.** An instance $\mathcal{P}^l$ is fresh, if $\mathcal{P}^l$ is in accept state and the session key $SKEY$ is not revealed to the adversary $\mathcal{A}$.

- **Semantic security.** If in an event, an adversary $\mathcal{A}$ tries to chose a bit c by executing $Test(\mathcal{P}^l)$ queries and returns a guessed bit c'. $\mathcal{A}$ will win if $c' = c$. Let $Succ_{\mathcal{A}}$ be the event that $\mathcal{A}$ wins the game. The advantage function of $\mathcal{A}$ by guessing the correct bit

c' for violating the semantic security of the protocol $\mathscr{P}$ can be defined as $Adv_{\mathscr{P}}^{PRDoS}(\mathcal{A}) = |2P_r[Succ_{\mathcal{A}}] - 1| = |2.P_r[c = c'] - 1|$.

Definition 5.1. *The semantic security of the proposed PRDoS biometrics password based authentication protocol is depends on the advantage function* $Adv_{\mathscr{P}}^{PRDoS}(\mathcal{A})$ *is slightly larger than* $\max\{q_s * \frac{1}{|\mathcal{D}|}, q_s * \frac{1}{2^{l_b}}, q_s * \varepsilon_{bm})\}$ *[168], where* q_s, $|\mathcal{D}|$, l_b *and* ε_{bm} *are describes in Table 5.2.*

Theorem 5.1. *Suppose a polynomial time bounded adversary* $\mathcal{A}$ *wants to break the semantic security of the proposed PRDoS authentication protocol* $\mathscr{P}$ *within the time limit* $t_{\mathcal{A}}$. *Then, the advantage function of* $\mathcal{A}$ *will be:*

$$Adv_{\mathscr{P}}^{PRDoS}(\mathcal{A}) \leq \frac{q_h^2 + 18q_h}{2^{l_h}} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}$$

where q_h, q_e, q_s, l_h, l_b, l_r, $\mathcal{D}$, *and* ε_{bm} *are as described in Table 5.2.*

Proof. The formal proof of the proposed protocol consists of five games Gm_i $(i = 0, 1, 2, 3, 4)$ between participants $\mathcal{P}^l$ and adversary $\mathcal{A}$. Using $Test$ query, an adversary $\mathcal{A}$ can try to explore a correct bit c in the game Gm_i. The event $Succ_i$ is used to define the success of $\mathcal{A}$ in the game Gm_i and the associated probability is represented by $P_r[Succ_i]$. Starting from Gm_0, which is an actual attack against the proposed PRDoS protocol $\mathscr{P}$, I slowly increase the attack ability of $\mathcal{A}$ and terminate the game when $\mathcal{A}$ exhausts its ability except the tossing.

- **Game G_0:** In this initial game, the real situation of the protocol is simulated, according to the semantic security, I have

$$Adv_{\mathscr{P}}^{PRDoS}(\mathcal{A}) = |2P_r[Succ_0] - 1|. \tag{5.1}$$

- **Game G_1:** In this game, G_0 is transformed into G_1 by simulating the passive attack ability of $\mathcal{A}$. $\mathcal{A}$ eavesdrops a message m_i using $Execute$ query. Using $Test$ query, $\mathcal{A}$ checks whether the outcome of the query is the genuine session key $SKEY$ or not. But the detailed procedure of $Send$, $Test$, $Execute$ oracle queries are described in Table 5.3, which specifies that the execution of these queries does not increase the winning probability of $\mathcal{A}$ in this game. Moreover, some lists L_h, L_r, and L_m are used in this

game for storing the outputs of several oracle queries. Therefore, G_1 is equivalent to G_0, and I have,

$$P_r[Succ_1] = P_r[Succ_0]. \tag{5.2}$$

- **Game G_2:** By simulating the $Send$ and $Hash$ oracle queries, game G_1 is transformed into G_2. An active attack is modeled in game G_2. The adversary $\mathcal{A}$ sends several $Hash$ queries to identify the collisions. Two messages, $Msg_u = \{TID^*_{mu_i}, X_{u_1}, H_{i_1}, TS_u\}$ and $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$ are communicated between MU_i and SNS_j. Both the participants MU_i and SNS_j use random numbers R_u and R_s, and current timestamp TS_u and TS_s respectively and the collision probability of random numbers is at most $\frac{(q_s+q_e)^2}{2^{l_r+1}}$. According to the birthday paradox, the collision probability of $Hash$ query is at most $\frac{q_h^2}{2^{l_h+1}}$. Therefore, I get,

$$|P_r[Succ_2] - P_r[Succ_1]| \leq \frac{(q_s+q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}}. \tag{5.3}$$

- **Game G_3:** The motivation of the game G_3 is to simulate the collision probability of other oracle queries and model the situation where adversary $\mathcal{A}$ can guess the correct message fortunately without considering $Hash$ oracle queries. As the proposed protocol needs two message communications for login and authentication, I have to consider the following two cases:

  - **Case 1:** According to the $Send(SNS_j, Msg_u)$ query on $Msg_u\{TID^*_{mu_i}, X_{u_1}, H_{i_1}, TS_u\}$, the hash value is $H_{i_1} = H(IDmu_i \,||X_{u_1}\, ||R_u\, ||TS_u) \in L_r$ must hold. Collision probability at most will be $\frac{q_h}{2^{l_h}}$. For successful attack launching, $H(PWmu_i||\ \gamma_i) \oplus a$ of F_i^1, $H(IDmu_i\ ||PWmu_i||\ \gamma_i\ ||a)$ of F_i^2 and $B_{us} \oplus R_u \oplus TS_u \oplus H(ID_{ss})$ of Xi_1 should be exposed to $\mathcal{A}$. Thus, the overall collision probability for this $Hash$ queries will be $\frac{4q_h}{2^{l_h}}$. Moreover, message Msg_u contains parameter R_u, and $Msg_u \in L_m$, hence the collision probability will be $\frac{q_s}{2^{l_r}}$.
  - **Case 2:** I have considered the authentication message Msg_s that will be sent by SNS_j to MU_i. To respond to the $Send(MU_i,\ Msg_s)$ query of $\mathcal{A}$, $H_{j_2} \in L_r$ must hold and the collision probability will be $\frac{q_h}{2^{l_h}}$. Moreover, for launching the attack positively, $H(H(ID_{mu_i} \oplus Z_{us})||S_j)$ of D_{su}, $H(ID_{ss})$ of P_1, $H(ID_{mu_i}\ ||X_{u_1}\ ||P_1\ ||TS_u)$ of H_{j_1}, and $H(ID_{mu_i}||ID_{ss}||D_{su}||P_1||R_s||TS_u||TS_s)$ of $SKEY_{ji}$ must be disclosed to $\mathcal{A}$. Hence, the collision probability of this part will be $\frac{5q_h}{2^{l_h}}$. Finally, for the message transcript $Msg_s \in L_m$, and I have the probability $\frac{q_s}{2^{l_r}}$.

Considering all cases, I have obtained,

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

- **Game G_4:** In this game G_4, an attack is modeled using *Corrupt* queries where the adversary $\mathcal{A}$ tries to find out the password and biometric in both online and offline situations. Executing *Corrupt* query, $\mathcal{A}$ tries to guess the password PW_{mu_i}, with maximum probability up to $\frac{q_s}{|\mathcal{D}|}$ and user biometric γ_i having probability $\max\{q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}$ [157], [36]. Except these guessing attacks, game G_3 and G_4 are identical. Therefore, I can say,

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

Considering all the above games, no information about the correct bit of b is revealed to $\mathcal{A}$ other than the guessing. Hence, I can obtain,

$$Pr[S_4] = \frac{1}{2}. \tag{5.6}$$

Using the rule of triangular inequality, I can obtain,

$$\begin{aligned} |Pr[S_0] - \frac{1}{2}| &= |Pr[S_1] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_4]| \\ &\leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad + |Pr[S_3] - Pr[S_4]|. \end{aligned} \tag{5.7}$$

Solving Equations (5.1)-(5.7), I can get,

$$\begin{aligned} \frac{1}{2}Adv_{\mathscr{P}}^{PRDoS} &= |Pr[S_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_r+1}} + \frac{q_h^2}{2^{l_h+1}} + \frac{2q_s}{2^{l_r}} + \frac{9q_h}{2^{l_h}} \\ &\quad + \max\{\frac{q_s}{|\mathcal{D}|}, q_s(\frac{1}{2^{l_b}}, \varepsilon_{bm})\}. \end{aligned} \tag{5.8}$$

Finally, multiplying both sides of Equation (5.8) by 2 and reorganizing the terms, I can obtain

$$Adv_{\mathscr{P}}^{PRDoS} \leq \frac{q_h^2 + 18q_h}{2^{l_h}} + \frac{(q_s + q_e)^2 + 4q_s}{2^{l_r}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}.$$

Therefore, the theorem is proved. □

5.4.2 Informal security analysis

In this subsection, I discuss on how the proposed $PRDoS$ scheme resists various active and passive security attacks.

DDoS attack

According to the proposed protocol, at the time of login, MU_i sends $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ to OSN server SNS_j via cellular tower CT. After receiving Msg_u, SNS_j extracts temporary id TID_{mu_i} and the random number R_u and checks whether it is a repeated message or not. While SNS_j receives a login request from the same MU_i more than three times within a short time frame, SNS_j blocks the user MU_i. Therefore, if an adversary $\mathcal{A}$ wants to launch a denial-of-service attack, maximum he/she can send the login request message four times. So, denial-of-service only by sending a huge service request is not possible in this protocol. Hence, the protocol can withstand DoS or DDoS attacks.

Replay attack

In the proposed protocol, MU_i sends $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ to OSN server SNS_j. After receiving Msg_u, SNS_j verify if $|TS^*_u - TS_u| \geq \triangle T$, where $\triangle T$ is the permissible transmission delay. SNS_j discards Msg_u, if the verification fails. Further, SNS_j computes hash value $H_{j_1} = H(ID_{mu_i} || X_{u_1} || P_1 || TS_u)$ and verify whether $H_{j_1} = H_{i_1}$ or not, and discards Msg_u if verification fails. Moreover, SNS_j also saved the parameter $\langle ID_{mu_i}, TID_{mu_i}, Z_{us} \rangle$ in its database and can extract the random number R_u which is used to differentiate two different login requests. Similarly, SNS_j sends the authentication message Msg_s, MU_i verifies the timestamp, random nonce, and hash value, and discards Msg_s if any one of the verification will fail. Therefore, when an adversary $\mathcal{A}$ wants to launch a reply attack he/she has to send the replay message $Msg^*_u = \{TID^*_{mu_i}, X^*_{i_1}, H^*_{i_1}, TS^*_u\}$. SNS_j first checks the timestamp, then extracts TID_{mu_i} and checks if it is the same with stored TID_{mu_i}. Then Msg^*_u will be treated as a replay message. Thus, the proposed protocol can resist strong replay attacks.

Man-in-the-middle attack

In this type of attack, adversary $\mathcal{A}$ establishes an independent connection with both SNS_j and MU_i. For launching this attack, $\mathcal{A}$ may modify some parameters of the login request message Msg_u of MU_i or authentication message of SNS_j for invalidating the message. But in the proposed protocol, all the parameters of either Msg_u or Msg_s both are generated using

random nonces, hash function, and bitwise XOR operations. Therefore, for modifying those parameters $\mathcal{A}$ needs the original credentials B_{us}, E_{su}, R_u, ID_{ss}, etc. Hence, the protocol can withstand the man-in-the-middle attack.

Stolen/lost mobile device attack

If an adversary $\mathcal{A}$ gets the mobile device of a user either by stealing or due to losing, $\mathcal{A}$ may try to masquerade himself or herself as that legitimate user. But $\mathcal{A}$ cannot generate the login request Msg_u from the easily accessible stored parameters F_i^1 and F_i^2. For generating Msg_u, $\mathcal{A}$ needs the identity ID_{mu_i} , password PW_{mu_i} and biometric key γ_i. But it is not feasible to regenerate those credentials from saved parameters $F_i^1 = H(PW_{mu_i}|| \gamma_i) \oplus a$ and $F_i^2 = H(ID_{mu_i} ||PW_{mu_i}|| \gamma_i||a)$ due to the one-way property of $H(\cdot)$. Hence, $\mathcal{A}$ cannot misuse the mobile device and my protocol can withstand such types of attacks.

Password guessing attack

According to my proposed protocol, user MU_i needs ID_{mu_i}, password PW_{mu_i} and biometric key γ_i, for generating login message Msg_u. Obtaining the mobile device an adversary $\mathcal{A}$ can extract F_i^1 and F_i^2 but cannot regenerate ID_{mu_i} and PW_{mu_i} as these are not feasible to guess from F_i^1 and F_i^2. Thus, the protocol can resist offline password guessing attacks.

Known-key secrecy or forward secrecy

Forward secrecy confirms that an adversary $\mathcal{A}$ does not get any help for computing past session keys from a compromised session key. In my proposed protocol, a session key $SKEY_{ji}$ $(= SKEY_{ij} = H(ID_{mu_i}|| ID_{ss}||D_{su}|| P_1||R_s||TS_u||TS_s))$ is generated using random number R_s, R_u, current timestamp TS_u, and TS_s. The use of timestamps and random nonce make the session keys unique and fresh. Therefore, a compromised session key does not provide any essential information that can help $\mathcal{A}$ for computing previous session keys.

User anonymity

According to the proposed protocol, an adversary $\mathcal{A}$ may overhear a login request Msg_u $= \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ transmitted between MU_i to SNS_j, where instead of its original identity ID_{mu_i}, MU_i sends its temporary identity TID_{mu_i} embedded with other parameters $TID^*_{mu_i} = TID_{mu_i} \oplus H(PID_{ss}||TS_u)$. Further, though the hash value $H_{i_1} = H(ID_{mu_i}||X_{u_1}||R_u||TS_u)$ and $X_{u_1} = B_{us} \oplus R_u \oplus TS_u \oplus H(ID_{ss})$ contains the original identity

of MU_i, from one-way hash function it is not possible to regenerate ID_{mu_i}. Therefore, $\mathcal{A}$ cannot extract the original identity of MU_i from Msg_u. Similarly, from the authentication reply $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$ transmitted between SNS_j to MU_i, $\mathcal{A}$ cannot extract original identity of SNS_j. Thus, the user anonymity property is preserved in the proposed scheme.

Parallel session and reflection attack

As discussed in Sections 5.4.2 and 5.4.2, an adversary $\mathcal{A}$ can extract the user's credentials neither from the mobile device nor from an eavesdropped message. Therefore, if $\mathcal{A}$ wants to set up a parallel session with SNS_j and pretends himself or herself as a genuine user, that is not possible by $\mathcal{A}$ as he/she cannot generate an effective login message Msg_u. Thus, the proposed protocol can withstand The Parallel Session and Reflection Attack.

Session-key security

As discussed in Section 5.3.3, in the proposed protocol mutually shared session key $SKEY_{ij}$ $(= SKEY_{ji})$ is generated as

$$\begin{aligned} SKEY_{ij} &= H(ID_{mu_i}||ID_{ss}||B_{us}||R_u||P_2||TS_u||TS_s) \\ &= H(ID_{mu_i}||ID_{ss}||D_{su}||R_u||P_2||TS_u||TS_s) \\ &= H(ID_{mu_i}||ID_{ss}||D_{su}||P_1||P_2||TS_u||TS_s) \\ &= H(ID_{mu_i}||ID_{ss}||D_{su}||P_1||R_s||TS_u||TS_s) = SKEY_{ji} \end{aligned}$$

Therefore, for computing a new session key an adversary $\mathcal{A}$ requires ID_{mu_i}, ID_{ss}, $D_{su} = (B_{us})$, $P_1 = (R_u)$, etc. as parameters. As discussed earlier, these parameters are not feasible to compute. Thus, the mutually shared session key is completely secure.

Server impersonation attack

An adversary $\mathcal{A}$ may try to pretend himself or herself as a social network server SNS_j by trying to reply MU_i with a valid message. After receiving a login request message Msg_u from MU_i, SNS_j responds with an authorization reply $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$. Msg_s has two parameters $X_{j_1} = D_{su} \oplus R_s \oplus TS_s \oplus ID_{mu_i}$, and $H_{j_2} = H(ID_{mu_i}||\ P_1||R_s||\ TS_u||TS_s||\ SKEY_{ji})$ and $B_{us}{=}D_{su} = H(H(ID_{mu_i} \oplus Z_{us})||S_j)$. $\mathcal{A}$ cannot generate these parameters as he or she does not have the server secret key S_j and random number Z_{us}. Hence, the proposed protocol can resist server impersonation attacks.

5.4.3 Mutual authentication proof using BAN logic

The basic notations and logical postulates of BAN logic are described in Section 2.5. According to the analytic procedures of the BAN logic, the security of the proposed protocol will be proven, if the following two goals must hold:

- **Goal** G_1. $MU_i \mid\equiv (MU_i \overset{SKey}{\longleftrightarrow} SNS_j)$.
- **Goal** G_2. $SNS_j \mid\equiv (MU_i \overset{SKey}{\longleftrightarrow} SNS_j)$.

The generic forms of the messages in the proposed protocol are as follows:

- **Message 1.** $MU_i \rightarrow SNS_j$: $\{T(ID^*_{mu_i}),\ H(H(ID_{mu_i} \oplus Z_{us}) \mid\mid S_j) \oplus R_u \oplus TS_u \oplus H(ID_{ss}),\ TS_u,\ H_{i_1}\}$.
- **Message 2.** $SNS_j \rightarrow MU_i : \{D_{su} \oplus R_s \oplus TS_s \oplus (ID_{mu_i}),\ TS_s,\ H_{j_2}\}$.

According to the process of BAN logic, the above generic messages must be transformed to the idealized forms as given below.

- **Message 1.** $MU_i \rightarrow SNS_j$: $\{T(ID_{mu_i}),\ TS_u,\ \langle (ID_{mu_i}),\ Z_{us},\ R_u,\ TS_u,\ H(ID_{ss})\rangle_{S_j},\ H_{i_1}\}$.
- **Message 2.** $SNS_j \rightarrow A$: $\{TS_s,\ \langle R_s,\ TS_s,\ (ID_{mu_i})\rangle_{S_j},\ H_{j_2}\}$.

To complete the authentication proof of the proposed protocol, the following assumptions are made:

- Assumption 1: $MU_i \mid\equiv \#(TS_s)$;
- Assumption 2: $SNS_j \mid\equiv \#(TS_u)$;
- Assumption 3: $MU_i \mid\equiv (MU_i \overset{B_{us}}{\rightleftharpoons} SNS_j)$;
- Assumption 4: $SNS_j \mid\equiv (MU_i \overset{B_{us}}{\rightleftharpoons} SNS_j)$;
- Assumption 5: $MU_i \mid\equiv SNS_j \Rightarrow (ID_{ss}, R_s, TS_s)$;
- Assumption 6: $SNS_j \mid\equiv MU_i \Rightarrow ((ID_{mu_i}), R_u, TS_u)$;
- Assumption 7: $MU_i \mid\equiv TS_u$;
- Assumption 8: $MU_i \mid\equiv R_u$;

- Assumption 9: $MU_i \mid\equiv ID_{mu_i}$;
- Assumption 10: $MU_i \mid\equiv ID_{ss}$;
- Assumption 11: $SNS_j \mid\equiv TS_s$;
- Assumption 12: $SNS_j \mid\equiv R_s$;
- Assumption 13: $SNS_j \mid\equiv ID_{ss}$.

Considering the fundamental rules of the BAN logic and the assumptions are made, I analyze and provide the procedures of achievement of both the goals **Goal** G_1 and **Goal** G_2 as follows.

According to the **Message 1**, I can get

- S_1: $SNS_j \lhd \{ID_{mu_i}, TS_u, \langle ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss})\rangle_{S_j}, H_{i_1}\}$.
- S_2: Based on the rule AR, I have, $SNS_j \lhd \langle ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss})\rangle_{S_j}$.
- S_3: As per the rule MMR and Assumption 4, I get, $SNS_j \mid\equiv MU_i \mid\sim (ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss}))$.
- S_4: Based on the rule FCR and Assumption 2, I obtain, $SNS_j \mid\equiv \#(ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss}))$.
- S_5: Using rule NVR, I obtain, $SNS_j \mid\equiv MU_i \mid\equiv (ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss}))$.
- S_6: According to Assumption 6 and JR, I have, $SNS_j \mid\equiv (ID_{mu_i}, Z_{us}, R_u, TS_u, H(ID_{ss}))$.
- S_7: Using rule AR and S_6, I get, $SNS_j \mid\equiv R_u$, $SNS_j \mid\equiv TS_u$, $SNS_j \mid\equiv ID_{mu_i}$.
- S_8: According to Assumption 11, 12, and 13, I obtain, $SNS_j \mid\equiv ID_{ss}$, $SNS_j \mid\equiv TS_s$ and $SNS_j \mid\equiv R_s$.
- S_9: As $SKey_{ji} = H((ID_{mu_i} \mid\mid ID_{ss} \mid\mid D_{su} \mid\mid M_1 \mid\mid R_s \mid\mid TS_u \mid\mid TS_s)$ and the results in Steps S_7 and S_8 give $SNS_j \mid\equiv (MU_i \overset{SKey_{ji}}{\longleftrightarrow} SNS_j)$. **(Goal G_2)**
- S_{10}: According to rule AR and the message 2, I get, $MU_i \lhd \langle R_s, TS_s\rangle_{S_j}$.
- S_{11}: Using rule MMR and Assumption 3, I obtain, $MU_i \mid\equiv SNS_j \mid\sim (R_s, TS_s)$.

- S_{12}: Based on rule FCR and Assumption 1, I get, $MU_i \mid\equiv \#(R_s, TS_s)$.
- S_{13}: According to rule NVR, I have, $MU_i \mid\equiv SNS_j \mid\equiv (R_s, TS_s)$.
- S_{14}: According to rule JR and Assumption 5, I get, $MU_i \mid\equiv (R_s, TS_s)$.
- S_{15}: Rule AR and S_{14} give, $MU_i \mid\equiv R_s$, $MU_i \mid\equiv TS_s$.
- S_{16}: Using Assumption 7-10, I get, $MU_i \mid\equiv ID_{mu_i}$, $MU_i \mid\equiv ID_{ss}$, $MU_i \mid\equiv TS_u$, $MU_i \mid\equiv R_s$.
- S_{17}: According to the results of Steps S_{15} and S_{16}, I obtain, $MU_i \mid\equiv (MU_i \overset{SKey_{ji}}{\longleftrightarrow} SNS_j)$. **(Goal G_1)**

Finally I reached to **Goal** G_1 and **Goal** G_2, ensuring that both MU_i and SNS_j mutually authenticate each other.

