Security and Privacy for Mobile Social Networks

by

Kuan Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the ever-increasing demands of people's social interactions, traditional online social networking applications are being shifted to the mobile ones, enabling users' social networking and interactions anywhere anytime. Due to the portability and pervasiveness of mobile devices, such as smartphones, wearable devices and tablets, Mobile Social Network (MSN), as a promising social network platform, has become increasingly popular and brought immense benefits. In MSN, users can easily discover and chat with social friends in the vicinity even without the Internet; vehicle drivers and passengers can exchange traffic information, videos or images with other vehicles on the road; customers in a shopping mall can share sale information and recommend it to their friends. With MSNs, massive opportunities are created to facilitate people's social interactions and enlarge the inherent social circle.

However, the flourish of MSNs also hinges upon fully understanding and managing the challenges, such as security threats and privacy leakage. Security and privacy concerns rise as the boom of MSN applications comes up, but few users have paid adequate attentions to protect their privacy-sensitive information from disclosing. First of all, to initiate social interactions, users sometimes exchange their social interests or preferences with each other (including strangers in the vicinity) without sufficient protections. As such, some private information may be inferred from the exchanged social interests by attackers and untrusted users. Secondly, some malicious attackers might forge fake identities or false contents, such as spam and advertisements, to disrupt MSNs or mislead other users. These attackers could even collude and launch a series of security threats to MSNs. In addition, massive social network data are usually stored in untrusted cloud servers, where data confidentiality, authentication, access control and privacy are of paramount importance. Last but not least, the trade-off between data availability and privacy should be taken into account when the data are stored, queried and processed for various MSN applications. Therefore, novel security and privacy techniques become essential for MSN to provide sufficient and adjustable protections.

In this thesis, we focus on security and privacy for MSNs. Based on the MSN architecture and emerging applications, we first investigate security and privacy requirements for MSNs and introduce several challenging issues, i.e., spam, misbehaviors and privacy leakage. To tackle these problems, we propose efficient security and privacy preservation schemes for MSNs. Specifically, the main contributions of this thesis can be three-fold. Firstly, to address the issues of spam in autonomous MSNs, we propose a personalized fine-grained spam filtering scheme (PIF), which exploits social characteristics during data delivery. The PIF allows users to create personalized filters according to their social inter-

ests, and enables social friends to hold these filters, discarding the unwanted data before delivery. We also design privacy-preserving coarse-grained and fine-grained filtering mechanisms in the PIF to not only enable the filtering but also prevent users' private information included in the filters from disclosing to untrusted entities. Secondly, to detect misbehaviors during MSN data sharing, we propose a social-based mobile Sybil detection scheme (SMSD). The SMSD detects Sybil attackers by differentiating the abnormal pseudonym changing and contact behaviors, since Sybil attackers frequently or rapidly change their pseudonyms to cheat legitimate users. As the volume of contact data from users keeps increasing, the SMSD utilizes local cloud servers to store and process the users' contact data such that the burden of mobile users is alleviated. The SMSD also detects the collusion attacks and prevents user's data from malicious modification when employing the untrusted local cloud server for the detection. Thirdly, to achieve the trade-off between privacy and data availability, we investigate a centralized social network application, which exploits social network to enhance human-to-human infection analysis. We integrate social network data and health data to jointly analyze the instantaneous infectivity during human-to-human contact, and propose a novel privacy-preserving infection analysis approach (PIA). The PIA enables the collaboration among different cloud servers (i.e., social network cloud server and health cloud server). It employs a privacy-preserving data query method based on conditional oblivious transfer to enable data sharing and prevent data from disclosing to untrusted entities. A privacy-preserving classification-based infection analysis method is also proposed to enable the health cloud server to infer infection spread but preserve privacy simultaneously.

Finally, we summarize the thesis and share several open research directions in MSNs. The developed security solutions and research results in this thesis should provide a useful step towards better understanding and implementing secure and privacy-preserving MSNs.

Acknowledgements

The past four years of my PhD research at Waterloo are truly the most unique, precious and awarding time in my life. This thesis would not have been possible without the helps and supports from my supervisor, my thesis committee members, my colleagues and families.

First and foremost, my deepest and sincerest gratitude goes to my supervisor, Professor Xuemin (Sherman) Shen. It is his support, guidance, encouragement, patience and genuine expertise in the past four years that have made this dissertation possible. What I appreciate the most of Professor Shen is his great patience and understanding to me. I am really inspired by his dedication and enthusiasm to his work, his students and his family. In addition, I would like to thank Professor Kui Ren for serving as my thesis external examiner. I also appreciate the honorable members of my thesis committee, Professor Wei-Chau Xie, Professor Mahesh Tripunitara and Professor Liang-liang Xie. Their insightful comments have significantly affected the substance and presentation of my work.