5.5 Formal Security Verification Using ProVerif

ProVerif is an applied pi calculus based simulation tool that is used in practice to test if an attacker is able to compromise the session key [1]. This section presents the formal security verification of proposed *PRDoS* using the ProVerif simulation tool.

Figure 5.5 shows the required code for channel declaration, constants, functions, variables, equations, events, and queries declaration. Figure 5.6 shows the required code for the process of mobile user registration, login, and key establishment. The required code of the OSN server that would execute in parallel at the time of registration and authentication are modeled in Figure 5.7.

Finally, the codes discussed above are executed in ProVerif 1.93 simulation tools. The obtained results are presented in Figure 5.8. The results show that the shared session key is secured, all other events and queries are also secured from the attackers. Therefore, the security of the proposed PRDoS scheme is successfully verified.

5.6 Performance Analysis and Comparison

At the beginning of this section, I have compared our proposed scheme with some existing authentication schemes [202], [82], [147], [35], [117], [17] in terms of functionality and security. From the comparison report, which is presented in Table 5.4, it is observed that the proposed

```
(* ———— channels ————*)
free pubch: channel. (* public channel *)
free scrch: channel [private]. (* private channel *)
(* ———— shared keys ———— *)
free SKEYij:bitstring [private].(* the session key of user *)
free SKEYji:bitstring [private].(*the session key of server*)
(* ———— Servers secret key ———— *)
free Sj:bitstring [private].
free Zus:bitstring [private].
(* ———— constants ———— *)
free IDss:bitstring [private].
free IDmui:bitstring [private].
free TIDmui:bitstring [private].
free PWmui:bitstring [private].
const Bmui:bitstring [private].
(* ———— functions and equations ———— *)
fun h(bitstring):bitstring. (* hash function *)
fun Gen(bitstring):bitstring.(*Fuzzy extractor Gen function*)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring.(*string concatenatn*)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(* ———— aims for verification ———— *)
query attacker(SKEYij).
query attacker(SKEYji).
query muID:bitstring; inj-event(MUserAuth(muID)) ==⟩ inj-event(MUserStart(muID)).
(* ———— event ———— *)
event MUserStart(bitstring). (* User starts authentication *)
event MUserAuth(bitstring). (* User is authenticated *)
```

Figure 5.5: The ProVerif code for declaration of channels, variables, events

scheme has the same security as the scheme proposed in [35]. Compared with [17], our protocol can provide resistance to DoS attack. The protocols are given in [202], [82], [147], [117] cannot provide login phase efficiency, so these schemes are not suitable for mobile online

```
(*————user starts———*)
let MUser=
new a:bitstring;
new b:bitstring;
let gamma = Gen(Bmui) in
let BPWi = h(con(IDmui,h(con(PWmui,con(gamma,a))))) in
out(scrch,(IDmui,xor(BPWi,b)));
in(scrch,(tEus:bitstring, tLemi:bitstring));
let F1 = xor(h(con(PWmui,gamma)),a) in
let F2 = h(con(IDmui,con(PWmui,con(gamma,a)))) in
!(
event MUserStart(IDmui);
let a1 = xor(F1,h(con(PWmui,gamma))) in
let tF2 = h(con(IDmui,con(PWmui,con(gamma,a1)))) in
if F2 = tF2 then
let tBus = xor(tEus,BPWi) in
new Ru:bitstring;
new TSu:bitstring;
let Xu1 = xor(tBus,xor(Ru,xor(TSu,h(IDss)))) in
let Hi1 = h(con(IDmui,con(Xu1,con(Ru,TSu)))) in
out(pubch,(TIDmui,Xu1,Hi1,TSu));
in(pubch,(tXj1:bitstring,tTSs:bitstring,tHj2:bitstring));
let P2 = xor(tXj1,xor(tTSs,xor(IDmui,tBus))) in
let SKEYij = h(con(IDmui,con(IDss,con(tBus,con(Ru,con(P2,con(TSu,tTSs))))))) in
let Hi2 = h(con(IDmui,con(Ru,con(P2,con(TSu,con(tTSs,SKEYij)))))) in
if Hi2 = tHj2 then
0
).
```

Figure 5.6: The ProVerif code for mobile user

social networks. Therefore, considering security and functionality concerns, my proposed scheme is the most efficient and secure one.

```
let SReg =
in(scrch,(muID:bitstring,uBPW:bitstring));
let Bus = h(con(h(xor(muID,Zus)),Sj)) in
let Eus = xor(Bus,uBPW) in
new Lemi: bitstring;
out(scrch,(Eus,Lemi)).
```

```
let SAuth =
in(pubch,(muID:bitstring,mXu1:bitstring,mTSu:bitstring,mHi1:bitstring));
let Dsu = h(con(h(xor(muID,Zus)),Sj)) in
let P1 = xor(Dsu,xor(mXu1,xor(mTSu,h(IDss)))) in
let Hj1 = h(con(muID,con(mXu1,con(P1,mTSu)))) in
if Hj1 = mHi1 then
event MUserAuth(muID);
new Rs:bitstring;
new TSs:bitstring;
let Xj1 = xor(Dsu,xor(Rs,xor(TSs,muID))) in
let SKEYji = h(con(muID,con(IDss,con(Dsu,con(P1,con(Rs,con(mTSu,TSs))))))) in
let Hj2 = h(con(muID,con(P1,con(Rs,con(mTSu,con(TSs,SKEYji)))))) in
out(pubch,(Xj1,TSs,Hj2)).
let S = SReg — SAuth.
process !MUser — !S
(*SReg — SAuth.*)
```

Figure 5.7: The ProVerif code for OSN server

Next, I analyze the communication overhead and computation cost of the proposed scheme.

5.6.1 Communication cost analysis

To analyze the communication overhead of the proposed scheme, I have considered the standard size of the various cryptographic functions. The size of the hash function $H(\cdot)$ outputs (for SHA-1 hash function [63]) and identity of the entity both are 160 bits long. The random numbers and symmetric encryption/decryption outputs (for AES-128 [64]) are 128 bits long.

```
RESULT not attacker(SKEYij[]) is true.
– Query not attacker(SKEYji[])
nounif mess(scrch[],(muID_4402,uBPW_4403))/-
5000
Completing...
200 rules inserted. The rule base contains 200 rules.
22 rules in the queue.
Starting query not attacker(SKEYji[])
RESULT not attacker(SKEYji[]) is true.
– Query inj-event(MUserAuth(muID_29)) ==⟩ inj-event(MUserStart(muID_29))
nounif mess(scrch[],(muID_8355,uBPW_8356))/-
5000
Completing...
200 rules inserted. The rule base contains 200 rules.
28 rules in the queue.
Starting query inj-event(MUserAuth(muID_29)) ==⟩ inj-event(MUserStart(muID_29))
RESULT inj-event(MUserAuth(muID_29)) ==⟩ inj-event(MUserStart(muID_29))
is true.
```

Figure 5.8: The ProVerif simulation results

The size of timestamps is 32 bits. For calculating communication overhead, I have considered only the required message communications in the user login and authentication phase and have not considered the message communications require in the registration phase as it is a one-time process.

In the proposed scheme, a mobile user MU_i sends a login message $Msg_u = \{TID^*_{mu_i}, X_{i_1}, H_{i_1}, TS_u\}$ in Section 5.3.2 and after receiving the message Msg_u, SNS_j responds with $Msg_s = \{X_{j_1}, H_{j_2}, TS_s\}$ in Section 5.3.3. The total size of Msg_u is $(160 + 160 + 160 + 32)$=512 bits and Msg_s is $(160 + 160 + 32)$= 352 bits. As both messages need to be retransmitted by the cellular tower, the total communication overhead of the proposed scheme is 1728 bits. I have analyzed and shown the communication overhead of the proposed scheme with some other existing related schemes in Table 5.5.

From Table 5.5, it is observed that my proposed scheme requires less communication overhead than other schemes except for the scheme in [147]. The scheme in [147] incurs a lower number of bits for cost, but it requires 3 rounds of message communication, which can

Table 5.4: Comparison of security and functionality features with related existing schemes

Security/Functionality Attribute	Zhang *et al.* [202]	He *et al.* [82]	Qu-Tan [147]	Challa *et al.* [35]	Li *et al.* [117]	Banerjee *et al.* [17]	Our
Denial of service attack	X	✓	✓	✓	✓	X	✓
Strong replay attack	✓	X	✓	✓	✓	✓	✓
Man-in-the-middle attack	X	X	X	✓	✓	✓	✓
Stolen/lost device attack	NA	NA	✓	✓	✓	✓	✓
Password guessing attack	NA	NA	X	✓	✓	✓	✓
Privileged insider attack	NA	NA	✓	✓	✓	✓	✓
Parallel session and reflection attack	X	X	X	✓	✓	X	✓
Server impersonation attack	X	X	✓	✓	✓	✓	✓
Known session key secrecy	✓	X	✓	✓	✓	✓	✓
User anonymity/traceability provision	✓	✓	✓	✓	✓	X	✓
Forward secrecy	X	X	✓	✓	✓	✓	✓
Session key security	✓	✓	✓	✓	X	✓	✓
Login phase efficiency	X	X	X	✓	X	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓
Formal security analysis	X	X	X	X	✓	✓	✓
Simulation using ProVerif/AVISPA	X	X	X	✓	✓	✓	✓

Note: X: does not support a particular feature; ✓: supports a particular feature.

Table 5.5: Comparison of communication costs

Scheme	Communication rounds	No. of bits
Zhang *et al.* [202]	2	$2048d + 7328$
He *et al.* [82]	4	$2048(4d+1)$
Qu-Tan [147]	3	768
Challa *et al.* [35]	3	2528
Li *et al.* [117]	4	2720
Banerjee *et al.* [17]	3	2560
Our	4	1728

Note: d is a positive number.

be expensive in respect of communication cost. Message propagation time in [147] can be a valuable parameter due to the large size of the online social networks.

Remark 5.1. *As discussed in Section 5.3.4, if the system enters into a denial-of-service*

attack, no extra communication overhead will be required. MU_i *only sends login requests, size of* 512 *bits. But* SNS_j *does not send any reply message. For resynchronize the system again,* MU_i *sends the synchronization request* $MSync_u$*, size of* $(160 + 160 + 32) = 352$ *bits and after receiving that* SNS_j *will communicate the normal authentication message, size of 352 bits. Hence, the communication overhead can be maximum* $(352 + 352) = 704$ *bits.*

Table 5.6: Approximate time complexity of cryptographic functions [35], [202]

Symbol	Description	Execution time (in seconds)
T_h	One-way hash function	0.00032
T_{sym}	symmetric key encryption/decryption	0.0087
T_a	Elliptic curve point addition	0.0044
T_m	Elliptic curve point multiplication	0.0171
T_{fe}	Fuzzy extractor operation	$\approx T_m$
T_{exp}	Modular exponential operation	$\approx \frac{1}{2}T_{sym}$
T_x	Bitwise XOR operation	$\approx Negligible$

5.6.2 Computation cost analysis

According to the experimental results described in [35], I have considered and discussed various cryptographic functions, with their notations and execution times on a 2.4 GHz processor with 4 GB RAM, in Table 5.6. Mobile user MU_i registration and OSN server registration SNS_j both will execute only once, I have not considered them for computation overhead calculation. For extracting user biometric B_{mu_i}, I required one fuzzy extractor $Rep(\cdot)$ operation of $T_{fe} \approx T_m$ and 8 hash operation T_h and 8 XOR operations T_x in the user login phase 5.3.2.

Next, cellular tower (CT) requires $2T_h + T_x$ operations before forwarding the login message. In authentication phase 5.3.3, SNS_j needs $8T_h + 8T_x$. Moreover, MU_i requires extra $2T_h + 3T_x$ operation for mutual authentication. Hence, altogether the proposed scheme requires $20T_h + 17T_x + T_{fe}$ operation time as overhead. As bitwise XOR operation requires negligible time for computation, the computation cost will be $20T_h + T_{fe} = 20*0.00032 + 0.0171 = 0.0235$ seconds. Similarly, I have calculated the computation overhead and execution time. The comparative analysis of the same is presented in Table 5.7, which shows that the required

computation overhead of my proposed scheme is $20T_h + T_{fe}$. As the computation time of the one-way hash function is minimal compared with the elliptic curve point addition, elliptic curve point multiplication and symmetric key encryption/decryption, the proposed scheme incurs less communication overhead compared to others. It is clear from the above discussion that the proposed scheme provides better security and performance efficiency compared to other related schemes.

Table 5.7: Comparison of computation costs with related existing schemes

Scheme	Computation overhead	Execution Time (in milliseconds)
Zhang *et al.* [202]	$8T_{exp}$+$3T_m$	$\approx$ 0.0861
He *et al.* [82]	$18T_{exp}$+$13T_m$	$\approx$ 0.3006
Qu-Tan [147]	$13T_h$ +$14T_{exp}$	$\approx$0.06506
Challa *et al.* [35]	$14T_m$+ $12T_h$	$\approx$ 0.24324
Li *et al.* [117]	$19T_h$+$6T_m$	$\approx$0.10868
Banerjee *et al.* [17]	$19T_h$+$10T_{sym}$ +T_{fe}	$\approx$0.11018
Our	$20T_h + T_{fe}$	$\approx$ 0.0235

Remark 5.2. *According to Section 5.3.4, when an adversary launches DoS attacks, no extra computation overhead will be required for MU_i as well as for SNS_j. Only MU_i requires T_{FE} $+8T_h$ for sending the login request and CT requires $2T_h$ for forwarding that message. But, before completing the verification process of that login message, SNS_j receives three or more login requests from MU_i. SNS_j stops message verification and blocks MU_i. To resynchronize the system again, MU_i sends the synchronization request $MSync_u$ which requires only one T_h and after receiving $MSync_u$, SNS_j reply with a normal authentication reply message that requires $8T_h$ overhead and after receiving authentication reply, MU_i requires $2T_h$ overhead. Hence, altogether $11T_h$ extra overhead will be required for the DoS attack remedy phase.*

5.7 Experimental Results and Discussions

In this section, I perform the extensive practical experiments and simulate the proposed scheme with two different objectives. First, in presence of DDoS attacker nodes, I study the impact of the proposed scheme on network throughput and network delay. This simulation

is performed using the widely-recognized NS3 simulator, which is a discrete-event network simulator for the i=Internet systems. Second, using the popular machine learning algorithms, I validate my scheme in a real attack scenario with the standard DoS attacks dataset and observe the accuracy of attack resistance capability.

5.7.1 NS3 simulation study

I undertook a simulation study using the widely popular NS3 (3.33) simulator [137] in order to verify the correctness of the proposed DoS mitigation mechanism.

I simulate a social network with 100 mobile users connecting through a singular cellular tower server and an OSN server. I simulate the network under normal operating conditions, each user is attempting to authenticate independently distributed over the simulation period. The user nodes move randomly in an area containing 150 m^2 centered around the tower. The nodes utilize the 2.4 GHz IEEE 802.11 wi-fi standard to communicate. The specific details about the simulations are summarized in Table 5.8.

For modeling the DoS attack, I model an adversary that captures an authentication request and keeps retransmitting the message. Under normal conditions, this would force the OSN server to reply to each of these messages. This can overburden the server resulting in its ability to operate normally. I simulate two scenarios where an adversary executes the DOS attack. The two simulations differ only in the sense that one implements the proposed mitigation mechanism and the other does not execute the DoS mitigation. I also simulate two more scenarios with and without the proposed mitigation mechanism, where two adversaries execute the DoS attack.

Table 5.8: Simulation parameters

Platform	NS3(3.31) / Ubuntu 20.04 LTS	
No. of users	No. of adversary	Proposed scheme
100	0	NA
	1	Not applied
	2	Applied
Simulation time	1200 sec	

Impact on network throughput

In data transmission, the network throughput is the amount of data moved successfully from one place to another in a given time period. Figure 5.9 provides a perspective on how the DoS attacks affect the throughput and how effective the proposed mitigation scheme is. I calculate the throughput with the expression $(\nu_r \times |\rho|)/T_\delta$, where T_δ, ν_r and $|\rho|$ represent the total time in seconds, the number of received packets and their size, respectively.

I observe the network throughput in presence of a DDoS attacker and plot the results in Figure 5.9. It is evident that there is no significant drop in average throughput, when I implement the proposed DDoS mitigation policy into the system. However, in absence of my proposed protocol, an attacker is successful to keep an adversarial effect on the network throughput.

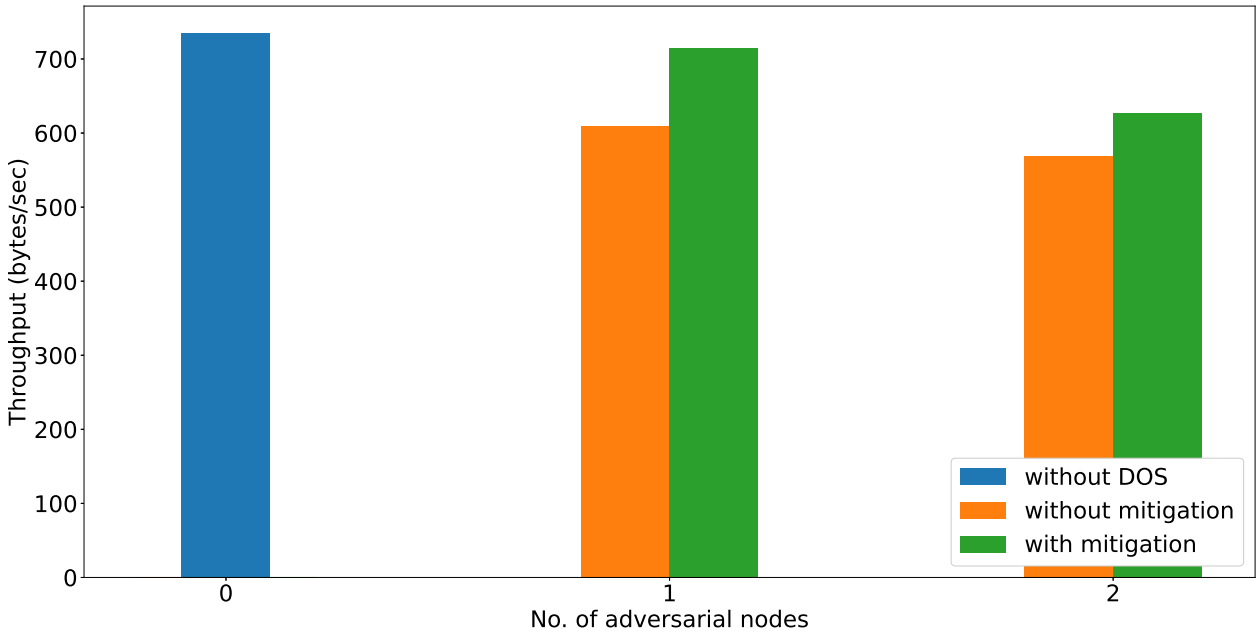

Figure 5.9: Throughput in bytes per second

Impact on end-to-end delay

The end-to-end delay for packets in the network is calculated by the expression $\sum_{i=1}^{\nu_p}(T_{rcv_i} - T_{snd_i})/\nu_p$, where ν_p, T_{rcv_i} and T_{snd_i} represent the total number of packets, the time needed for receiving and sending a data packet i, respectively.

Figure 5.10 plots the average network delays in both scenarios where a DDoS attacker is

present and absent, respectively. A close observation in Figure 5.10 reveals that the mitigation strategy of the proposed scheme does not allow to increase the average network delay, even when the DDoS attacker nodes are present in the network. Consequently, the effectiveness of the proposed scheme towards defending the DDoS attack is obvious.

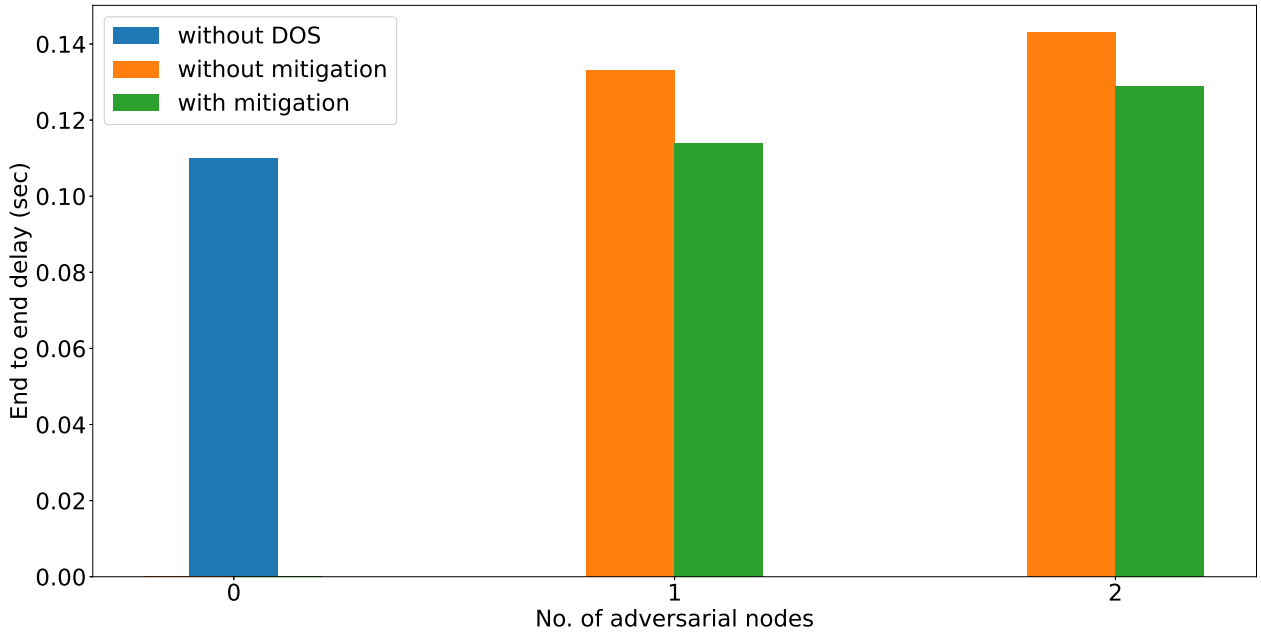

Figure 5.10: End-to-end delay in seconds

5.7.2 Implementation using machine learning methods

In this section, I provide the details of the simulation of my proposed scheme to measure the performance of the scheme using the available real attack and benign datasets. I use three machine-learning algorithms– namely, K-Neighbour Classifier (KNN), Gaussian Naive Bayes (GAUSSIAN NB) classifier, and Multilayer Perceptron classifier (MLP) to demonstrate the performance of the scheme. The key reason for selecting these methods is that these methods provide very precise classification, and the provided precision of the results is also remarkable.

Dataset description

The objective of DoS is to congest the network or server(s) with an overwhelming traffic. Some of the subcategories include "Transmission Control Protocol (TCP) flood", "User Datagram

Table 5.9: Some important features in the dataset [41]

Feature Name	Description
User_ID	Identification number of a user
Flow_Duration	Connection time between source and server
Total_Fwd_Packets	Total number of packets forwarded
Max_Packet_Length	Maximum length of a packet
Min_Packet_Length	Minimum length of a packet
SYN_Flag_Count	Synchronization flag count of a connection
Timestamp	Received timestamp of a message
Label	The ground truth label

Table 5.10: Performance comparison of classifiers

Classifer	TPR(%)	FTP(%)	Accuracy (%)	Precision (%)	AUC (%)
K-Neighbour	99.54	94.52	97.05	99.64	100
Gaussian Naive Bayes	94.52	94.58	95.48	95.88	98.00
MLP	99.54	94.48	96.66	95.88	97.00

Protocol (UDP) flood", "Lightweight Directory Access Protocol (LDAP)", Portmap, "Network Basic Input/Output System (NetBIOS)", etc. I have collected the popular datasets from Canadian Institute for Cybersecurity (CIC DoS dataset 2017) [41], [42], [43]. The datasets are generated by capturing the normal and DoS attack packets separately and pre-processed for testing. Next, I have merged all three datasets for my experiment. The above-said datasets have several classes of DoS attacks, but for the simplicity I have converted them into the set of binary classification of DoS or benign packets. There are 809361 rows and 41 columns in my training dataset, while there are 509 rows and 40 columns in the test dataset. Table 5.9 consists of some important features that are present in my dataset. The last column 'label' is the ground truth (actual label) for the row.

Results and analysis

For assessing the results, I have used the confusion matrix that is applied to evaluate the performance of the proposed scheme. For describing the possible outcomes of classification, the

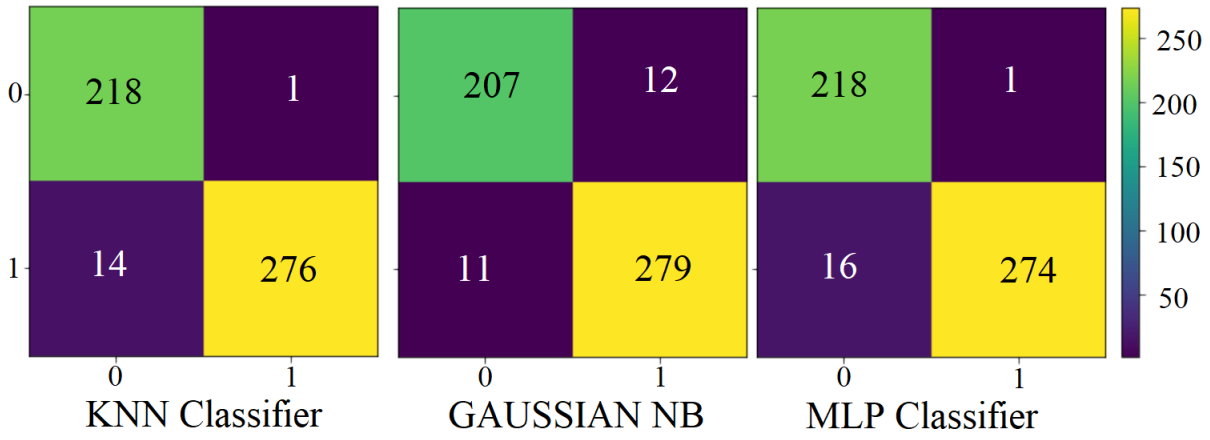

Figure 5.11: Confusion matrix

confusion matrix presented in Figure 5.11 is used. The above-said dataset has several classes of DoS attacks, but for the simplicity I have converted it into the set of binary classification of DoS or benign packets. Therefore, the results are also in binary form of '0' which denotes benign or normal packet, and '1' denoting that there is a chance of DoS attack. The confusion matrix consists of fmy different cases:

- True Negative (TN): A benign labeled packet is correctly classified as a benign or normal packet.
- True Positive (TP): A DoS attack labeled packet is correctly classified as a positive attack packet.
- False Negative (FN): A DoS attack labeled packet is falsely classified as a benign or normal packet.
- False Positive (FP): A benign labeled packet is falsely classified as a benign or normal packet.

Accuracy is the percentage of events that are properly classified as either correct benign packet or DoS attack packet, and it is calculated as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP}.$$

The true positive rate (TPR) is the percentage of events that are correctly classified as DoS attack packet and it is calculated as

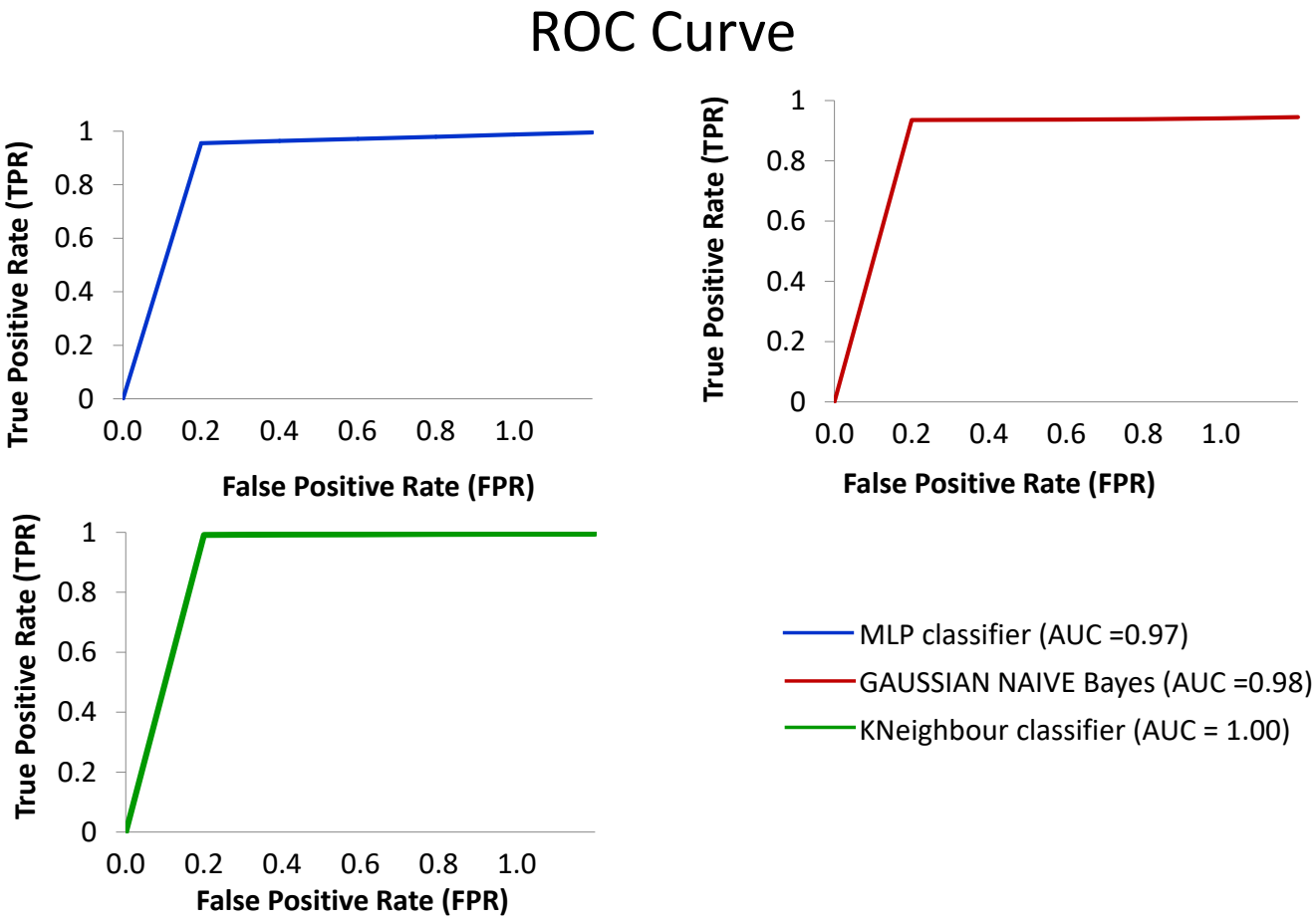

Figure 5.12: The ROC curve and AUC

$$TPR = \frac{TP}{TP+FN}.$$

The false-positive rate (FPR) is the percentage of events incorrectly classified as benign packet and it is calculated as

$$FPR = \frac{FP}{FP+TN}.$$

As I have said that I have used the CIC (Canadian Institute for Cybersecurity) DoS dataset 2017, and converted all the records only in normal benign packets and DoS attack packets. The experiment is evaluated using three machine-learning (ML) techniques – KNN classifier, Gaussian NB classifier, and MLP classifier. According to I scheme, if a server receives three consecutive same packets from the same user within a short period of time T_α, it will be detected as a DoS attack and all the packets from that user will be discarded after a period of resynchronizing time T_p.

I have plotted all the available TPR and FPR in a chart to generate the "receiver operating characteristic curve (ROC curve)", which is given in Figure 5.12 to show the performance of I scheme. Note that an ROC curve shows the "performance of a classification model at all classification thresholds". I have also calculated the area under the ROC curve, known as "Area Under the Curve (AUC)" to measure the accuracy of the algorithm. I have calculated the AUC for all three classification algorithms and demonstrated the aggregate measure of the performance. I have then provided the experimental results given in Table 5.10, which shows that the KNN provides the highest accuracy 97.05%, whereas MLP (96.66%) and Gaussian Naive Bayes (95.48%) are following it.

5.8 Summary

DDoS attack becomes a serious threat to the online social networks. In this chapter, I proposed a DDoS attack resisting authentication protocol for mobile based OSNs. The proposed scheme can efficiently resist the adversary users that repeatedly aim to login attempt to the server. Later on, through key-refilling, the genuine users are allowed to login to the server. Keeping the resource constrained nature of a mobile device, the proposed scheme is made lightweight, effective, and resource-friendly. Various security analysis and simulation results prove that the scheme is secured against different attacks. Low computation and communication overheads, high security and efficient re-synchronization process of the proposed scheme make it appropriate for the real-world application of the mobile OSN.

Regarding the future work of the proposed $PRDoS$ scheme, I want to implement the proposed $PRDoS$ scheme in a real-world testbed environment. Next, I would like to make the scheme to be fine-tuned, if needed, to provide a much better security solution for the real-world OSN.

Chapter 6

ASPA-mOSN Scheme for Resisting Phishing Attacks in mOSN

As discussed in the literature,phishing is an online crime that employs both technical subterfuge and social engineering to steal consumers' personal identities, financial account credentials, and other sensitive information. Phishing attacks are a crucial problem in the current mOSN using some malware or normal keystrokes or posting fraudulent communications with harmful Uniform Resource Locators (URLs), that pretend to come from a reputable source, phishers steal the personal credentials of the users. The problem of designing user authentication protocols for mitigating phishing attacks in OSNs is a challenging research problem. The existing phishing attacks resisting schemes of mOSNs suffer from several security drawbacks.

In this chapter, I have proposed a secure and lightweight cryptography based authentication scheme, called ASPA-mOSN, that provides resistance to phishing and other related attacks in OSNs. According to my proposed scheme, a mobile user and an OSN server register and establish a shared symmetric session key via a cellular tower. At the time of login, users need not be required to provide all credentials, and all credentials will be encrypted with the session key before transmission. Because of this end-to-end encryption, an adversary will have little chance to steal or reveal the credentials of users. Through the security analysis, I have shown that the proposed ASPA-mOSN scheme can resist phishing attacks in the mOSN environment completely.

6.1 Research Contributions

The research contributions, I have achieved in this chapter are discussed below:

- Mediums used for phishing attacks have changed from traditional emails to social media-based phishing. In mobile-based social network phishing, phishers steal the login or other sensitive information of a victim user using a phishing hook. An efficient authentication scheme can provide complete security. Moreover, phishing attack resisting scheme for mOSN must be lightweight, as mobile devices are battery-limited.
- The major research contribution of this scheme is to propose a new robust and efficient secure authentication in the mOSN environment that resists phishing attacks in the mobile-based OSN environment. Through the results of the security analysis, I have shown that the ASPA-mOSN scheme can resist phishing attacks in the mOSN environment completely. It also withstands various other security attacks.

6.2 System Models

In this section, I explain the network model and the security models which are required to explain the design of proposed ASPA-mOSN.

6.2.1 Proposed network model

The proposed network model contains three basic entities:

1. **Mobile user** (U_i) : U_i is a mobile user. It registers into the online social sites and uses the service as an authorized user.
2. **Online Social Network Server** ($OSNServer$): All the required information for an online social network service is stored in the $OSNserver$. It shares information and services at U_i.
3. **Cellular Tower** (CT) : CT helps U_i and $OSNServer$ to communicate with each other. As a coordinator, it also checks whether U_i or $OSNServer$ is genuine or not.

To access the OSN services, a new mobile user U registers itself through a cellular tower CT. A new $OSNserver$ also registers itself to CT. A registered user can only send a login request to an $OSNserver$. A registered $OSNserver$ has the list of users' credentials with it. Therefore after receiving the login request, it checks the authenticity of the user. It sends the response reply to the user. After this authentication process, U will be eligible to access the service provided by the $OSNserver$.

6.2.2 Security model

The proposed scheme adopts the widely-accepted Dolev-Yao threat model (DY model) [55]. An attacker or a malicious user has all the capabilities of executing all potential attacks defined in the classical DY model [55].

Furthermore, I use the stronger threat model, known as the "Canetti and Krawczyk's (CK) adversary model" [33] which is widely regarded as the "current de facto standard model in modeling key-exchange protocols". Here, adversary $\mathcal{A}$ in addition to having all capacities of the DY-threat model, can also compromise ephemeral information like session-specific states and keys. Thus, in the presence of the CK-adversary, a user authentication scheme must be designed such that leakage of ephemeral secrets should have minimal impact on the security of unrelated entities in the authenticated key-exchange scheme [16].

6.3 The Proposed ASPA-mOSN Scheme

This section presents various phases of the our proposed scheme. There are five basic phases of the proposed ASPA-mOSN scheme, namely (1) registration of the different entities, (2) mobile user login, (3) user authentication and key establishment, (4) password change, and (5) dynamic server addition phase. All basic notations used in this scheme are described in Table 6.1.

6.3.1 Registration phase of ASPA-mOSN

A mobile OSN user U_i must need to register into the OSN server OS_k via the cellular tower CT before using the service. U_i also needs to register on the cellular tower CT to be an authorized user. Similarly, OS_k also needs to be registered with the CT. Both the registration phase of the user and the OSN server are separate and can be executed independently and only once. Messages of these phases are communicated using a secure channel. The various message communications in this phase are presented in Figure 6.1.

mOSN User registration phase

To use the OSN service, U_i registers itself to the cellular tower and the OSN server. The user registration process will perform as described in the following steps:

Table 6.1: Notations of the proposed ASPA-mOSN and their meaning.

Symbol	Description
CT	Cellular Tower
U_i	i^{th} Mobile User
OS_k	k^{th} OSN Server
M_{id}	Identity of U_i
K_s	1024-bit master secret key of OSN server
SM_{ik}	1024-bit random number selected by CT for U_i and OS_k pair
$H(\cdot)$	One-way cryptographic hash function
$\|\|, \oplus$	Concatenation, bitwise XOR operations
TS_m	Timestamp generated by U_i
TS_{os}	Timestamp generated by OS_k
R_{mu}	128-bit U_i's random number
R_s	128-bit OS_k's random number
$X \xrightarrow{\langle msg \rangle} Y$	Message (msg) transmission from X to Y
$\triangle T$	Maximum transmission delay

Step MR1:

1. A new user U_i chooses a new identity M_{id} that is not used in the system earlier, password M_{pw} and enters its bio-metrics $\mathcal{B}_i$ into the mobile device.
2. To randomize the parameter, U_i also selects two random numbers p and q that are 128-bit long.

Step MR2:

1. With fuzzy extractor generation method [165], $(\delta_i, \rho_i) = Generation(\mathcal{B}_i)$ is calculated in the mobile device.
2. A masked password $RMB_{pw_i} = (H(M_{id}|| \ H(M_{pw}|| \ \delta_i \ ||p)))\oplus q$ will be calculated next.
3. After that U_i sends a request message $Msg1 = \{M_{id}, RMB_{pw_i}\}$ for registration to the CT through a secure channel.

Step MR3:

1. CT provides a unique master secret key K_s to each OSN server OS_k which is 1024-bit long.

2. After getting the registration request from a new user, CT further chooses a random number (1024-bit) SM_{ik} for each U_i and OS_k pair. CT also computes some parameters, such as $U_{ik} = H(H(M_{id} \oplus SM_{ik}) \,||K_s)$, $D_{ik} = SM_{ik} \oplus RMB_{pw_i}$.

3. CT generates the pseudo-identity of OS_k as $ROS_{id} = H(OS_{id}||K_s)$ also.

Step MR4:

1. Instead of using the original identity, CT anonymize the identity M_{id} of U_i, with a unique temporary identity TM_{id}.

2. Using a secure mechanism CT stores a list of n combinations of $\{TM_{id}, (OS_{id}, D_{ik}, ROS_{id}) \mid 1 \leq k \leq n\}$ in the mobile device of U_i. Moreover, CT collects the serial number SN_m of U_i's mobile device and saves the pair of (M_{id}, SN_m) into its own database.

Step MR5:

1. After receiving all the above said parameters, the following parameters will be computed in the mobile device of U_i:
   $G_i^1 = H(M_{pw}|| \,\delta_i) \oplus p$,
   $G_i^2 = H(M_{pw}|| \,\delta_i) \oplus M_{id}$,
   $G_i^3 = H(M_{id} \,||M_{pw} \,||\delta_i \,||p)$,
   $D'_{ik} = D_{ik} \oplus q = U_{ik} \oplus H(M_{id}|| \,H(M_{pw}|| \,\delta_i \,||p))$,
   $RM_{id} = TM_{id} \oplus H(M_{id} \,||D'_{ik})$, $ROS'_{id} = ROS_{id} \oplus H(\delta_i||p)$ for $1 \leq k \leq n$.

2. At the end, U_i stores $(\rho_i, G_i^1, G_i^2, G_i^3, D'_{ik}, RM_{id}, ROS'_{id})$ into his or her mobile device, and removes D_{ik}, TM_{id} and ROS_{id} from the mobile device.

OSN server registration phase

As an OSN server OS_k stores the social profile with sensitive information and provides the profile access and manipulation services depending on the users' requests, it also must be an authorized entity. Therefore, a new OS_k will join the network by registering itself to CT, executing the following steps:

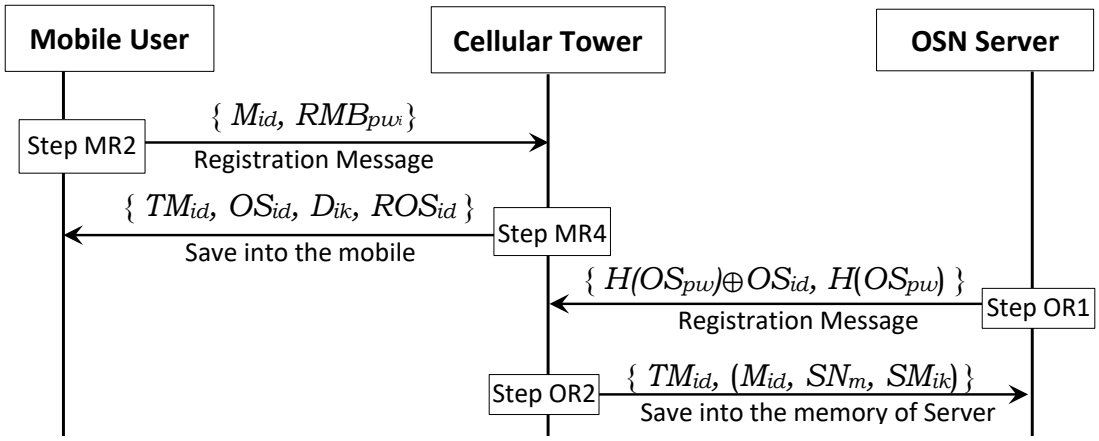

Figure 6.1: Message communications in the registration phase of the proposed scheme

Step OR1:

1. OS_k also chooses a unique identity OS_{id} and password OS_{pw}. OS_k calculates $H(OS_{pw})$.
2. OS_k sends a registration request message $\{H(OS_{pw})\oplus\ OS_{id},\ H(OS_{pw})\}$ via a secure channel to CT.

Step OR2:

1. After receiving the registration request message, CT extracts $OS_{id} = (H(OS_{pw})\oplus OS_{id})\oplus H(OS_{pw})$.
2. CT provides a unique master secret key K_s to OS_k that is 1024-bit long. CT provides a pseudo-identity $ROS_{id} = H(OS_{id}||K_s)$ to OS_k.
3. Using a secure mechanism CT stores the sensible information $\{TM_{id}, (M_{id}, SN_m, SM_{ik})\}$ of all OSN users, in the database of OS_k.
4. Moreover, in the database of OS_k, CT also saves $\{OS_{id}, K_s\}$.

6.3.2 Mobile user login phase of ASPA-mOSN

While a registered mobile user, U_i requests to login into the OSN server OS_k, U_i first sends the login request to CT. CT does further computation. How U_i sends the login message, is explained in this phase. The steps of login, user authentication, and session key establishment phases are presented in Figure 6.2. Figure 6.3 shows the message exchange of this phase.