Many friends and colleagues from Broadband Communications Research (BBCR) group have made my life at the University of Waterloo a colorful and enjoyable experience. I wish to especially thank Dr. Rongxing Lu, Dr. Xiaohui Liang, Professor Xiaodong Lin, Dr. Kan Yang, Ju Ren, Jianbing Ni, Professor Yaoxue Zhang, Professor Kan Zheng, Professor Hai Zhao, Dr. Henry H. Luo, Professor Zhou Su and Dr. Mrinmoy Barua for their inspiring discussions and invaluable insights on my research. I also wish to thank Dr. Tom H. Luan, Dr. Ning Lu, Dr. Ning Zhang, Nan Cheng, Nan Chen, Dr. Chunhe Song, Dr. Wei Jing, Ran Zhang, Dr. Miao Wang, Dr. Yong Zhou, Professor Zhiguo Shi, Dr. Haibo Zhou, Dr. Hao Liang, Dr. Yongkang Liu, Dr. Hassan Omar, Dr. Qinghua Shen, Dr. Jian Qiao, Dr. Ye Wang, Dr. Xiaoxia Zhang, Dr. Zhongmin Zheng, Dr. Chengzhe Lai, Qiang Ye, Miao He, Wenchao Xu, Dr. Amila P. K. Tharaperiya Gamage, Dr. Shibo He, Professor Mi Wen, Professor Yuanguo Bi, Professor Yi Zhou, Professor Juntao Gao, Professor Shaohua Wu and many others. I gratefully acknowledge all BBCR group members for their continuous encouragement, selfless help and all the good time we spent together.

The thesis is dedicated to my parents. I would not be completing my studies if they did not teach me the value of hard work and dedication. I owe them everything, and fear I cannot love them enough in return for that. Thanks to them all for their continuous and ever-caring support which made me always feel their presence so near to me.

Dedication

To my family and teachers from whom I have learned so much.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Mobile Social Networks

Online Social Networks (OSNs), as a kind of popular social networking platforms, allow users (even strangers) to interact with each other for information sharing and other social activities over the Internet. With the assistance of OSNs, people's social circle and community have been extended from their family, colleagues and friends to the Internet users having similar interests and preferences. In every minute of the day, 31.25 million posts are shared on Facebook with over 4.16 million post-likes; around $350,000$ tweets are generated on Twitter; over 300-hour-length new videos are uploaded on Youtube with more than 2 million views; and $2,400,000$ searches are operated on Google [1]. Offering these diverse social network services, OSNs have already become an integral part of people's daily life.

Meanwhile, with the advancement of wireless communication technologies and ever-increasing volume of smartphones, mobile social networks (MSNs) emerge to offer a novel social networking paradigm. MSNs are fueled with heterogeneous wireless communications (e.g., cellular, WiFi, BlueTooth and short range communications), mobile devices (e.g., smartphones and wearable devices) and powerful social network servers (e.g., cloud servers) to build a fantastic platform for users' ubiquitous social activities. In MSNs, smartphones take place of traditional desktop and allow users to have various types of social activities: from friend discovery to multimedia sharing; from universal content searching to local information query [2]; and from the global Internet to the physical proximity [3]. MSN users can not only select and download their interested multimedia contents or global up-to-date information over the Internet [4], but also share the local information directly to their friends or even strangers who have similar interests in vicinity of each other [5, 6]. In