Login phase	
Mobile User (U_i)	OSN Server (OS_k)
Input M_{pw} and $\mathcal{B}'_i$. Compute $\delta_i = Reproduction(\mathcal{B}'_i, \rho_i)$, $p' = G_i^1 \oplus H(M_{pw}\|\|\delta_i)$, $M_{id} = G_i^2 \oplus H(M_{pw}\|\|\delta_i)$. Verify if $G_i^3 = H(M_{id}\|\|M_{pw}\|\|\delta_i\|\|p')$? If verification holds, compute $RMB_{pw_i} = H(M_{id}\|\| H(M_{pw}\|\| \delta_i \|\|p'))$, $U_{ik} = D'_{ik} \oplus RMB_{pw_i}$. Select random number R_{mu} and timestamp TS_m. Compute $W_u = U_{ik} \oplus R_{mu} \oplus TS_m \oplus H(OS_{id})$, $H_{u_1} = H(M_{id}\|\|W_u\|\|R_{mu}\|\|TS_m)$, $TM_{id} = RM_{id} \oplus H(M_{id}\|\|D'_{ik})$, $ROS_{id} = ROS'_{id} \oplus H(\delta_i\|\|p')$, $TM^*_{id} = TM_{id} \oplus H(ROS_{id}\|\|TS_m)$. $\xrightarrow[\text{public channel}]{\{TM^*_{id}, W_u, H_{u_1}, TS_m\}}$	
Authentication phase	
	Verify if $\|TS^*_m - TS_m\| \leq \triangle T$? If verification holds, compute $ROS_{id} = H(OS_{id}\|\|K_s)$, $TM_{id} = TM^*_{id} \oplus H(ROS_{id}\|\|TS_m)$. Search record $\langle M_{id}, TM_{id}, SM_{ik}\rangle$ from its database. Compute $V_{ik} = H(H(M_{id} \oplus SM_{ik})\|\|K_s)$, $m_1 = W_u \oplus TS_m \oplus H(OS_{id}) \oplus V_{ik}$ $= R_{mu}$,as $U_{ik} = V_{ik} = H(H(M_{id} \oplus SM_{ik})\|\|K_s)$, Compute $H_{s_1} = H(M_{id}\ \|W_u\ \|m_1\ \|TS_m)$.
Verify if $\|TS^*_{os} - TS_{os}\| \leq \triangle T$?	Verify if $H_{s_1} = H_{u_1}$?
After successful verification, compute $m_2 = W_s \oplus TS_{os} \oplus M_{id} \oplus U_{ik} = R_s$ as $U_{ik} = V_{ik}$	If verification holds, generate R_s, compute $W_s = V_{ik} \oplus R_s \oplus TS_{os} \oplus M_{id}$ $SKEY_{os} = H(M_{id}\|\|OS_{id}\|\|V_{ik}\|\|m_1\|\|R_s\|\|TS_m\|\|TS_{os})$,
$SKEY_{mu} = H(M_{id}\|\|OS_{id}\|\|U_{ik}\|\|R_{mu}\|\|m_2\|\|TS_m\|\|TS_{os})$,	$\xleftarrow[\text{public channel}]{\{W_s, H_{s_2}, TS_{os}\}}$
$H_{u_2} = H(M_{id}\ \|R_{mu}\ \|m_2\ \|TS_m\ \|TS_{os}\ \|SKEY_{mu})$.	
Verify if $H_{u_2} = H_{s_2}$? If verification holds,	
Store session key $SKEY_{mu}(= SKEY_{os})$.	Store session key $SKEY_{os}(= SKEY_{mu})$.

Figure 6.2: Login and authentication phase of the proposed ASPA-mOSN

The steps are as follows:

Step ML1:

1. At the time of login U_i does not enter his or her user id. The user enters only its password M_{pw} and its own bio-metrics $\mathcal{B}'_i$ into the mobile device.

2. With the stored variable ρ_i and reproduction method of the fuzzy extractor, U_i computes $\delta_i = Reproduction(\mathcal{B}'_i, \rho_i)$.

Step ML2:

1. U_i regenerates $p = G_i^1 \oplus H(M_{pw}\ \|\delta_i)$ and $M_{id} = G_i^2 \oplus H(M_{pw}\ \|\delta_i)$ with the help of saved parameters G_i^1 and G_i^2.

2. U_i verifies whether $G_i^3 = H(M_{id}\|\|M_{pw}\|\|\delta_i\|\|p)$ is hold or not, by calculating $H(M_{id}\ \|M_{pw}\ \|\delta_i\ \|p)$. Execution stops if verification fails, otherwise proceed further.

Step ML3:

1. U_i computes a masked password, $RMB_{pw_i} = H(M_{id}|| \ H(M_{pw}|| \ \delta_i \ ||p'))$.

2. U_i further masked the password using the stored parameter D'_{ik} and generates $U_{ik} = D'_{ik} \oplus \ RMB_{pw_i}$.

Step ML4:

1. U_i selects a random number R_{mu} which is 128 bit long and generating current timestamp TS_m computes: $W_u = U_{ik} \oplus R_{mu} \oplus TS_m \oplus H(OS_{id}) = H(H(M_{id} \oplus SM_{ik})||K_s) \oplus R_{mu} \oplus TS_m \oplus H(OS_{id})$, $H_{u_1} = H(M'_{id}||W_u||R_{mu}||TS_m)$, $TM_{id} = RM_{id} \oplus H(M'_{id}||D'_{ik})$, $ROS_{id} = ROS'_{id} \oplus H(\delta_i||p')$, $TM^*_{id} = TM_{id} \oplus H(ROS_{id}||TS_m)$.

2. At the end, U_i sends a login request $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ to CT through a public channel.

Step ML5:

1. After getting the message from U_i, CT calculates $H(ROS_{id}||TS_m)$, and extracts $TM_{id} = TM^*_{id} \oplus H(ROS_{id}||TS_m)$.

2. Next, CT checks if the user U_i is a registered user or not, by searching its own database.

3. CT just forwards Msg_l to the OSN server OS_k while TM_{id} exists in the database.

6.3.3 User authentication and key establishment phase of ASPA-mOSN

When an OSN server OS_k received a login request from a registered user U_i, how OS_k authenticates the user, is explained in this phase. Moreover, a secret session key for that session also will establish for message transmission. The following steps will explain the detailed procedure:

Step AK1:

1. After receiving the U_i's message Msg_l, OS_k first matches the current timestamp TS^*_m with the timestamp received in the Msg_l. It checks whether $|TS^*_m - TS_m| \leq \triangle T$, is true or not, where $\triangle T$ is the maximum acceptable transmission delay.

2. Msg_l will be discarded immediately if the verification does not succeed, else, the execution will proceed to the next step.

Step AK2:

1. OS_k generates its own pseudo-identity $ROS_{id} = H(OS_{id}||K_s)$.

2. Next, after obtaining $TM_{id} = TM^*_{id} \oplus H(ROS_{id} ||TS_m)$, OS_k search the records $\langle M_{id}, SM_{ik} \rangle$ that is related with TM_{id} from its database.

Step AK3:

1. OS_k calculates the following parameters using its own id OS_{id} and master key K_s, $V_{ik} = H(H(M_{id} \oplus SM_{ik})||K_s)$, $m_1 = W_u \oplus TS_m \oplus H(OS_{id}) \oplus V_{ik} = U_{ik} \oplus R_{mu} \oplus TS_m \oplus H(OS_{id}) \oplus TS_m \oplus H(OS_{id}) \oplus V_{ik} = R_{mu} = R_{mu}$, as $U_{ik} = V_{ik} = H(H(M_{id} \oplus SM_{ik})||K_s)$.

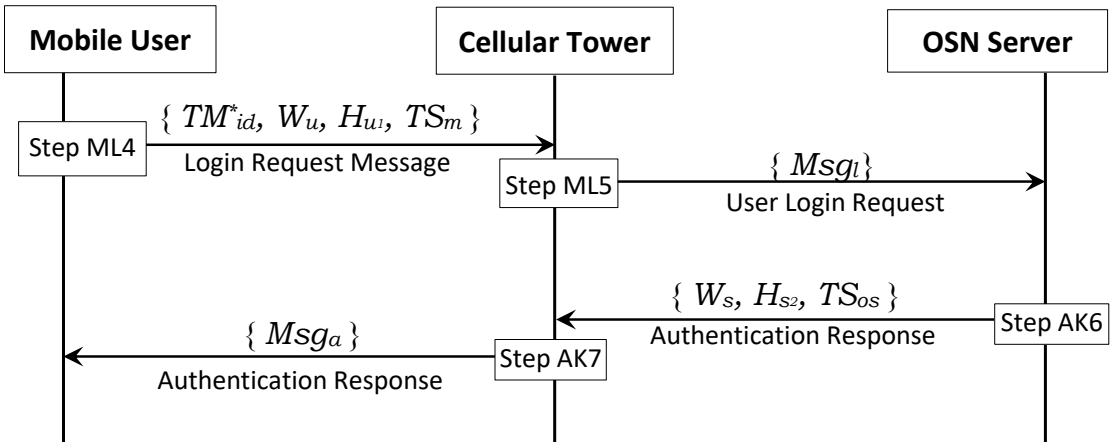

Figure 6.3: Message communications in the login and authentication phase of the proposed scheme

Step AK4:

1. After computing the hash value $H_{s_1} = H(M_{id} ||W_u ||m_1 ||TS_m)$, OS_k checks whether the hash value H_{u_1} received from U_i and derived hash value $H_{s_1} \stackrel{?}{=} H_{u_1}$ or not.

2. Execution terminates with the failure of the verification. Otherwise, OS_k accepts U_i's request and the execution continues further.

Step AK5:

1. OS_k saves the record $\langle M_{id}, R_{mu}, TS_m \rangle$ into its own database.
2. OS_k chooses a random number R_s that is 128 bits long. Next, using R_s the current timestamp TS_{os}, OS_k produces $W_s = V_{ik} \oplus R_s \oplus TS_{os} \oplus M_{id}$.

Step AK6:

1. Finally, OS_k produces the mutually shared session key $SKEY_{os} = H(M_{id} \,||OS_{id} \,||V_{ik} \,||m_1 \,||R_s \,||TS_m \,||TS_{os})$. OS_k will use this key to encrypt the messages before transmitting them to U_i.
2. Additionally, OS_k generates a hash value $H_{s_2} = H(M_{id} \,||m_1 \,||R_s \,||TS_m \,||TS_{os} \,||SKEY_{os})$.
3. The authentication response message $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ will then send to CT via a public channel.

Step AK7:

1. After getting the message, CT checks whether OS_k is a registered server or not.
2. CT forwards message Msg_a, if it finds the record containing OS_k from its database. Otherwise, discard the message.

Step AK8:

1. U_i verifies the difference between messages received timestamp, TS^*_{os}, and the timestamp received with Msg_a, $|TS^*_{os} - TS_{os}| \leq \triangle T$ as soon as it receives the response message, Msg_a from OS_k. With the successful verification, execution proceeds further.
2. U_i then generates $m_2 = W_s \oplus TS_{os} \oplus M_{id} \oplus U_{ik} = V_{ik} \oplus R_s \oplus TS_{os} \oplus M_{id} \oplus TS_{os} \oplus M_{id} \oplus U_{ik} = R_s$ as $U_{ik} = V_{ik} = H(H(M_{id} \oplus SM_{ik}) || K_s)$.

Step AK9:

1. Next U_i regenerates the shared session key $SKEY_{mu} = H(M_{id} \,||OS_{id} \,||U_{ik} \,||R_{mu} \,||m_2 \,||TS_m \,||TS_{os})$ by using the calculated parameter m_2 and received timestamp TS_{os}.
2. U_i also calculates a new hash value $H_{u_2} = H(M_{id} \,||R_{mu} \,||m_2 \,||TS_m \,||TS_{os} \,||SKEY_{mu})$ and verifies if $H_{u_2} \stackrel{?}{=} H_{s_2}$ or not.

3. The shared session key $SKEY_{mu}$ (= $SKEY_{os}$) is mutually established and confirmed while the verification succeeds.

In that session, for message transmissions with OS_k, this key will be used.

6.3.4 Password change phase

As the OSN servers or cellular towers do not save the password of any users and the password is only saved locally in the mobile device, the password change phase in this scheme happens inside the device only. The old password M_{pw} of the user U_i is replaced by the new password M'_{pw}. To change the password, the following steps are needed to be executed:

Step PC1:

1. In the password change phase also, the user U_i enters only his or her old password M_{pw} along with biometrics $\mathcal{B}'_i$.

2. Using the reproduction procedure of fuzzy extractor and the stored variable ρ_i, U_i computes $\delta_i = Reproduction(\mathcal{B}'_i, \rho_i)$.

3. Moreover, using the stored parameters G_i^1 and G_i^2, U_i generates $p' = G_i^1 \oplus H(M_{pw} ||\delta_i)$ and $M_{id} = G_i^2 \oplus H(M_{pw} ||\delta_i)$.

Step PC2:

1. After computing $H(M_{id} ||M_{pw} ||\delta_i ||p')$, U_i checks whether $G_i^3 = H(M'_{id}||M_{pw}||\delta_i||p')$ is true or false. Only after successful verification, the execution moves to the next step.

2. Whenever the system asks for the new password, U_i will enter the new password M'_{pw}. With the change of the old password, the stored parameters which are embedded with the password will become invalid.

Step PC3:

1. For computation of the new value for the stored parameters, U_i calculates $TM_{id} = RM_{id} \oplus H(M_{id}|| D'_{ik})$.

2. Next, U_i computes the new values of $G_i^{1*} = H(M'_{pw} ||\delta_i) \oplus p'$, $G_i^{2*} = H(M'_{pw} ||\delta_i ||p') \oplus M_{id}$, $G_i^{3*} = H(M_{id} ||M'_{pw} ||\delta_i ||p')$, $D^*_{ik} = D'_{ik} \oplus H(M_{id}|| H(M_{pw}|| \delta_i ||p')) \oplus H(M_{id}|| H(M'_{pw}|| \delta_i ||p'))$, $RM^*_{id} = TM_{id} \oplus H(M_{id} ||D^*_{ik})$.

Step PC4:

1. Finally, the old values of the stored parameters of U_i's mobile device will update with $G_i^1 = G_i^{1*}$, $G_i^2 = G_i^{2*}$, $G_i^3 = G_i^{3*}$, $D'_{ik} = D^*_{ik}$ and $RM_{id} = RM^*_{id}$. All other temporary variables will be deleted from its memory.

6.3.5 Dynamic server addition phase

For efficient operation of Online social networks, multiple servers will be required. Therefore, any efficient authentication scheme should have the provision to scale up the network by adding new servers to the existing system. This phase will explain how to add a new OSN server into an existing system.

Step AS1:

1. When a new OSN server OS_k will be installed into the network, first, it must be registered into the cellular tower CT.

2. CT will provide a unique identity OS_{id} and a 1024-bit long unique master secret key K_s.

3. CT further computes a list of secret numbers (1024-bit) SM_{ik} for each U_i and OS_k pair and stores the sensible information $\{TM_{id}, (M_{id}, SN_m, SM_{ik})\}$ of all OSN users and $\{OS_{id}, K_s\}$ in the database of OS_k.

Step AS2:

1. After installation, CT notifies the user U_i about the addition of OS_k. And U_i does not require any additional information to save into the mobile device of U_i.

Remark 6.1. *For adding a new server into an existing OSN, no cryptographic operation will be required to perform. Only some random numbers need to compute. Hence, no extra computational overhead will be required. At the time of addition, a new server needs to store* $\{OS_{id}, K_s\}$*, and* $\{TM_{id}, (M_{id}, SN_m, SM_{ik})\}$ *of all OSN users which must be sent by a CT through a secure channel. As it is a one-time process for a new server, I am not considering this cost for the calculation of communication cost.*

6.4 Formal Security Analysis Using ROR model

In this section, I have illustrated the security analysis of the proposed ASPA-mOSN scheme formally through the Random Oracle Model (ROR) [186] to demonstrate that the proposed protocol is secure against an attacker for disclosing the session key.

Participants: In my proposed protocol, the authentication is done between a user U_i and a server OS_k. They are two participants $\mathcal{P}^t$ of the protocol. Using $\mathcal{P}_{U_i}^{t_1}$ and $\mathcal{P}_{OS_k}^{t_2}$ I represent the instances of U_i and OS_k respectively, in random oracles.

An attacker $\mathcal{A}$ can communicate with the participants' $\mathcal{P}_t$ of the protocol via some oracle queries, that model the ability of the attacker in a real attack.

According to the ROR model, at the time of execution, $\mathcal{A}$ may create several concurrent instances of a participant $\mathcal{P}_t$ (U_i or OS_k) and various queries. I simulate different security attacks and all possible oracle queries that can be required to proof of the proposed ASPA-mOSN scheme $\mathscr{P}$ formally.

Execute(U_i, OS_k)**:** Some passive attacks may be modeled using this query, where $\mathcal{A}$ listen a message m is communicated between U_i and OS_k in an honest execution of the protocol.

Send$(U_i/OS_k, m_1)$**:** Active attacks are modeled using this query. $\mathcal{A}$ may intercept a message m_1 and either generate a new one, or revise it or forward it to $\mathcal{P}^t$, and $\mathcal{P}^t$ responds to $\mathcal{A}$ as per the rules of the protocol.

Reveal$(\mathcal{P}^t)$**:** This query is asked by $\mathcal{A}$ to disclose the existing session key $SKEY$ defined between $\mathcal{P}^t$ and its other partner.

Test$(\mathcal{P}^t)$**:** $\mathcal{A}$ may ask this query for the fresh session key $SKEY$ to $\mathcal{P}^t$ and if the session key is not defined, gets a *null* value. Else, $\mathcal{P}^t$ return the session key when b = 1 or if b = 0, it returns a random key of the same size, where b is an unbiased flipped coin.

Corrupt(U_i, a)**:** This query is used to model the ability of $\mathcal{A}$ to acquire the secret

credentials, such as a password or biometric of a participant U_i, according to the value of a, thus corrupting the protocol.

For proving the session key security of the proposed protocol, I consider the following semantic security notion of the ROR model and prove Theorem 6.1.

Theorem 6.1. *Let $\mathcal{A}$ is an adversary running within polynomial time $t_{\mathcal{A}}$. Suppose to disclose the semantic security of the proposed ASPA-mOSN protocol $\mathscr{P}$, $\mathcal{A}$ generates Hash, Send and Execute oracle queries at most q_H, q_s and q_e times, respectively. Hence, the advantage function of $\mathcal{A}$ is:*

$$Adv_{\mathscr{P}}^{ASPA-mOSN} \leq \frac{q_H^2 + 18q_H}{2^{l_H}} + \frac{(q_s+q_e)^2 + 4q_s}{2^{l_R}} + 2\max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\}$$

where q_H, q_s, q_e are total number of hash, send, execute oracle queries, l_H, l_R, l_b are string lengths of the outputs of hash oracle queries, random oracle queries, and user biometrics, ε_{bm} is the probability of false-positive in biometrics, and $\mathcal{D}$ is A finite Password space dictionary with size $|\mathcal{D}|$.

Proof. To prove Theorem 6.1, I have defined game G_i with set size five, where $(i = 0, 1, 2, 3, 4)$. To get the correct session key, an adversary $\mathcal{A}$ may try to predict bit b in *Test* oracle query successfully. This event, S_i refers to the success probability of the event is defined by $P_r[S_i]$. The details of these games are described below.

Game G_0: This game is referred to as the original attack game, and I assume this initial game is identical to the actual protocol in random oracles. Therefore, I get,

$$Adv_{\mathscr{P}}^{ASPA-mOSN} = |2P_r[S_0] - 1|. \tag{6.1}$$

Game G_1: In game G_1 all the oracle queries, such as *Test*, *Corrupt*, *Reveal*, *Execute*, and *Send* except *Hash* are simulated. The working procedure of *Execute*, and *Send* queries for the proposed ASPA-mOSN scheme are simulated in Table 6.2. Three lists are generated to record the output of various oracle queries: L_H , L_r, and L_t. The simulation of the game G_1 is considered to be indistinguishable from game G_0. Therefore, I have,

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

Table 6.2: Simulation of oracle query execute & send

Send simulation query performs as follows. **(a)** For a $Send(U_i, \textbf{start})$ query, U_i gives the following response. Compute TM^*_{id}, W_u, H_{u_1}, TS_m as in Figure 6.2. Output $Msg_l = \langle TM^*_{id}, W_u, H_{u_1}, TS_m \rangle$. **(b)** Let OS_k be the target state. For a $Send(OS_k, \langle TM^*_{id}, W_u, H_{u_1}, TS_m \rangle)$ query,OS_k gives the following response. Verify whether $\lvert TS^*_m - TS_m \rvert \leq \triangle T$ and upon the successful verification, compute W_s and H_{s_1} and verifies value of $H_{s_1} \stackrel{?}{=} H_{u_1}$ or not. A mismatch rejects the session. Further, OS_k computes H_{s_2} and $SKEY_{os}$, and output $Msg_a = \langle W_s, H_{s_2}, TS_{os} \rangle$. **(c)** U_i answers $Send(U_i, \langle W_s, H_{s_2}, TS_{os} \rangle)$ query as follows. Verify whether $\lvert TS^*_{os} - TS_{os} \rvert \leq \triangle T$, A mismatch leads to termination of the session. Compute H_{u_2} and then verify if $H_{u_2} \stackrel{?}{=} H_{s_2}$. A mismatch leads to termination of the session. Otherwise, establish $SKEY_{mu}$ as the session key. Finally, both U_i and OS_k accept the successful termination of the session.
Simulation of *Execute* (U_i, S) query occurs in succession with simulation of *Send* queries as shown below. Compute W_u, H_{u_1} as given in Figure 6.2. U_i sends message Msg_l to OS_k, where $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ Computing W_s and H_{s_1}, OS_k sends authentication message Msg_a to U_i, where $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ Note that $\langle TM^*_{id}, W_u, H_{u_1}, TS_m \rangle \leftarrow Send(U_i, \textbf{start})$, $\langle W_s, H_{s_2}, TS_{os} \rangle \leftarrow Send(OS_k, \langle TM^*_{id}, W_u, H_{u_1}, TS_m \rangle)$. Finally, Msg_l and Msg_a are returned.

Game G_2: The collision probability with $Hash$ oracle query results and random numbers for all the transmitted messages in login and authentication phases of our ASPA-mOSN scheme $\mathscr{P}$ are simulated in this game. The login message $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ and authentication $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$, contain random numbers R_{mu} and R_s that are session-specific. Therefore, the collision probability is at most $\frac{(q_s+q_e)^2}{2^{l_R+1}}$. Depending on the birthday paradox,

maximum collision probability of the *hash* query can be $\frac{q_H^2}{2^{l_H+1}}$. Hence, I get,

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

Game G_3: As the collision probability of the oracle query $Hash$ is already being described in G_2, I have tried to simulate this game that $\mathcal{A}$ is trying to predict the original message using remaining other oracle queries. I have to consider the following two cases according to the message transmission of login and authentication phases:

Case 1: The login message $Send(OS_k, Msg_l)$ query is considered and tried to respond to it. So, the hash value $H_{u_1} = H(M'_{id}||W_u||R_{mu}||TS_m) \in L_{\mathcal{A}}$ must hold, otherwise the session will terminate. It has the probability at most $\frac{q_H}{2^{l_H}}$. Next, $\mathcal{A}$ needs to search at least four hash values as given in Figure 6.2 for launching an attack successfully. Hence, the total calculated probability is at most $\frac{4q_H}{2^{l_H}}$. Finally, for random variable RN_u in message $Msg_l \in L_T$, needs the maximum probability $\frac{q_s}{2^{l_R}}$.

Case 2: At the time of sending authentication message Msg_a from OS_k to U_i, if $\mathcal{A}$ execute the oracle query $Send(U_i, Msg_a)$ and as $H_{s_2} \in L_{\mathcal{A}}$, it must hold with probability $\frac{q_H}{2^{l_H}}$. Moreover, before sending the message Msg_a, OS_k computes five H hash operations for calculating ROS_{id}, TM_{id}, V_{ik}, hash value H_{s_1} and the session key $SKEY_{os}$ as given in Figure 6.2 with the total probability of $\frac{5q_H}{2^{l_H}}$. For transcript message with random number R_s, $Msg_a \in L_T$ needs the maximum probability $\frac{q_s}{2^{l_R}}$.

Considering both cases, I obtain,

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

Game G_4: I have considered all the guessing attacks exploited by adversary $\mathcal{A}$ in this game. $\mathcal{A}$ can execute *Corrupt* oracle query to capture both password and biometrics for our protocol $\mathscr{P}$.

Case 1: For guessing the password online, $\mathcal{A}$ can run query $Corrupt(U_i, 1)$. $\mathcal{A}$ may select a password quickly from dictionary $\mathcal{D}$ and then executes *Send(*OS_k*,*M_1*)* query with time q_s, having probability $\frac{q_s}{|\mathcal{D}|}$.

Case 2: Similarly, for guessing biometrics online, $\mathcal{A}$ can execute query $Corrupt(U_i, 2)$. The guessing probability will be at most $\frac{1}{2^{l_b}}$, as l_b is the string length of extracted biometric. Furthermore, the probability of the unintentional guessing of "false positive" case is ε_{bm}. In general, it can be observed that for fingerprints, ε_{bm}.

Considering both cases and the games G_4 and G_3 are identical, I get,

$$|P_r[S_4] - P_r[S_3]| \leq \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} \tag{6.5}$$

Considering all the above games, $\mathcal{A}$ has only one option left for guessing the correct bit b, I get

$$|P_r[S_4] = \frac{1}{2} \tag{6.6}$$

Using the triangular inequality law, I obtain the following:

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

Using Equations (6.1)-(6.7), I get,

$$\begin{aligned} \frac{1}{2} Adv_{\mathscr{P}}^{PPAP} &= |P_r[S_0] - \frac{1}{2}| \\ &\leq \frac{(q_s + q_e)^2}{2^{l_R+1}} + \frac{q_H^2}{2^{l_H+1}} + \frac{2q_s}{2^{l_R}} + \frac{9q_H}{2^{l_H}} \\ &\quad + \max\{q_s(\frac{1}{|\mathcal{D}|}, \frac{1}{2^{l_b}}, \varepsilon_{bm})\} \end{aligned} \tag{6.8}$$

Finally, multiplying by 2 in both sides of Equation (6.8) and rearranging all the terms, I achieve the desired result. Therefore, the theorem is proved. □

6.5 Authentication Proof of ASPA-mOSN Using BAN Logic

Using BAN logic the mutual authentication between two communicating participants can be described in a network. I demonstrate that the proposed ASPA-mOSN scheme can achieve the authentication goals with the broadly-used BAN logic. The basic notations of BAN logic and their meanings are presented in Table 6.3.

The logical postulates for the BAN logic are described with a set of rules (laws) as written below [31], [172].

Table 6.3: Notations of BAN logic and their meaning

Notations	Description
$A \mid\equiv S$	A believes that the statement S is true
$A \lhd S$	A can see the statement S
$\#(S)$	Formula S is considered as fresh
$A \mid\sim S$	A said the statement S once
$A \Rightarrow S$	A keeps jurisdiction over the statement S
$\langle S \rangle_K$	Formula S is combined with the formula K
$A \overset{K}{\longleftrightarrow} B$	Only A and B know the value of the key K and it is used for communication between them
$A \overset{S}{\rightleftharpoons} B$	Only A and B know the secret statement S. Principals trusted by A & B may know S
$SKEY$	Current session key

- Rule of Message Meaning (RMM):
  $\frac{A \mid\equiv B \overset{K}{\rightleftharpoons} A, A \lhd \langle S \rangle_K}{A \mid\equiv B \mid\sim S}$.

- Rule of Freshness Conjuncatenation Rule (RFC):
  $\frac{A \mid\equiv \#(S)}{A \mid\equiv \#(S,T)}$.

- Rule of Nonce Verification (RNV):
  $\frac{A \mid\equiv \#(S), A \mid\equiv B \mid\sim S}{A \mid\equiv B \mid\equiv S}$.

- Rule of Jurisdiction (RJ):
  $\frac{A \mid\equiv B \Rightarrow S, A \mid\equiv B \mid\equiv S}{A \mid\equiv S}$.

- Rule of Addition (RA):
  $\frac{A \mid\equiv (S,T)}{A \mid\equiv S}$, $\frac{A \lhd (S,T)}{A \lhd S}$, $\frac{A \mid\equiv B \sim (S,T)}{A \mid\equiv B \sim S}$.

The proposed ASPA-mOSN scheme must fulfill the following two goals, for completing the authentication proof:

Goal No. 1. $U_i \mid\equiv (U_i \overset{SKEY}{\longleftrightarrow} OS_k)$.

Goal No. 2. $OS_k \mid\equiv (U_i \overset{SKEY}{\longleftrightarrow} OS_k)$.

The common message types in the proposed ASPA-mOSN scheme are as follows:

Type 1. $U_i \rightarrow OS_k$: $\{TM^*_{id}, H(H(M_{id} \oplus SM_{ik}) || K_s) \oplus R_{mu} \oplus TS_m \oplus H(OS_{id}), TS_m, H_{u_1}\}$.

Type 2. $OS_k \rightarrow U_i : \{V_{ik} \oplus R_s \oplus TS_{os} \oplus M_{id}, TS_{os}, H_{s_2}\}$.

The idealized representation of the above said messages are stated as follows:

Msg Type 1. $U_i \rightarrow OS_k$: $\{TM_{id}, TS_m, \langle M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id})\rangle_{K_s}, H_{u_1}\}$.

Msg Type 2. $OS_k \rightarrow U_i$: $\{TS_{os}, \langle R_s, TS_{os}, M_{id}\rangle_{K_s}, H_{s_2}\}$.

The starting of the authentication proof for the proposed ASPA-mOSN scheme begins with some basic assumptions as described below:

- A1: $U_i \mid\equiv \#(TS_{os})$;
- A2: $OS_k \mid\equiv \#(TS_m)$;
- A3: $U_i \mid\equiv (U_i \overset{U_{ik}}{\rightleftharpoons} OS_k)$;
- A4: $OS_k \mid\equiv (U_i \overset{U_{ik}}{\rightleftharpoons} OS_k)$;
- A5: $U_i \mid\equiv OS_k \Rightarrow (OS_{id}, R_s, TS_{os})$;
- A6: $OS_k \mid\equiv U_i \Rightarrow (M_{id}, R_{mu}, TS_m)$;
- A7: $U_i \mid\equiv TS_m$;
- A8: $U_i \mid\equiv R_{mu}$;
- A9: $U_i \mid\equiv M_{id}$;
- A10: $U_i \mid\equiv OS_{id}$;
- A11: $OS_k \mid\equiv TS_{os}$;
- A12: $OS_k \mid\equiv R_s$;
- A13: $OS_k \mid\equiv OS_{id}$.

Using idealized representation, the basic assumptions, and the logical postulates, I have demonstrated that both the goals **Goal No. 1** and **Goal No. 2** are achieved. As per the message type 1, I get,

- S_1: $OS_k \lhd \{M_{id}, TS_m, \langle M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id})\rangle_{K_s}, H_{u_1}\}$.

```
(*—-Phishing attack prevention Scheme for mOSNs—-*)
(* ————— channels —————*)
free ch_pb: channel. (* public channel *)
free ch_pv: channel [private]. (* private channel *)
(* ————— shared keys of OSN Server and Mobile User ————— *)
free SKEYus:bitstring [private].(* the session key of Mobile user *)
free SKEYom:bitstring [private]. (* the session key of OSN server *)
(* ————— Servers secret key ————— *)
free Ks:bitstring [private].
free SMik:bitstring [private].
(* ————— constants parameters ————— *)
free OSid:bitstring [private].
free ROSid:bitstring [private].
free Mid:bitstring [private].
free TMid:bitstring [private].
free Mpw:bitstring [private].
const Bi:bitstring [private].
(* ————— functions and equations ————— *)
fun h(bitstring):bitstring. (* hash function *)
fun BE(bitstring):bitstring. (* Biometric Fuzzy extractor function *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring. (* string concatenation *)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(* ————— aims for verification ————— *)
query attacker(SKEYus).
query attacker(SKEYom).
query Mid:bitstring; inj-event(MUsrAuthen(Mid)) ==⟩ inj-event(MUsrStart(Mid)).
(* ————— event ————— *)
event MUsrStart(bitstring). (* Mobile User starts authentication *)
event MUsrAuthen(bitstring). (* Mobile User is authenticated *)
```

Figure 6.4: The ProVerif code for declaration of channels, variables, events

- S_2: Applying the inference rule RA, I obtain, $OS_k \lhd \langle M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id})\rangle_{K_s}$.
- S_3: Using A4 and rule RMM, I obtain, $OS_k \mid\equiv U_i \mid\sim (M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id}))$.
- S_4: Applying A2 and rule RFC, I obtain, $OS_k \mid\equiv \#(M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id}))$.
- S_5: According to rule RNV, there exist, $OS_k \mid\equiv U_i \mid\equiv (M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id}))$.
- S_6: Applying A6 and rule RJ, I obtain, $OS_k \mid\equiv (M_{id}, SM_{ik}, R_{mu}, TS_m, H(OS_{id}))$.
- S_7: Using S_6 and rule RA, I have, $OS_k \mid\equiv R_{mu}$, $OS_k \mid\equiv TS_m$, $OS_k \mid\equiv M_{id}$.
- S_8: According to A11, A12, A13, I obtain, $OS_k \mid\equiv OS_{id}$, $OS_k \mid\equiv TS_{os}$ and $OS_k \mid\equiv R_s$.
- S_9: Since $SKEY_{os} = H(M_{id} \mid\mid OS_{id} \mid\mid V_{ik} \mid\mid m_1 \mid\mid R_s \mid\mid TS_m \mid\mid TS_{os})$ and merging the results of Steps S_7 and S_8, I get $OS_k \mid\equiv (U_i \overset{SKEY_{os}}{\longleftrightarrow} OS_k)$. **(Goal No. 2)**
- S_{10}: With the help of message type 2 and rule RA, I get, $U_i \lhd \langle R_s, TS_{os}\rangle_{K_s}$.
- S_{11}: Using to A3 and rule RMM, I obtain, $U_i \mid\equiv OS_k \mid\sim (R_s, TS_{os})$.
- S_{12}: Applying A1 and rule RFC, I get, $U_i \mid\equiv \#(R_s, TS_{os})$.
- S_{13}: Applying rule RNV, I get, $U_i \mid\equiv OS_k \mid\equiv (R_s, TS_{os})$.
- S_{14}: With the help of A5 and rule RJ, I get, $U_i \mid\equiv (R_s, TS_{os})$.
- S_{15}: Using S_{14} and rule RA, I get, $U_i \mid\equiv R_s$, $U_i \mid\equiv TS_{os}$.
- S_{16}: Depending on A7-A10, I have, $U_i \mid\equiv M_{id}$, $U_i \mid\equiv OS_{id}$, $U_i \mid\equiv TS_m$, $U_i \mid\equiv R_s$.
- S_{17}: The output of steps S_{15} and S_{16} can show $U_i \mid\equiv (U_i \overset{SKEY_{os}}{\longleftrightarrow} OS_k)$. **(Goal No. 1)**

The consequence of the above, **Goal No.1** and **Goal No. 2** have confirmed and both U_i and OS_k mutually authenticate each other.

```
let OSNSReg =
in(ch_pv,(mMid:bitstring,mRMBpw:bitstring));
let Aik = h(con(h(xor(mMid,SMik)),Ks)) in
let Dik = xor(mRMBpw,SMik) in
out(ch_pv,(Dik)).
```

```
let OSNSAuth =
in(ch_pb,(sTMid:bitstring,sW1:bitstring,sTSm:bitstring,sK1:bitstring));
let aik = h(con(h(xor(sTMid,SMik)),Ks)) in
let m1 = xor(aik,xor(sW1,xor(sTSm,h(OSid)))) in
let K2 = h(con(sTMid,con(sW1,con(m1,sTSm)))) in
if K2 = sK1 then
event MUsrAuthen(sTMid);
new OSrn:bitstring;
new TSos:bitstring;
let W2 = xor(aik,xor(OSrn,xor(TSos,sTMid))) in
let SKEYom = h(con(sTMid,con(OSid,con(aik,con(m1,con(OSrn,con(sTSm,TSos))))))) in
let K3 = h(con(sTMid,con(m1,con(OSrn,con(sTSm,con(TSos,SKEYom)))))) in
out(ch_pb,(W2,TSos,K3)).
let OSNS = OSNSReg — OSNSAuth.
process !MUser — !OSNS
(*OSNSReg — OSNSAuth.*)
```

Figure 6.5: The ProVerif code for OSN server

6.6 Formal Security Verification Using ProVerif

In this section, I present the formal security verification of our proposed scheme in the ProVerif simulation tool. Whether an adversary may be able to attack the session key or not, can be tested using this tool.

I have simulated our proposed scheme in ProVerif 1.93 tool by specifying the declaration of channels, constants, functions, equations, events, queries used in the proposed ASPA-mOSN scheme as shown in Figure 6.4. I have also simulated the process of registration, login, and authentication phases of OSN server OS_k and mobile user U_i. The code for registration (OSNSReg)and authentication (OSNSAuth) of the OSN sever OS_k is demonstrated in Figure 6.5 and mobile user U_i in Figure 6.6. Here, I present the simulation results only.

```
(*————Mobile user starts———*)
let MUser=
new p:bitstring;
new q:bitstring;
let delta = BE(Bi) in
let RMBpw = h(con(Mid,h(con(Mpw,con(delta,p))))) in
out(ch_pv,(Mid,xor(RMBpw,q)));
in(ch_pv,(sDik:bitstring));
let G1 = xor(h(con(Mpw,delta)),p) in
let G2 = xor(h(con(Mpw,delta)), Mid) in
let G3 = h(con(Mid,con(Mpw,con(delta,p)))) in
!(
event MUsrStart(Mpw);
let p1 = xor(G1,h(con(Mpw,delta))) in
let nMid = xor(G2, h(con(Mpw,delta))) in
let nG3 = h(con(nMid,con(Mpw,con(delta,p1)))) in
if G3 = nG3 then
new Rmu:bitstring;
new TSm:bitstring;
let nAik = xor(sDik,RMBpw) in
let W1 = xor(nAik,xor(Rmu,xor(TSm,h(OSid)))) in
let K1 = h(con(nMid,con(W1,con(Rmu,TSm)))) in
let nTMid = xor(TMid,h(con(ROSid,TSm))) in
out(ch_pb,(nTMid,W1,TSm,K1));
in(ch_pb,(sW2:bitstring,sTSos:bitstring,sK3:bitstring));
let m2 = xor(sW2,xor(sTSos,xor(nMid,nAik))) in
let SKEYus = h(con(nMid,con(OSid,con(nAik,con(Rmu,con(m2,con(TSm,sTSos))))))) in
let K4 = h(con(nMid,con(Rmu,con(m2,con(TSm,con(sTSos,SKEYus)))))) in
if K4 = sK3 then
0
).
```

Figure 6.6: The ProVerif code for mobile user

```
– Query not attacker(SKEYus[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKEYus[])
RESULT not attacker(SKEYus[]) is true.
– Query not attacker(SKEYom[])
Completing...
200 rules inserted. The rule base contains 200 rules. 22 rules in the queue.
Starting query not attacker(SKEYom[])
RESULT not attacker(SKEYom[]) is true.
– Query inj-event(MUsrAuthen(Mid_28)) ==⟩ inj-event(MUsrStart(Mid_28))
Completing...
200 rules inserted. The rule base contains 200 rules. 28 rules in the queue.
Starting query inj-event(MUsrAuthen(Mid_28)) ==⟩ inj-event(MUsrStart(Mid_28))
RESULT inj-event(MUsrAuthen(Mid_28)) ==⟩ inj-event(MUsrStart(Mid_28)) is true.
```

Figure 6.7: The ProVerif simulation results

I execute these codes in the latest version of `ProVerif 1.93` simulator. Figure 6.7 is displaying the complete results of session key confidentiality and authentication are obtained. The observations of the execution are as follows:

- RESULT not attacker(SKEYus[]) is true.
- RESULT not attacker(SKEYom[]) is true.
- RESULT inj-event(MUsrAuthen(Mid_28)) ==> inj-event(MUsrStart (Mid_28)) is true.

Thus, the proposed scheme has achieved security verification.

6.7 Informal Security Analysis of ASPA-mOSN

The robustness of the proposed ASPA-mOSN scheme is discussed in this section. I have discussed the informal security analysis and shown how the proposed ASPA-mOSN scheme can prevent phishing attacks and various other well-known security attacks.

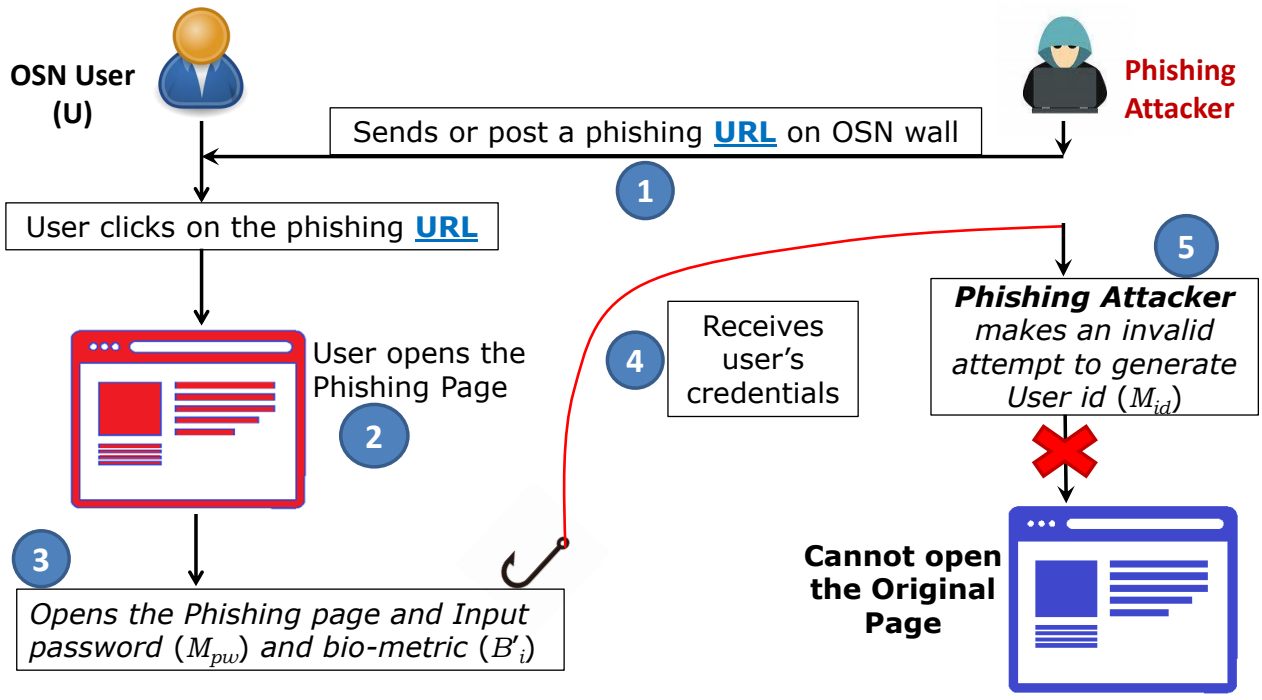

Figure 6.8: Resistance of phishing attack: First scenario.

6.7.1 Resilience against phishing attacks in mOSN

In online social networks, attackers can launch phishing attacks to steal login and other sensitive information of the users. An attacker can launch phishing attacks either by sending or posting the link of the compromised website which will look like the original one as described in Figure 6.8. When a user enters its information into that fake website, the phisher gets those sensitive credentials and uses them to login into the original website. Otherwise, an attacker steals the login messages to extract the original credentials as described in Figure 6.9.

Remedy 1: Phishing Attacks

In this case, the following steps are involved:

- Phishing Attackers or phishers send or post the link of the compromised website.
- An OSN user, U_i can click that link and provide his/her password (M_{pw}) and bio-metrics $\mathcal{B}_i$ into that fake website for login but never enters its id (M_{id}).
- Phishers get the original password and bio-metric but do not get the user id (M_{id}).

- Phishers cannot log in to the original website because without any user id (M_{id}) they cannot extract other credentials or generate the login message Msg_l to access the service.

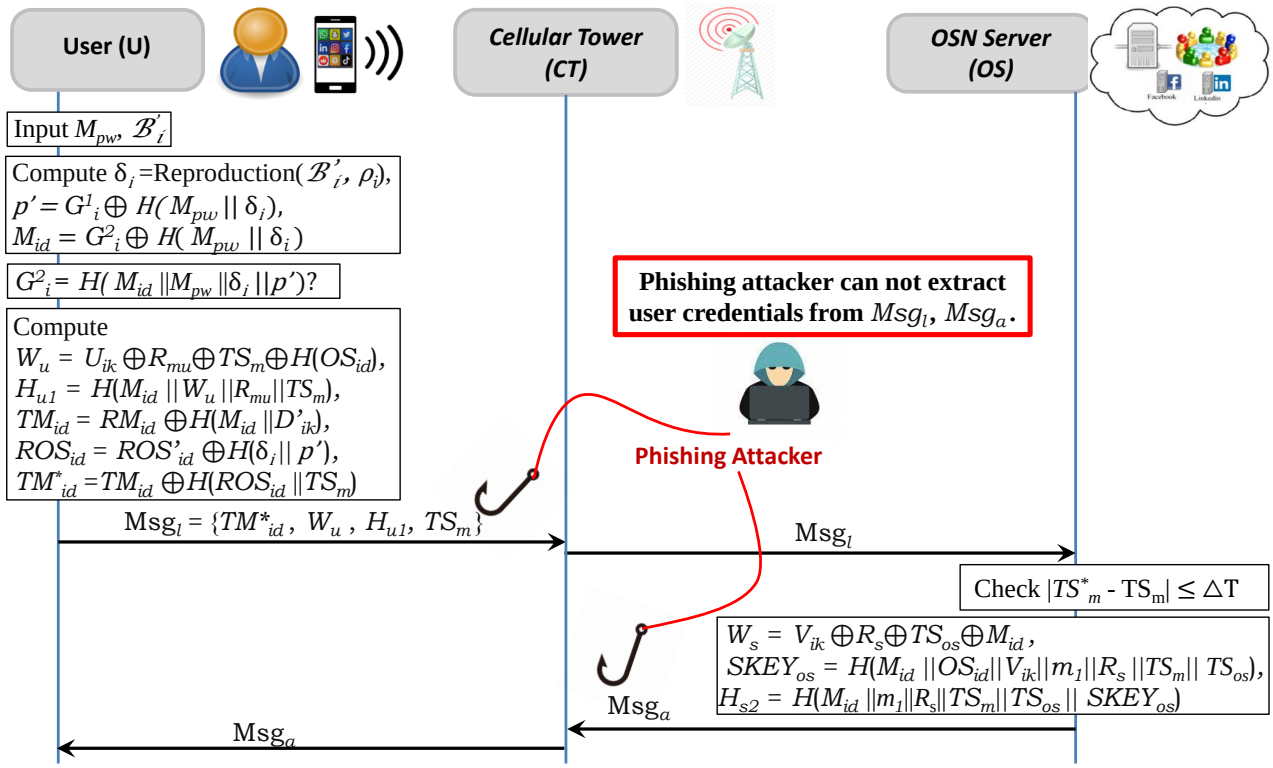

Figure 6.9: Resistance of phishing attack: Second scenario.

Remedy 2: Phishing Attacks

In this case, the following steps are involved:

- An OSN user, U_i enters its password (M_{pw}) and bio-metrics $\mathcal{B}_i$ into the mobile device.
- The user id (M_{id}) will be regenerated using the entered password (M_{pw}) and bio-metrics $\mathcal{B}_i$ and stored parameters G^1_i and G^2_i.
- User, U_i sends $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ to CT for sending the message to OS_k, after generating the parameters W_u, H_{u_1}, and timestamp TS_m.
- After receiving the Msg_l, OSN server, OS_k checks whether the message is received within a permissible time delay or not.
- After some other verification, OS_k sends the reply message $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ to to user U_i through CT.

A phisher can capture the login message Msg_l or reply message Msg_a from the network. An attacker can obtain the value TM^*_{id}, W_u, H_{u_1} etc. from those messages but cannot regenerate the original M_{id} using them. Thus, our proposed scheme is providing the complete remedy of phishing attacks in mobile OSN.

For login to the system, an OSN user (U_i) only inputs its password (M_{pw}) never enters its id (M_{id}). Adversary $\mathcal{A}$ does not extract the all original credentials by using any fake URL. $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ will be send to OS_k by U_i in the user login phase, similarly $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ will be sent to U_i by OS_k. All the original credentials are either hashed or masked with other various parameters before sending through the public network. An adversary can obtain the value TM^*_{id}, W_u, H_{u_1} etc. but he/she cannot derive the original credentials from them. Hence, our proposed ASPA-mOSN scheme will withstand phishing attacks in mobile OSN.

6.7.2 Discussion of various other security attacks on ASPA-mOSN

The informal analysis of security for the proposed ASPA-mOSN is demonstrated in this section.

Man-in-the-Middle Attack

An adversary $\mathcal{A}$ may attempt to modify the login message of a particular session for invalidating the login request of user U_i. Or it may attempt to build up third-party connectivity between both U_i and OS_k to launch a man-in-the-middle attack. To modify a message in the proposed ASPA-mOSN scheme, $\mathcal{A}$ needs the credentials, such as U_{ik}, OS_{id}, D'_{ik} and ROS_{id} or the shared key $SKEY_{os}$. Further, both messages are generated using various combinations of hash function, bitwise XOR and concatenation operations. Hence, our proposed ASPA-mOSN is very secure to resist man-in-the-middle attacks.

Replay Attack

When OS_k receives a replay message Msg_l from an adversary $\mathcal{A}$, as soon as possible OS_k checks the duration between the received timestamp TS^*_m of the message and the timestamp attached TS_m with Msg_l. OS_k immediately discards Msg_l if $|TS^*_m - TS_m| > \triangle T$. Therefore, $\mathcal{A}$ needs to send the message Msg_l within the time frame $\triangle T$ for a replay attack. The record $\langle M_{id}, R_{mu}, TS_m \rangle$ has been saved in the database of OS_k. Therefore, while another login request message, say $Msg'_l = \{TM^*_{id}, W'_u, H'_{u_1}, TS'_m\}$ will be received by OS_k, after

timestamp verification, further OS_k extract $R'_{mu} = W'_u \oplus TS'_m \oplus H(OS_{id}) \oplus V_{ik}$ and match it whether it is equal to the saved one in the database. Msg'_l will be considered as a replay message, if it exists in the database of OS_k. Thus, our proposed ASPA-mOSN can withstand a strong replay attack.

Forward Secrecy

An adversary $\mathcal{A}$ can get the session key $SKEY_{us}$ or $SKEY_{os}$ of some session. But the session key in our proposed ASPA-mOSN scheme, is calculated as $SKEY_{os} = SKEY_{us} = H(M_{id} ||OS_{id} ||U_{ik} ||R_{mu} ||R_s ||TS_m ||TS_{os})$ where $U_{ik} = V_{ik} = H(H(M_{id} \oplus SM_{ik}) ||K_s)$. Using R_{mu}, TS_m, R_s, and TS_{os}, the new session key $SKEY_{us}$ $(= SKEY_{os})$ is computed in an exclusive way for every new session. Hence, adversary $\mathcal{A}$ will not get any advantage or crucial information from that compromise session key, to compute the past session keys. Therefore, the proposed ASPA-mOSN ensures forward secrecy totally.

Stolen/lost Mobile Device Attack

ρ_i, $G_i^1 = H(M_{pw} ||\delta_i) \oplus p$, $G_i^2 = H(M_{pw} ||\delta_i) \oplus M_{id}$, and $G_i^3 = H(M_{id} ||M_{pw} ||\delta_i ||p)$ are saved in U_i's mobile device. When an attacker $\mathcal{A}$ acquires the mobile device of the user U_i, using those stored parameters $\mathcal{A}$ does not extract or predict the original credentials such as M_{id}, M_{pw},biometrics $\mathcal{B}_i$ etc. as it is not feasible to compute. Some other parameters like $D'_{ik}(= U_{ik} \oplus RMB_{pw_i}$, $U_{ik} = H(H(M_{id} \oplus SM_{ik}) ||K_s))$ and $RMB_{pw_i} = H(M_{id} ||H(M_{pw} ||\delta_i ||p))$ are also saved on the mobile device. But $H(\cdot)$ is a collision-resistant function and SM_{ik} and p are random integers, therefore within polynomial time, it is not possible to calculate M_{id}, M_{pw} and δ_i from D'_{ik} and U'_{ik}. Hence, the proposed ASPA-mOSN will resist this attack.

Session Key Security

As described in our proposed ASPA-mOSN scheme, a mutually shared session key $SKEY_{mu}$ $(= SKEY_{os})$ will establish between U_i and OS_k for future message communication as per Section 6.3.3. The session key will be calculated as $SKEY_{mu} = H(M_{id} ||OS_{id} ||U_{ik} ||R_{mu} ||m_2 ||TS_m ||TS_{os}) = H(M_{id} ||OS_{id} ||V_{ik} ||R_{mu} ||m_2 ||TS_m ||TS_{os}) = H(M_{id} ||OS_{id} ||V_{ik} ||m_1 ||R_s ||TS_m ||TS_{os}) = SKEY_{os}$. Without these original credentials M_{id}, OS_{id}, U_{ik} $(= V_{ik})$, R_s $(= m_2)$, an adversary $\mathcal{A}$ cannot generate the session key. Only mutually authenticate U_i and OS_k can establish the session key. Thus, the session key security can be maintained in the proposed ASPA-mOSN scheme.

Anonymity and Untraceability

An attacker $\mathcal{A}$ may able to capture the login messages transmitted by U_i. But the user identity M_{id} is hidden in the $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ as $TM^*_{id} = TM_{id} \oplus H(ROS_{id}||TS_m)$. Further, $\mathcal{A}$ cannot generate M_{id} from $W_u = U_{ik} \oplus R_{mu} \oplus TS_m \oplus H(OS_{id})$ and $H_{u_1} = H(M_{id} ||W_u ||R_{mu} ||TS_m)$, because it is computationally infeasible. OS_k sends $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ to U_i at the time of authentication. Similarly, from $W_s (= V_{ik} \oplus R_s \oplus TS_{os} \oplus M_{id})$ and $H_{s_2} (= H(M_{id} ||m_1 ||R_s ||TS_m ||TS_{os} ||SKEY_{os}))$ $\mathcal{A}$ cannot extract M_{id} as hash function (one-way) $H(\cdot)$ can resist collision. Thus, the proposed ASPA-mOSN will support the property of user anonymity.

In our proposed ASPA-mOSN scheme, I use either random nonce or timestamp to generate the messages Msg_l and Msg_a, that's why they are unique and dynamic in nature for every session. Due to these, there is no similarity between the two login messages of different sessions. Hence, an adversary never can trace the identity of a user in different sessions and the proposed ASPA-mOSN also supports the untraceability property.

Table 6.4: Comparison of security and functionality features.

Security Attributes	Bin *et al.* [26]	Saeed *et al.* [158]	He *et al.* [83]	Moghimi *et al.* [129]	Bojjagani *et al.* [27]	Munivel *et al.* [132]	Mustafa *et al.* [9]	Thakur and Yoshiura [174]	Lara textitet al. [107]	ASPA-mOSN
Login phase efficiency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Phishing attack	✓	✓	X	✓	✓	✓	✓	✓	✓	✓
Denial-of-service (DoS) attack	✓	✓	X	✓	✓	✓	✓	X	✓	✓
Strong reply attack	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
Server impersonation attack	X	✓	✓	X	✓	✓	X	✓	✓	✓
Stolen mobile device attack	X	X	✓	X	X	✓	✓	X	X	✓
Man-in-the-middle attack	X	✓	X	✓	✓	✓	✓	✓	X	✓
Forward secrecy	X	✓	✓	X	✓	✓	X	✓	✓	✓
Untraceability	X	✓	✓	X	X	X	X	X	✓	✓
Session key security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Efficient password change	X	X	X	X	X	X	X	✓	X	✓
Formal security proof	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Simulation using ProVerif/Others	X	X	X	X	X	✓	✓	✓	✓	✓

Server Impersonation Attack

An attacker $\mathcal{A}$ can launch the impersonation attack by pretending itself as OS_k and reply with a valid authorization message $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ to U_i. But Msg_a contains the hash value $H_{s_2} (= H(M_{id} ||m_1 ||RN_{os} ||TS_m ||TS_{os} ||SKEY_{os}))$ and $W_s (= V_{ik} \oplus RN_{os} \oplus TS_{os} \oplus$

M_{id}). For generating those parameters, $\mathcal{A}$ needs the credential $V_{ik}(= U_{ik} = H(H(M_{id} \oplus SM_{ik}) ||K_s))$ that can be generated only with the random number SM_{ik} and the secret key of server K_s. Thus, the proposed ASPA-mOSN can defend server impersonation attacks.

6.8 Performance Comparison of ASPA-mOSN

In the proposed ASPA-mOSN scheme, I have not used the traditional way of log in with the user ID and password. This proposed ASPA-mOSN is secure about the user information leakages because it stores either hashed value or masked values in the device. In this section, I will describe the analysis of the security and functionality of the proposed ASPA-mOSN scheme. A detailed comparison of different security issues among our proposed ASPA-mOSN with some other related authentication protocols is presented in Table 6.4.

Table 6.5: Computational costs comparison

Scheme	Cost of Computation	Execution Time (ms)
Saeed *et al.* [158]	7 T_h + 4 T_m+T_{sym}	≈ 264.52
He *et al.* [83]	> 9 T_h + 3 T_m	≈ 193.74
Bojjagani *et al.* [27]	2 T_h + 4 T_m	≈ 254.32
Munivel *et al.* [132]	2 T_h + 2 T_m	≈ 127.16
Mustafa *et al.* [9]	4 T_h + 6 T_m	≈ 380.48
Lara *et al.* [107]	10 T_h + 6 T_m + 2 T_{sym}	≈ 400.88
ASPA-mOSN	17 T_h + 1 T_m	≈ 71.58

6.8.1 Computation cost

Our proposed ASPA-mOSN scheme has used the collision-resistant (one-way) hash function $H(\cdot)$ most of the time, which requires very little computation overhead. The required execution time of T_m (ECC point multiplication)is ≈ 63.08 ms, T_{sym} (symmetric key encryption/decryption)is ≈ 8.70 ms, T_{FE} (fuzzy extractor function) is ≈ T_m ms, T_h (one-way hash function) is ≈ 0.50 ms, and T_x is *negligible* in a personal computer with an Intel 2.6 GHz processor and 1 GB RAM. Bilinear pairing and some other cryptographic operations have been avoided in the proposed ASPA-mOSN scheme, due to their high computation overhead. There are mainly two different phases in the authentication process. At the very beginning as described in Section 6.3.1, a mobile user U_i or OSN server OS_k register itself to CT (a Cellular

Tower). This is a one-time process; therefore those steps are not considered for computation cost overhead. In the next phase, user U_i sends a login message, and OS_k authorized him or her. U_i requires T_{FE} +10* T_h +10* T_x in the login phase, and OS_k required 7*T_h + 10*T_x time overhead for authentication. As the time required to execute an XOR operation (T_x) is almost negligible, the entire computation overhead of the login and authentication process is T_{FE} +17* T_h. The comparisons of computation cost between our proposed ASPA-mOSN and some other recent authentication schemes are displayed in Table 6.5. As the computation cost of T_h is very small compared to T_m, our proposed ASPA-mOSN scheme will require less computation time than others.

Remark 6.2. *As discussed in Section 6.7.1, at the time of phishing attacks, no extra computation overhead will be required. Only on the side of U_i's execution of login phase(T_{FE} +10* T_h) will be required but as the message Msg_l will not reach to OS_k, no further execution will be required on the server-side.*

6.8.2 Communication cost

According to the proposed ASPA-mOSN scheme, the registration process will execute once. So, I have not considered that for communication cost calculation. In the login phase $Msg_l = \{TM^*_{id}, W_u, H_{u_1}, TS_m\}$ is send from U_i and OS_k sent $Msg_a = \{W_s, H_{s_2}, TS_{os}\}$ to U_i for authentication. The outputs of the hash function and the identity have the size of 160 bits, and the size of the timestamp field is 32 bits. Therefore, Msg_l has $(160 + 160 + 160 + 32) = 512$ bits and Msg_a has $(160+160+32) = 352$. Hence, a total of 864 bits has been transmitted at the time of login and authentication. I have compared the communication costs between our proposed ASPA-mOSN and some other recent authentication schemes and the study is shown in Table 6.6.

It is observed in Table 6.6 that our proposed ASPA-mOSN scheme requires less communication overhead than all other schemes except for the schemes in [132] and [158]. The scheme described in [132] has a lesser number of bits and the scheme in [158] requires the same number of bits for communication, but they require 3 rounds of message communication, which will be more expensive in respect of message propagation time. Due to the huge size of OSN networks, message propagation time will be an important parameter for communication cost.

Remark 6.3. *At the time of phishing attacks, no extra communication overhead will be required. As given in Section 6.7.1, only U_i's password and biometric will be inserted into the*

fake website and maximum Msg_l *can be generated. Hence, the communication cost can be a maximum of* 512 *bits. No further communication will be required on the server-side.*

Table 6.6: Communication Costs Comparison

Scheme	No. of Messages	No. of Bits
Saeed *et al.* [158]	3	864
He *et al.* [83]	4	3296
Bojjagani *et al.* [27]	7	1672
Munivel *et al.* [132]	3	576
Mustafa *et al.* [9]	5	1312
Lara *et al.* [107]	3	1376
ASPA-mOSN	2	864

6.9 Summary

Stealing of users' credentials using phishing attacks is one of the most common social engineering practices for cyber-attackers. Traditional authentication methods are also not suitable due to the dynamic topographical nature of the mobile OSN. To address this challenge of phishing attack resistance in mOSNs, I proposed an efficient scheme that is both phishing-secured and lightweight. The security and performance evaluation of the proposed ASPA-mOSN scheme shows its merit over other existing schemes. It can be observed that the mediums used for phishing attacks have changed from traditional emails to social media-based phishing. There is a clear lag between sophisticated phishing attacks and existing countermeasures. There is no single solution for the phishing problem due to the heterogeneous nature of the attack vector. So more research is effort is needed.

My future research could be to design a content-based phishing detection mechanism and to select the best training models among different machine learning models. More research with various hybrid models, ML and nature inspired based phishing detection models, and deep learning based models with enlarged data-set are needed to study the effect of phishing attack detection accuracy.

Chapter 7

Conclusion and Future Works

This thesis has been motivated by the problem of security and privacy issues of Online Social networks (OSN). By introducing new features, and new functionalities, social networking sites are increasing their popularity. Users are engaging in surfing the site hours after hours. At the same time, new security and privacy threats are also emerging. In this thesis, we have tried to propose some authentication schemes to resist several security threats on OSN. In this chapter, we have summarized the major contributions of the thesis. The roadmap for future research directions of OSN is also highlighted.

7.1 Contributions

In this thesis, we have proposed some efficient user authentication schemes for resisting some privacy and security hazards in Online social networks, which are summarized as follows:

- Privacy-Preserving Efficient Location-Sharing Scheme for mOSN.
- DDoS Attack Resisting Authentication Protocol for mOSN Applications.
- Authentication Scheme ASPA-mOSN for Resisting Phishing Attacks in mOSN.

The first contribution is provided in **Chapter** 4. This chapter presents an efficient location sharing scheme for mOSNs and shows the ability to resist various active and passive security attacks that are present in the existing schemes. The proposed scheme integrates LBS and SNS into a set of single entity servers, thereby reducing their internal communication overhead. Our location sharing scheme for mOSNs shows both efficiency and flexibility in location update, sharing, and query of social friends and social strangers. Formal security

verification, authentication proof, and simulation results prove the security strength of the proposed scheme.

The second contribution is provided in **Chapter 5**. In this chapter, we proposed a DDoS attack resisting authentication protocol for mobile-based OSNs. The proposed scheme can efficiently resist the adversary users that repeatedly aim to login attempts to the server. Later on, through key-refilling, the genuine users are allowed to login into the server. Keeping the resource constrained nature of a mobile device, the proposed scheme is made lightweight, effective, and resource-friendly. Various security analyses and simulation results prove that the scheme is secured against different attacks. Low computation and communication overheads, high security, and efficient re-synchronization process of the proposed scheme make it appropriate for the real-world application of the mobile OSN.

The third and final contribution is provided in **Chapter 6**. Stealing users' credentials using phishing attacks is one of the most common social engineering practices for cyber-attackers. Traditional authentication methods are also not suitable due to the dynamic topographical nature of the mobile OSN. To address this challenge of phishing attack resistance in mOSNs, we proposed an efficient scheme that is both phishing-secured and lightweight. The security and performance evaluation of the proposed ASPA-mOSN scheme shows its merit over other existing schemes. It can be observed that the mediums used for phishing attacks have changed from traditional emails to social media-based phishing. There is a clear lag between sophisticated phishing attacks and existing countermeasures. There is no single solution for the phishing problem due to the heterogeneous nature of the attack vector. So more research is effort is needed.

7.2 Limitations of Current Work

This thesis proposes some schemes to resist some well-known security threats in mobile OSN. However, these schemes are derived with some limitations as listed below:

- We have designed and proposed our schemes based on those users who use the social network sites using mobile/laptop browsers with 4G mobile internet facility. But there are several other connectivity options, such as wifi, leased lines, etc. that are not considered. But according to published reports [34], [30], it is observed that users are using mobile applications more than browsers and wi-fi plays an important role in mobile connectivity.

- According to the proposed schemes cellular towers will act as a coordinator between users and OSN servers, which make CTs overburdened with the high computational overhead and CTs also need the large storage space for storing databases as well as $OSN servers$.

- The proposed protocols discussed in this thesis are aimed to design for real-life applications, but I have not evaluated them in the real-world environment. Hence, the societal impact of these protocols is not known to us.

7.3 Future Research Directions

Security and privacy issues are not separable in OSN. In this section, we present a roadmap of research in this domain. There are five layers in the map as depicted in Figure 7.1. Different Social network computing theories, information security technology, and Multimedia communication technology can be grouped in the first layer as the key theories and technologies. The Main OSN research aspects, such as control over personal information, user's perception of security, and defense against attacks are discussed in the second layer. The main open issues for research are in layer 3. The security measures related to those open issues are delivered in layer 4, and the probable solutions of each security measure are shown in layer 5.

In this vast research area, one direction can be exploring new privacy threats. Another direction can be the development of new protection mechanisms against these threats. We have suggested some research directions for possible future works as follows.

7.3.1 The exchange of personal information must be controlled by the owner

Personal information that will be shared and transferred by an OSN user must be controlled. For achieving these issues, an OSN user must have control over its data for sharing and spreading in the OSN. Users can select the privacy settings for controlling their shared information in current OSNs. Due to the lack of knowledge, most users do not correctly set their privacy settings in the network. So, these options do not work fruitfully in the OSNs. Moreover, users can not control the information shared by other users on the OSN. Therefore, this is an open issue for future research. Many researchers have discussed these issues in their work [142].

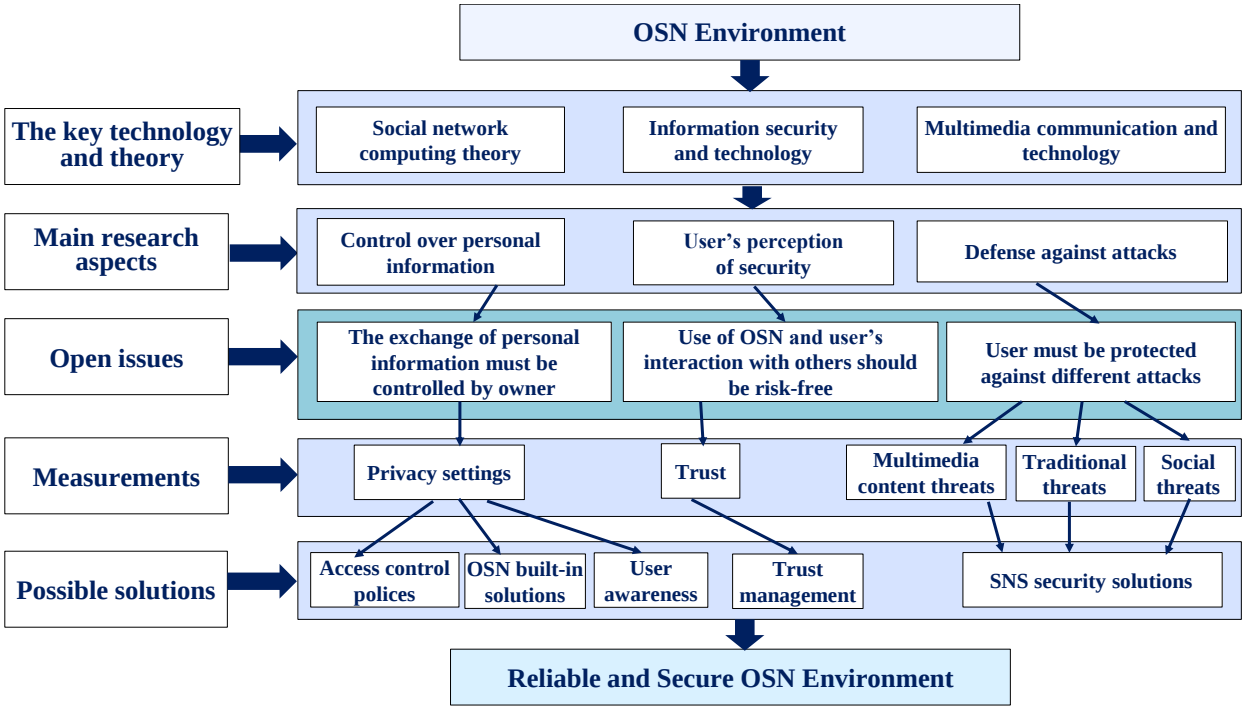

Figure 7.1: Future research roadmap in OSN

7.3.2 Use of OSN and user's interaction should be risk-free

OSN users thought that they are interacting only with their friends and family, whom they know very well. Hence, the information they are sharing on the OSN is secure. But sometimes, there are some people whom they never meet in the real world. They only know those people virtually and are connected in the OSN only. Most users do not have the concept of the possibilities of different types of hazards with such virtual friendships. Using the existing trust management techniques, the trustworthiness of any user can be predicted. Though several numbers of trust management and trust propagation research are there but still, it is an open challenge to calculate the trustworthiness of any user in the OSN.

7.3.3 User must be protected against different attacks

Different types of attacks are possible in the OSN platform, we have discussed these attacks in Chapter 1. Attackers use different ways to steal the information of the users. Attackers select someone as a victim and launch some cyber-attacks for harassing him or her. Several researchers have proposed some detection mechanisms and defense mechanisms but still, this is an open direction for the forthcoming researchers to find out a complete solution to defend

against the security attacks in this domain.

7.3.4 Existing deception related issues

I have also identified the following existing deception-related issues for future research.

Defense against malicious cyber deception

Many OSN users use social networking platforms for promoting themselves or their businesses. Adversaries use this activity with the malicious intention of cyber fraud. When an innocent OSN user believes them, the user can be victimized by some cyber deception. The existing techniques have a high chance to detect a genuine user as a malicious user. We need some defense mechanism to stop such malicious cyber deceptions.

Culture-awareness defense against OSN attacks

There are cultural variations all over the world. East Asian culture does not match western. Several interpersonal deceptions have been addressed by some research works, based on unconscious cultural beliefs. As cultural beliefs are a very sensitive issue, we have to develop some defense solutions for OSN attacks by increasing Culture-awareness.

Human awareness-based defense against OSN attacks

Some of the OSN attacks are incidents purely based on blind trust. Adversary establishes a trusted relationship with a victim user and traps him or her for OSN attacks, such as cyber-bullying, and cyber-grooming. We can stop these types of attacks by developing intent-aware or detectability-aware indications to understand the unrevealed OSN attacks.

Defense against automated social bots and human attackers

Previously attackers use some software programs (social bots) to launch any attacks. But nowadays human is used and paid for launching OSN attacks. The existing defense mechanism will not work properly as they use their human intelligence for spreading deceptive information. More attention to research is required to identify the human attacker and automated social bots.

Integrated defense techniques are required for detection, prevention, and mitigation

Several research works are there; some have proposed detection methods, some have proposed probable prevention solutions, and some have focused on mitigating techniques for OSN threats. Fake news or deceptive information etc. spread rapidly in OSN. A large number of defense solutions have been developed to detect manipulated or fake news. But a complete and integrated solution for detecting, preventing, and mitigating security threats is required in this domain.

Ethical and systematic guidelines will require for conducting OSN research

There should be some clear guidelines from the government and the research community for conducting OSN research without hampering users' security and privacy. Sometimes researchers purchase or crawl users' information from the OSN without their knowledge. Researchers published that data either after some naïve anonymization or as a graph. But by gathering that information from the different platforms, the adversary can launch some possible attacks. Researchers must preserve the confidentiality of the users' information before completing their research goal effectively.

Bibliography

[1] M. Abadi, B. Blanchet, and H. Comon-Lundh. Models and Proofs of Protocol Security: A Progress Report. In *21st International Conference on Computer Aided Verification (CAV'09), Springer*, pages 35–49, Grenoble, France, 2009.

[2] P. Abbate. Internet Crime Report 2020, Federal Bureau of Investigation, 2020. Available online: https://www.ic3.gov /Media/PDF/AnnualReport/2020_IC3Report.pdf, Accessed on June 2022.

[3] M. Abdalla, P. A. Fouque, and D. Pointcheval. Password-based authenticated key exchange in the three-party setting. In *International Workshop on Public Key Cryptography (PKC '05), Springer*, pages 65–84, Les Diablerets, Switzerland, 2005.

[4] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru. PhishAri: Automatic realtime phishing detection on Twitter. In *IEEE 2012 eCrime Researchers Summit*, pages 1–12, Las Croabas, Puerto Rico, 2012.

[5] R. Akbani, T. Korkmaz, and G. Raju. Mobile ad-hoc networks security. In *Recent advances in computer science and information engineering*, volume 4, pages 659–666. Springer, Berlin, Heidelberg, 2012.

[6] K. Akherfi, M. Gerndt, and H. Harroud. Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, 14(1):1–16, 2018.

[7] R. Alabdan. Phishing attacks survey: types, vectors, and technical approaches. *Future Internet*, 12(10):168, 2020.

[8] S. Ali, N. Islam, A. Rauf, I. U. Din, M. Guizani, and J. J. Rodrigues. Privacy and security issues in online social networks. *Future Internet*, 10(12):114, 2018.

[9] M. H. Alzuwaini and A. A. Yassin. An efficient mechanism to prevent the phishing attacks. *Iraqi Journal for Electrical & Electronic Engineering*, 17(1), 2021.

[10] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang. A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. *Future Generation Computer Systems*, 78:1005–1019, 2018.

[11] V. Anandpara, A. Dingman, M. Jakobsson, D. Liu, and H. Roinestad. Phishing IQ tests measure fear, not ability. In *11th International Conference on Financial Cryptography and Data Security (FC '07), Springer*, pages 362–366, Scarborough, Tobago, 2007.

[12] Article. Five Most DDoS Attacks, 2021. Available online: https://www.a10networks.com /blog/5-most–ddos-attacks/. Accessed on September, 2021.

[13] E. Athanasopoulos, A. Makridakis, S. Antonatos, D. Antoniades, S. Ioannidis, K. G. Anagnostakis, and E. P. Markatos. Antisocial networks: Turning a social network into a botnet. In *International Conference on Information Security (ISC '08)*, pages 146–160, Springer, Taipei, Taiwan, 2008.

[14] M. A. Awan. Pishing attacks in network security. *LC International Journal of STEM*, 1(1):29–33, 2020.

[15] S. Banerjee, A. K. Das, S. Chattopadhyay, S. S. Jamal, J. J. Rodrigues, and Y. Park. Lightweight failover authentication mechanism for iot-based fog computing environment. *Electronics*, 10(12):1417, 2021.

[16] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, and Y. Park. An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors*, 20(4):1215, 2020.

[17] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K.-K. R. Choo. A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment. *IEEE Internet of Things Journal*, 6(5):8739–8752, 2019.

[18] M. Barbeau, J. Hall, and E. Kranakis. Detecting impersonation attacks in future wireless and mobile networks. In *1st International Workshop on Secure Mobile Ad-hoc Networks and Sensors (MADNES '05)*, pages 80–95, Springer, Singapore, 2005.

[19] A. Beach, M. Gartrell, and R. Han. Solutions to security and privacy issues in mobile social networking. In *IEEE International Conference on Computational Science and Engineering (CSE '09)*, volume 4, pages 1036–1042, Vancouver, Canada, 2009.

[20] M. Belyaev and S. Gaivoronski. Towards load balancing in sdn-networks during ddos-attacks. In *2014 international science and technology conference (modern networking technologies)(MoNeTeC)*, pages 1–6, Moscow, Russia, 2014.

[21] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *9th ACM SIGCOMM Conference on Internet Measurement (IMC '09)*, pages 49—-62, Chicago, Illinois, USA, 2009.

[22] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[23] M. Beye, A. Jeckmans, Z. Erkin, P. Hartel, R. Lagendijk, and Q. Tang. Literature overview-privacy in online social networks. *Centre for Telematics and Information Technology, University of Twente*, 2010.

[24] M. Bhattacharya, S. Roy, S. Banerjee, and S. Chattopadhyay. Cryptanalysis of a Centralized Location-Sharing Scheme for Mobile Online Social Networks. In *Springer Advanced Computing and Systems for Security (ACSS-2020)*, pages 1–14, Kolkata, India, 2020.

[25] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *18th International Conference on World Wide Web*, pages 551–560, Madrid, Spain, 2009.

[26] S. Bin, W. Qiaoyan, and L. Xiaoying. A dns based anti-phishing approach. In *Second International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '10), IEEE*, volume 2, pages 262–265, Wuhan, Hubei, China, 2010.

[27] S. Bojjagani, B. Denslin, and P. Venkateswara. Phishpreventer: A secure authentication protocol for prevention of phishing attacks in mobile environment with formal verification. *Procedia Computer Science*, 171:1110–1119, 2020.

[28] J. Bonneau, J. Anderson, and G. Danezis. Prying data out of a social network. In *2009 international conference on advances in social network analysis and mining (ASONAM '09), IEEE*, pages 249–254, Athens, Greece, 2009.

[29] D. M. Boyd and N. B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230, 2007.

[30] P. Boyland. Time spent on Wifi in the US falls as users move to unlimited plans, April 19, 2018. https://www.opensignal.com/2018/04/19/time-spent-on-wifi-in-the-us-falls-as-users-move-to-unlimited-plans.

[31] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[32] M. Calabresi. Inside Russia's Social Media War on America., May 18, 2017. Available online: https://time.com/4783932/inside-russia-social-media-war-america/.

[33] R. Canetti and H. Krawczyk. Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '02), Springer*, pages 337–351, Amsterdam, The Netherlands, 2002.

[34] L. Ceci. Global mobile data traffic share 2021, by category, 2022. Statista, March 15, 2022. Available online: https://www.statista.com/statistics/383715/global-mobile-data-traffic-share/.

[35] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, and K.-Y. Yoo. Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access*, 5:3028–3043, 2017.

[36] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos. Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment. *IEEE Transactions on Dependable and Secure Computing*, 15(5):824–839, 2018.

[37] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse. Phishing attacks and defenses. *International Journal of Security and Its Applications*, 10(1):247–256, 2016.

[38] S. Chavhan, D. Gupta, B. Chandana, A. Khanna, and J. J. Rodrigues. Agent pseudonymous authentication-based conditional privacy preservation: An emergent intelligence technique. *IEEE Systems Journal*, 14(4):5233–5244, 2020.

[39] Z. Chen, X. Hu, X. Ju, and K. G. Shin. Lisa: Location information scrambler for privacy protection on smartphones. In *2013 IEEE Conference on Communications and Network Security (CNS '13)*, pages 296–304, Washington, DC, USA, 2013.

[40] S.-C. Chu, L. Chen, S. Kumar, S. Kumari, J. J. Rodrigues, and C.-M. Chen. Decentralized private information sharing protocol on social networks. *Security and Communication Networks*, 2020:1–12, 2020.

[41] CIC-DoS-dataset. Canadian Institute for Cybersecurity Dataset, 2017. University of New Brunswick, Available online at: https://www.unb.ca/cic/datasets/dos-dataset.html.

[42] CIC-IDS2017. Canadian Institute for Cybersecurity Dataset, 2017. University of New Brunswick, Available online: https:// www.unb.ca/cic/datasets/ids-2017.html.

[43] CIC-IDS2018. Canadian Institute for Cybersecurity Dataset on AWS, 2018. University of New Brunswick, Available online: https://www.unb.ca/cic/datasets/ids-2018.html.

[44] M. Conti, N. Dragoni, and V. Lesyk. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016.

[45] L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *5th International conference on Mobile systems, applications and services (MobiSys '07), ACM*, pages 233–245, San Juan, Puerto Rico, 2007.

[46] L. A. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, 2009.

[47] M. Darwish, A. Ouda, and L. F. Capretz. A cloud-based secure authentication (csa) protocol suite for defense against denial of service (dos) attacks. *Journal of Information Security and Applications*, 20:90–98, 2015.

[48] A. K. Das, S. Zeadally, and D. He. Taxonomy and Analysis of Security Protocols for Internet of Things. *Future Generation Computer Systems (Elsevier)*, 89:110–125, 2018.

[49] G. De Francisci Morales, A. Gionis, and C. Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *fifth ACM international conference on Web search and data mining (WSDM '12)*, pages 153–162, Seattle Washington, USA, 2012.

[50] Z. digital marketing. Top Valuable Facebook Statistics Infographic as of July 28, 2021. Available online: https://zephoria.com/top-valuable-facebook-statistics-as-of-july-28-2021/. Accessed 12 Jun 2022.

[51] X. Ding, L. Zhang, Z. Wan, and M. Gu. A brief survey on de-anonymization attacks in online social networks. In *2010 international conference on computational aspects of social networks (CASoN '10)*, pages 611–615, Taiyuan, China, 2010.

[52] Y. Ding, S. T. Peddinti, and K. W. Ross. Stalking Beijing from Timbuktu: A generic measurement approach for exploiting location-based social discovery. In *4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM '14)*, pages 75–80, Scottsdale Arizona, USA, 2014.

[53] M. Diomidous, K. Chardalias, A. Magita, P. Koutonias, P. Panagiotopoulou, and J. Mantas. Social and psychological effects of the internet use. *Acta informatica medica*, 24(1):66, 2016.

[54] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '04), Springer*, pages 523–540, Interlaken, Switzerland, 2004.

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

[56] R. D'Ovidio and J. Doyle. A study on cyberstalking: Understanding investigative hurdles. *FBI Law Enforcement Bulletin*, 72:10–17, 2003.

[57] H. Dreßing, J. Bailer, A. Anders, H. Wagner, and C. Gallas. Cyberstalking in a large sample of social network users: Prevalence, characteristics, and impact upon victims. *Cyberpsychology, Behavior, and Social Networking*, 17(2):61–67, 2014.

[58] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing*, 14(4):447–460, 2015.

[59] M. R. Faghani and H. Saidi. Malware propagation in online social networks. In *4th International Conference on Malicious and Unwanted Software (MALWARE '09)*, pages 8–14, Montréal, Quebec, Canada, 2009.

[60] A. Felt and D. Evans. Privacy Protection for Social Networking APIs. In *Web 2.0 Security and Privacy (W2SP '08)*, Oakland, California, 2008.

[61] M. A. Ferrag, M. Nafa, and S. Ghanemi. Security and privacy in mobile ad hoc social networks. In *Security, privacy, trust, and resource management in mobile and wireless communications*, pages 222–243. IGI Global, 2014.

[62] C. Fiesler and A. Bruckman. Copyright terms in online creative communities. In *Extended Abstracts on Human Factors in Computing Systems (CHI '14)*, pages 2551–2556. Association for Computing Machinery, Toronto, Ontario, Canada, 2014.

[63] FIPS PUB 180-1. Secure Hash Standard. National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. http://csrc.nist.gov/publications/fips /fips180-4/ fips-180-4.pdf. Accessed on August 2022.

[64] FIPS PUB 197. Advanced encryption standard. National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. Available : http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[65] M. Fire, G. Katz, and Y. Elovici. Strangers intrusion detection-detecting spammers and fake profiles in social networks based on topology anomalies. *Human journal*, 1(1):26–39, 2012.

[66] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. Doostari. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot. *Computer Networks*, 177:107333, 2020.

[67] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. N. Choudhary. Towards online spam filtering in social networks. In *19th Annual Network & Distributed System Security Symposium (NDSS '12)*, volume 12, pages 1–16, San Diego, California, 2012.

[68] A. Gaurav, B. Gupta, C. H. Hsu, D. Peraković, and F. J. G. Peñalvo. Filtering of distributed denial of services (ddos) attacks in cloud computing environment. In *IEEE International Conference on Communications Workshops (ICC Workshops '21)*, pages 1–6, Montreal, Canada, 2021.

[69] K. Ghazinour, S. Matwin, and M. Sokolova. Monitoring and recommending privacy settings in social networks. In *Joint EDBT/ICDT 2013 Workshops (EDBT '13)*, pages 164–168, Genoa, Italy, 2013.

[70] Google Ideas, Big Picture Team, Arbor Networks. Digital attack map. Available on: https://www.digitalattackmap.com/. Accessed on May 16 2021.

[71] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM conference on Computer and Communications Security (CCS'06)*, pages 89–98, Alexandria, Virginia, USA, 2006.

[72] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *17th ACM conference on Computer and communications security (CCS '10)*, pages 27–37, Chicago Illinois, USA, 2010.

[73] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *1st international conference on Mobile systems, applications and services (MobiSys '03)*, pages 31–42, San Francisco, California, 2003.

[74] G. G. Gulyás, B. Simon, and S. Imre. An efficient and robust social network de-anonymization attack. In *ACM Workshop on Privacy in the Electronic Society (WPES '16)*, pages 1–11, Vienna, Austria, 2016.

[75] L. Guo, C. Zhang, and Y. Fang. A Trust-based Privacy-Preserving Friend Recommendation Scheme for Online Social Networks. *IEEE Transactions on Dependable and Secure Computing*, 12(4):413 – 427, 2014.

[76] Y. Guo and S. Perreau. Detect ddos flooding attacks in mobile ad hoc networks. *International Journal of Security and Networks*, 5(4):259–269, 2010.

[77] B. B. Gupta and A. Dahiya. *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges, and Countermeasures*. CRC Press, 2021.

[78] F. Günther, M. Manulis, and T. Strufe. Key management in distributed online social networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '11)*, pages 1–7, Lucca, Italy, 2011.

[79] V. Harikrishnan, H. Sanket, K. Sahazeer, S. Vinay, and P. B. Honnavalli. Mitigation of ddos attacks using honeypot and firewall. In *International Conference on Data Analytics and Management (ICDAM '22), Springer*, pages 625–635, Jelenia Gora, Poland, 2022.

[80] D. Hayes, F. Cappa, and N. A. Le Khac. An effective approach to mobile device management: security and privacy issues associated with mobile applications. *Digital Business*, 1(1):100001, 2020.

[81] B. Z. He, C. M. Chen, Y. P. Su, and H. M. Sun. A defence scheme against identity theft attack based on multiple social networks. *Expert Systems with Applications*, 41(5):2345–2352, 2014.

[82] D. He, Z. Cao, X. Dong, and J. Shen. User self-controllable profile matching for privacy-preserving mobile social networks. In *IEEE International Conference on Communication Systems (ICCS '14)*, pages 248–252, Macau, China, 2014.

[83] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen. Efficient privacy-aware authentication scheme for mobile cloud computing services. *IEEE Systems Journal*, 12(2):1621–1631, 2016.

[84] D. He, N. Kumar, J. H. Lee, and R. S. Sherratt. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Transactions on Consumer Electronics*, 60(1):30–37, 2014.

[85] X. Hu, T. H. Chu, V. C. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. Chan. A survey on mobile social networks: Applications, platforms, system architectures, and future research directions. *IEEE Communications Surveys & Tutorials*, 17(3):1557–1581, 2014.

[86] D. Huang and H. Wu. Chapter 5 - mobile cloud offloading models. In *Mobile Cloud Computing*, pages 115–152. Morgan Kaufmann, 2018.

[87] R. Hummel. Netscout threat intelligence report. Available on: https://www.netscout.com /threatreport, 1H 2020.

[88] T. Hwang, I. Pearce, and M. Nanis. Socialbots: Voices from the fronts. *ACM Interactions*, 19(2):38–45, 2012.

[89] A. Irshad, S. A. Chaudhry, Q. Xie, X. Li, M. S. Farash, S. Kumari, and F. Wu. An enhanced and provably secure chaotic map-based authenticated key agreement in multi-server architecture. *Arabian Journal for Science and Engineering*, 43(3):811–828, 2017.

[90] N. Jabeur, S. Zeadally, and B. Sayed. Mobile social networking applications. *Communications of the ACM*, 56(3):71–79, 2013.

[91] C. Jianyong, W. Guihua, S. Linlin, and J. Zhen. Differentiated security levels for personal identifiable information in identity management system. *Expert Systems with Applications*, 38(11):14156–14162, 2011.

[92] M. Johns and S. Lekies. Tamper-resistant likejacking protection. In *International Workshop on Recent Advances in Intrusion Detection (RAID '13), Springer*, pages 265–285, Rodney Bay, St. Lucia, 2013.

[93] Y. Jung, Y. Nam, J. Kim, W. Jeon, H. Lee, and D. Won. Key Management Scheme Using Dynamic Identity-Based Broadcast Encryption for Social Network Services. In *Advances in Computer Science and its Applications (CSA '13), Springer*, volume 279, pages 435–443, Danang, Vietnam, 2013.

[94] K. Kalkan and F. Alagöz. A distributed filtering mechanism against ddos attacks: Scoreforcore. *Computer Networks*, 108:199–209, 2016.

[95] K. Kalkan, G. Gür, and F. Alagöz. Filtering-based defense mechanisms against ddos attacks: A survey. *IEEE Systems Journal*, 11(4):2761–2773, 2016.

[96] M. Y. Kharaji, F. S. Rizi, and M. R. Khayyambashi. A new approach for finding cloned profiles in online social networks. *arXiv preprint arXiv:1406.7377*, 2014.

[97] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *International Symposium on Spatial and Temporal Databases (SSTD '07), Springer*, pages 239–257, Boston, MA, USA, 2007.

[98] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services. In *21st International Conference on Data Engineering Workshops (ICDEW'05)*, pages 1248–1248, Tokyo, Japan, 2005.

[99] H. Kim, J. Tang, and R. Anderson. Social authentication: harder than it looks. In *International Conference on Financial Cryptography and Data Security (FC '12), Springer*, pages 1–15, Bonaire, Netherlands, 2012.

[100] L. Kocarev and S. Lian. Chaos-based cryptography: Theory, algorithms and applications. *SCI Book Series, Springer*, 2011.

[101] G. Kontaxis, I. Polakis, S. Ioannidis, and E. P. Markatos. Detecting social network profile cloning. In *IEEE international conference on pervasive computing and communications workshops (PerCom Workshops '11)*, pages 295–300, Seattle, WA, USA, 2011.

[102] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow. Amppot: Monitoring and defending against amplification ddos attacks. In *International Symposium on Recent Advances in Intrusion Detection (RAID '15), Springer*, pages 615–636, Kyoto, Japan, 2015.

[103] H. Krasnova, O. Günther, S. Spiekermann, and K. Koroleva. Privacy concerns and identity in online social networks. *Identity in the Information Society*, 2(1):39–63, 2009.

[104] K. Krombholz, H. Hobel, M. Huber, and E. Weippl. Advanced social engineering attacks. *Journal of Information Security and applications*, 22:113–122, 2015.

[105] S. Kumari, M. K. Khan, and M. Atiquzzaman. User authentication schemes for wireless sensor networks: A review. *Ad Hoc Networks (Elsevier)*, 27:159–194, 2015.

[106] L. Wu, X. Du and J. Wu. Mobifish: A lightweight anti-phishing scheme for mobile phones. In *23rd International Conference on Computer Communication and Networks (ICCCN '14)*, Shanghai, China, 2014.

[107] E. Lara, L. Aguilar, and J. A. García. Lightweight authentication protocol using self-certified public keys for wireless body area networks in health-care applications. *IEEE Access*, 9:79196–79213, 2021.

[108] G. Laura, H. Caroline, and W. Barry. Studying Online Social Networks. *Journal of Computer-Mediated Communication*, 3(1), 1997.

[109] C. C. Lee, D. C. Lou, C. T. Li, and C. W. Hsu. An extended chaotic-maps-based protocol with key agreement for multiserver environments. *Nonlinear Dynamics*, 76(1):853–866, 2014.

[110] Y. Lei, A. Quintero, and S. Pierre. Mobile services access and payment through reusable tickets. *Computer Communications*, 32(4):602–610, 2009.

[111] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen. Privacy leakage of location sharing in mobile social networks: Attacks and defense. *IEEE Transactions on Dependable and Secure Computing*, 15(4):646–660, 2016.

[112] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen. Privacy Leakage of Location Sharing in Mobile Social Networks: Attacks and Defense. *IEEE Transactions on Dependable and Secure Computing*, 15(4):646–660, 2018.

[113] J. Li, J. Li, X. Chen, Z. Liu, and C. Jia. {MobiShare}+: Security improved system for location sharing in mobile online social networks. *Journal of Internet Services and Information Security*, 4(1):25–36, 2014.

[114] J. Li, H. Yan, Z. Liu, X. Chen, X. Huang, and D. S. Wong. Location-sharing systems with enhanced privacy in mobile online social networks. *IEEE Systems Journal*, 11(2):439–448, 2015.

[115] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren. All your location are belong to us: Breaking mobile social networks for automated user location tracking. In *15th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '14)*, pages 43–52, Philadelphia Pennsylvania, USA, 2014.

[116] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen. Location Privacy Preservation in Collaborative Spectrum Sensing. In *The 31st Annual IEEE International Conference on Computer Communications (INFOCOM '12)*, pages 729–737, Orlando, Florida USA, 2012.

[117] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari. A robust ecc-based provable secure authentication protocol with privacy preserving for industrial internet of things. *IEEE Transactions on Industrial Informatics*, 14(8):3599–3609, 2017.

[118] X. Li, J. Niu, S. Kumari, S. H. Islam, F. Wu, M. K. Khan, and A. K. Das. A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security. *Wireless Personal Communications*, 89:1–29, 2016.

[119] H. Lin. Efficient mobile dynamic id authentication and key agreement scheme without trusted servers. *International Journal of Communication Systems*, 30(1):e2818, 2017.

[120] J. Lin, J. I. Hong, D. P. Siewiorek, and N. Sadeh. Rethinking Location Sharing: Exploring the Implications of Social-Driven vs. Purpose-Driven Location Sharing. In *12th ACM international conference on Ubiquitous computing ((UbiComp '10))*, pages 85–94, Copenhagen, Denmark, 2010.

[121] Z. Liu, J. Li, X. Chen, J. Li, and C. Jia. New privacy-preserving location sharing system for mobile online social networks. In *Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC '13), IEEE*, pages 214–218, Compiegne, France, 2013.

[122] Z. Liu, D. Luo, J. Li, X. Chen, and C. Jia. N-mobishare: new privacy-preserving location-sharing system for mobile online social networks. *International Journal of Computer Mathematics*, 93(2):384–400, 2016.

[123] H. Lu, J. Li, and M. Guizani. A novel id-based authentication framework with adaptive privacy preservation for vanets. In *Proceedings of Computing, Communications and Applications Conference (ComComAp '12)*, pages 345–350, 2012.

[124] R. Lundeen, J. Ou, and T. Rhodes. New ways im going to hack your web app. *Blackhat AD*, pages 1–11, 2011.

[125] H. A. M. Shafahi, L. Kempers. Phishing through social bots on twitter. In *Proceedings of IEEE International Conference on Big Data (Big Data)*, Washington, DC, USA, 2016.

[126] C. McCarthy. Twitter crippled by denial-of-service attack. Available on: https://www.cnet.com/news/twitter-crippled-by-denial-of-service-attack/. Accessed on August 06, 2009.

[127] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. *Internet denial of service: attack and defense mechanisms (Radia Perlman Computer Networking and Security)*. Prentice Hall PTR, 2004.

[128] P. S. Modi. How to fake your location on facebook, whatsapp and snapchat. Available on: https://www.mobigyaan.com/how-to-fake-your-location-on-facebook-whatsapp-and-snapchat-guide. Accessed on September 2020.

[129] M. Moghimi and A. Y. Varjani. New rule-based phishing detection method. *Expert systems with applications*, 53:231–242, 2016.

[130] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host identity protocol. Technical report, RFC 5201, April, 2008.

[131] MSrivatsa and M. Hicks. Deanonymizing mobility traces: Using social network as a side-channel. In *ACM conference on Computer and communications security (CCS '12)*, pages 628–637, Philadelphia, Pennsylvania, USA , 2012.

[132] E. Munivel and A. Kannammal. New authentication scheme to secure against the phishing attack in the mobile cloud computing. *Security and Communication Networks*, 2019:1–11, 2019.

[133] S. Namasudra and P. Roy. A new secure authentication scheme for cloud computing environment. *Concurrency Computation: Practice and Experience*, 29(20):e3864, 2017.

[134] M. Nauman, N. Azam, and J. Yao. A three-way decision making approach to malware analysis using probabilistic rough sets. *Information Sciences*, 374:193–209, 2016.

[135] H. Nissenbaum. A contextual approach to privacy online. *Daedalus*, 140(4):32–48, 2011.

[136] G. Noh, H. Oh, Y.-m. Kang, and C.-k. Kim. Psd: Practical sybil detection schemes using stickiness and persistence in online recommender systems. *Information Sciences*, 281:66–84, 2014.

[137] (NSF). U.S. National Science Foundation, NS-3.28, 2018. https://www.nsnam.org/releases/ns-3-33/. Accessed on December 2021.

[138] J. A. Obar and S. S. Wildman. Social media definition and the governance challenge-an introduction to the special issue. *Obar, JA and Wildman, S.(2015). Telecommunications policy*, 39(9):745–750, 2015.

[139] O. Okunoye, N. Azeez, and F. Ilurimi. A web enabled anti-phishing solution using enhanced heuristic based technique. *Futa Journal of Research in Science*, 2017.

[140] Y. Ouyang, Z. Le, Y. Xu, N. Triandopoulos, S. Zhang, J. Ford, and F. Makedon. Providing anonymity in wireless sensor networks. In *IEEE international conference on pervasive services (PERSER '07)*, pages 145–148, Istanbul, 2007.

[141] F. A. Ozbay and B. Alatas. Discovery of multi-objective overlapping communities within social networks using a socially inspired metaheuristic algorithm. *International Journal of Computer Networks and Applications*, 4(6):148–158, 2017.

[142] S. Peng, Y. Zhou, L. Cao, S. Yu, J. Niu, and W. Jia. Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*, 106:17–32, 2018.

[143] S. Pokin. Myspace'hoax ends with suicide of dardenne prairie teen. *St. Louis Post-Dispatch*, 2007.

[144] I. Polakis, P. Ilia, F. Maggi, M. Lancini, G. Kontaxis, S. Zanero, S. Ioannidis, and A. D. Keromytis. Faces in the distorting mirror: Revisiting photo-based social authentication. In *ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, pages 501–512, Scottsdale Arizona, USA, 2014.

[145] I. Polakis, M. Lancini, G. Kontaxis, F. Maggi, S. Ioannidis, A. D. Keromytis, and S. Zanero. All your face are belong to us: Breaking facebook's social authentication. In *Proceedings of the 28th annual computer security applications conference (ACSAC '12)*, pages 399–408, Orlando Florida, USA, 2012.

[146] Z. Qian, C. Chen, I. You, and S. Lu. ACSP: A novel security protocol against counting attack for UHF RFID systems. *Computers & Mathematics with Applications*, 63(2):492–500, 2012.

[147] J. Qu and X.-L. Tan. Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem. *Journal of Electrical and Computer Engineering*, 2014(16):1–6, 2014.

[148] F. Rahman, M. E. Hoque, F. A. Kawsar, and S. I. Ahamed. Preserve your privacy with pco: A privacy sensitive architecture for context obfuscation for pervasive e-community based applications. In *Proceedings of Second International Conference on Social Computing (SocialCom '10)*, pages 41–48, Minneapolis, Minnesota, USA, 2010.

[149] S. M. M. Rahman, M. Mambo, A. Inomata, and E. Okamoto. An anonymous on-demand position-based routing in mobile ad hoc networks. In *Proceedings of the International Symposium on Applications on Internet (SAINT'06)*, pages 300–306, Phoenix, AZ, USA, 2006.

[150] S. Rass, R. Wigoutschnigg, and P. Schartner. Doubly-Anonymous Crowds: Using Secret-Sharing to achieve Sender-and Receiver-Anonymity. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2(4):27–41, 2011.

[151] S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park. Social network security: Issues, challenges, threats, and solutions. *Information sciences*, 421:43–69, 2017.

[152] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, A. Flammini, and F. Menczer. Detecting and tracking political abuse in social media. In *Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11)*, volume 5, pages 297–304, Barcelona, Catalonia, Spain, 2011.

[153] U. U. Rehman, W. A. Khan, N. A. Saqib, and M. Kaleem. On detection and prevention of clickjacking attack for osns. In *11th International Conference on Frontiers of Information Technology (FIT '13)*, pages 160–165, Islamabad, Pakistan, 2013.

[154] P. Research. Wombat State of the Phish, 2019. Available online: https://www.proofpoint.com /sites/default/files/wombatsecurity/Wombat_Proofpoint_2019 State of the Phish Report _Final.pdf.

[155] L. Rongxing, L. Xiaodong, and S. Xuemin. Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *29th Annual IEEE International Conference on Computer Communications (INFOCOM '10)*, pages 1–9, San Diego, CA, USA, 2010.

[156] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos. On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. *IEEE Access*, 5(1):25808–25825, 2017.

[157] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo. Chaotic Map-based Anonymous User Authentication Scheme with User Biometrics and Fuzzy Extractor for Crowdsourcing Internet of Things. *IEEE Internet of Things Journal*, 5(4):2884–2895, 2018.

[158] M. Saeed and H. S. Shahhoseini. Appma-an anti-phishing protocol with mutual authentication. In *IEEE symposium on Computers and Communications (ISCC '10)*, pages 308–313, Riccione, Italy, 2010.

[159] A. Sahai and B. Waters. Fuzzy Identity-Based Encryption. In *Advances in Cryptology (EUROCRYPT '05), Springer*, pages 457–473, Aarhus, Denmark, 2005.

[160] P. Sarkar. A simple and generic construction of authenticated encryption with associated data. *ACM Transactions on Information and System Security*, 13(4):1–16, 2010. Article No. 33.

[161] B. Schneier. *Applied Cryptography Protocols Algorithms and Source Code in C.* John Wiley and Sons Inc., 2nd edition, 1996.

[162] S. Seng, M. N. Al-Ameen, and M. Wright. Understanding users' decision of clicking on posts in facebook with implications for phishing. In *Workshop on Technology and Consumer Protection (ConPro '18)*, San Francisco, CA, USA, 2018.

[163] N. Shen, J. Yang, K. Yuan, C. Fu, and C. Jia. An efficient and privacy-preserving location sharing mechanism. *Computer Standards & Interfaces*, 44:102–109, 2016.

[164] N. A. Shozi and J. Mtsweni. Big data privacy in social media sites. In *2017 IST-Africa Week Conference (IST-Africa)*, pages 1–6. Windhoek, Namibia, 2017.

[165] K. Simoens, J. Bringer, H. Chabanne, and S. Seys. A framework for analyzing template security and privacy in biometric authentication systems. *IEEE Transactions on Information Forensics and Security*, 7(2):833–841, 2012.

[166] Statista. Most popular social networks worldwide as of January 2022. Available online: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/. Accessed 12 Jun 2022.

[167] Statista. Number of social network users in India from 2015 to 2020, with estimates until 2040. Available online: https://www.statista.com/statistics/278407/number-of-social-network-users-in-india/. Accessed 12 Jun 2022.

[168] D. Stebila, P. Udupi, and S. Chang. Multi-factor password-authenticated key exchange. *Cryptology ePrint Archive, Paper 2008/214*, 2008. Available online: http://eprint.iacr.org/2008/214.

[169] G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao. Follow the green: growth and dynamics in twitter follower markets. In *2013 conference on Internet measurement conference (IMC '13)*, pages 163–176, Barcelona, Spain, 2013.

[170] Y. Sun, M. Chen, L. Hu, Y. Qian, and M. M. Hassan. ASA: Against statistical attacks for privacy-aware users in Location Based Service. *Future Generation Computer Systems*, 70:48–58, 2017.

[171] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[172] P. Syverson and I. Cervesato. The Logic of Authentication Protocols. In *International School on Foundations of Security Analysis and Design (FOSAD '00), Springer*, pages 63–136, Bertinoro, Italy, 2001.

[173] K. P. Tang, J. Lin, J. I. Hong, D. P. Siewiorek, and N. Sadeh. Rethinking location sharing: exploring the implications of social-driven vs. purpose-driven location sharing. In *12th ACM international conference on Ubiquitous computing (UbiComp '10)*, pages 85–94, Copenhagen, Denmark, 2010.

[174] T. N. Thakur and N. Yoshiura. Antiphimbs-auth: A new anti-phishing model to mitigate phishing attacks in mobile banking system at authentication level. In *International Conference on Database Systems for Advanced Applications (DASFAA '21)*, pages 365–380, Taipei, Taiwan, 2021.

[175] S. N. Thanh Vu, M. Stege, P. I. El-Habr, J. Bang, and N. Dragoni. A survey on botnets: Incentives, evolution, detection and current trends. *Future Internet*, 13(8):198, 2021.

[176] T. Theodoros and K. Loukas. *Online Social Network Phishing Attack, Encyclopedia of Social Network Analysis and Mining*. Springer, New York, NY, second edition, 2018.

[177] K. Thomas and D. M. Nicol. The koobface botnet and the rise of social malware. In *5th International Conference on Malicious and Unwanted Software (MALWARE '10)*, pages 63–70, Nancy, Lorraine, 2010.

[178] D. N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *6th USENIX symposium on Networked systems design and implementation (NSDI '09)*, volume 9, pages 15–28, Boston, Massachusetts, 2009.

[179] J. L. Tsai and N. W. Lo. A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card. *International Journal of Communication Systems*, 28(13):1955–1963, 2015.

[180] C. Umit and A. Bilal. A new direction in social network analysis: Online social network analysis problems and applications. *Physica A: Statistical Mechanics and its Applications*, 535:122372, 2019.

[181] B. E. Ur and V. Ganapathy. Evaluating attack amplification in online social networks. In *Web 2.0 Security and Privacy Workshop (W2SP '09)*, volume 2, pages 1–8, Oakland, California, 2009.

[182] H. C. Van Tilborg and S. Jajodia. *Encyclopedia of cryptography and security*. Springer Science & Business Media, second edition, 2014.

[183] S. Venkatesanand, V. A. Oleshchuk, C. Chellappan, and S. Prakash. Analysis of key management protocols for social networks. *Social Network Analysis and Mining*, 6(3):1–16, 2016.

[184] H. Wang, D. Guo, Q. Wen, and H. Zhang. Chaotic Map-Based Authentication Protocol for Multiple Servers Architecture. *IEEE Access*, 7:161340–161349, 2019.

[185] D. Warburton. DDoS Attack Trends for 2020, 2020. Available online: https://www.f5.com/labs/articles/threat-intelligence/ddos-attack-trends-for-2020/. Accessed on September, 2021.

[186] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu. Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for tmis. *Security and Communication Networks*, 9(13):1983–2001, 2016.

[187] W. Wei, F. Xu, and Q. Li. Mobishare: Flexible privacy-preserving location sharing in mobile online social networks. In *31st Annual IEEE International Conference on Computer Communications (INFOCOM '12)*, pages 2616–2620, Orlando, Florida USA, 2012.

[188] Wikipedia. Cell site. Available online: https://en.wikipedia.org/wiki/Cell_site. Accessed on September 2020.

[189] C. Wisniewski. Location-based threats: How cybercriminals target you based on where you live. Available online: https://news.sophos.com/en-us/2016/05/03/location-based-ransomware-threat-research/. 3 May 2016.

[190] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE symposium on security and privacy*, pages 223–238, Oakland, California, 2010.

[191] Worldometers and . B. World. Internet Live Stats—Internet Usage & amp. Social Media Statistics, Available online: http://www.internetlivestats.com/. Accessed 12 Jun 2022.

[192] B. Wu, J. Chen, J. Wu, and M. Cardei. A survey of attacks and countermeasures in mobile ad hoc networks. In *(eds.)Wireless network security. Signals and Communication Technology*, pages 103–135. Springer, Boston, MA, 2007.

[193] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi, and Z. Tianning. Andbot: towards advanced mobile botnets. In *4th USENIX conference on Large-scale exploits and emergent threats (LEET '11)*, pages 11–11, Boston MA, USA, 2011.

[194] X. Xiao, C. Chen, A. K. Sangaiah, G. Huc, R. Ye, and Y. Jiang. CenLocShare: A centralized privacy-preserving location-sharing system for mobile online social networks. *Future Generation Computer Systems*, 86:863–872, 2018.

[195] K. Xue, P. Hong, and C. Ma. A light weight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer and System Sciences*, 80:195–206, 2014.

[196] G. Yan, G. Chen, S. Eidenbenz, and N. Li. Malware propagation in online social networks: nature, dynamics, and defense implications. In *6th acm symposium on information, computer and communications security (ASIACCS '11)*, pages 196–206, Hong Kong, China, 2011.

[197] X. Yang, R. Lu, H. Liang, and X. Tang. SFPM: A secure and fine-grained privacy-preserving matching protocol for mobile social networking. *Big Data Research*, 3:2–9, 2016.

[198] S. Yardi, N. Feamster, and A. Bruckman. Photo-based authentication using social networks. In *First Workshop on Online Social Networks (WOSN '08)*, pages 55–60, Seattle, WA, USA, 2008.

[199] S. Yazji, P. Scheuermann, R. P. Dick, G. Trajcevski, and R. Jin. Efficient location aware intrusion detection to protect mobile devices. *Personal and Ubiquitous Computing*, 18(1):143–162, 2014.

[200] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *ACM conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '06)*, pages 267–278, Pisa, Italy, 2006.

[201] L. Zhang. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos, Solitons and Fractals 37*, 50(1):669–674, 2008.

[202] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan. Privacy-preserving profile matching for proximity-based mobile social networking. *IEEE Journal on Selected Areas in Communications*, 31(9):656–668, 2013.

[203] Z. Zhang, L. Zhou, X. Zhao, G. Wang, Y. Su, M. Metzger, H. Zheng, and B. Y. Zhao. On the validity of geosocial mobility traces. In *Twelfth ACM Workshop on Hot Topics in Networks (HotNets-XII)*, pages 1–7, College Park, Maryland, USA, 2013.