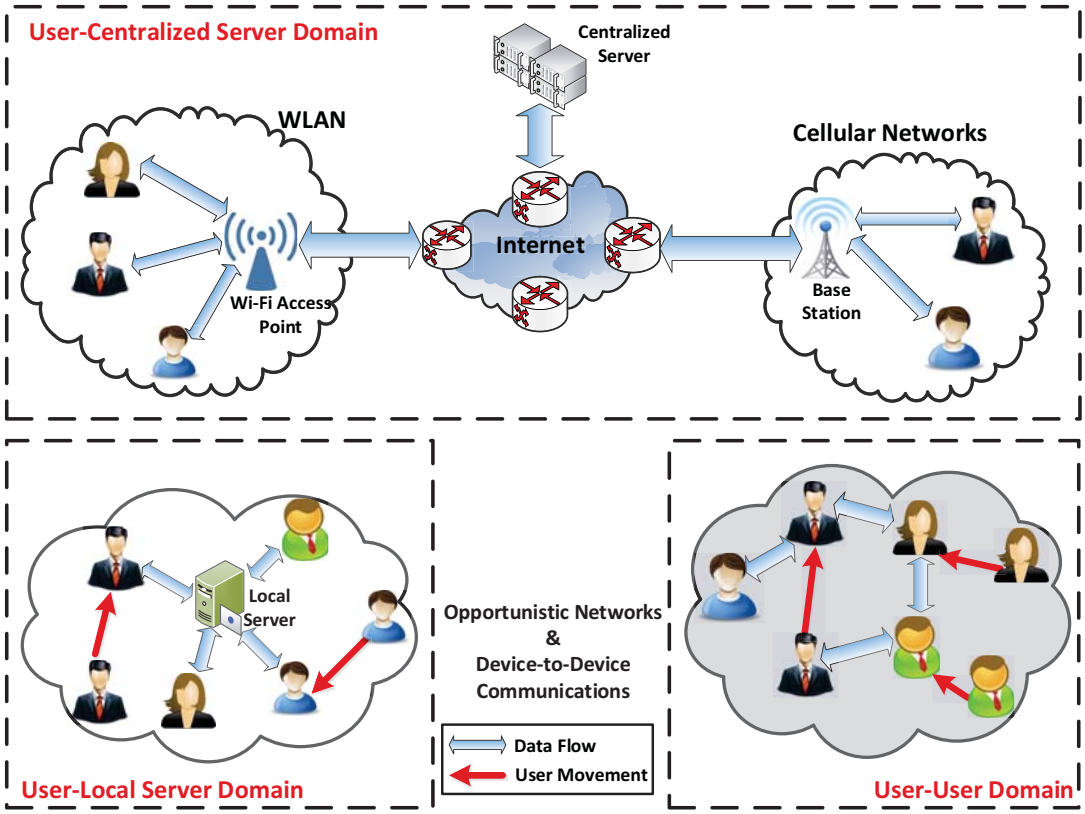


Figure 1.1: Mobile social network architecture

addition, a group of users are able to autonomously form a social community to share the personalized contents with opportunistic networks or device-to-device communications. In this chapter, we present the MSN architecture, applications and some challenges in MSNs.

1.1.1 MSN Architecture

MSNs consist of mobile users moving in a local geographical area, centralized servers connected by various types of wireless networks, and local servers deployed in the local area as shown in Fig. 1.1.

(1) *Mobile Users*

Mobile users take smartphones to communicate with each other. They either connect

to the Internet via WiFi/cellular networks or directly communicate with other nearby users via short range communication techniques, such as Bluetooth and NFC. These communication modes (or patterns) depend on the network conditions and requirements of different applications. For instance, when users search the Internet content, such as Youtube video clips, they should turn on the Internet mode if applicable on their smartphones and directly access the centralized servers to access the desirable contents. When some users are in the physical proximity, they are able to directly exchange their information via Bluetooth. In addition, mobile users could not only be the content owners, but also query contents from others. The contents of mobile users contain text (e.g., posts, microblog and news), audio (e.g., music), image, and video (e.g., movie or video clips).

(2) *Local Server*

The local server (LS) is a computing, communication and storage device/machine (e.g., router, small cell gateway, computer, smartphone or fog computing element) in the local area. The LS is able to offer nearby users with local service information, such as local tour information, store advertisements and service evaluation of the vendors. Local users' feedbacks, review comments or requests can be also collected by the LS. The LS either stores and processes a portion of this information, or acts as a relay between the centralized server and mobile users. The LS usually has storage-rich devices fueled by the adequate power. The contents from the LS are mainly related to the local information, such as service descriptions, advertisements, local introduction and tips.

(3) *Centralized Server*

The centralized server (CS) can be the Internet service provider or cloud server, providing the Internet-based services. The CS usually has very strong capabilities of storage, communication and computing compared with the LS. Mobile users access the CS either via WiFi and cellular networks, or through relaying by the LS. The contents from the CS are usually abundant and of diverse types, since the CS have the Internet connections with the world-wide content resources.

1.1.2 MSN Communication Patterns

In general, MSNs can be classified into three domains, i.e., User-CS, User-LS, and User-User domains (as shown in Fig. 1.2) according to different communication patterns and various types of contents.

(1) *User-CS Domain*

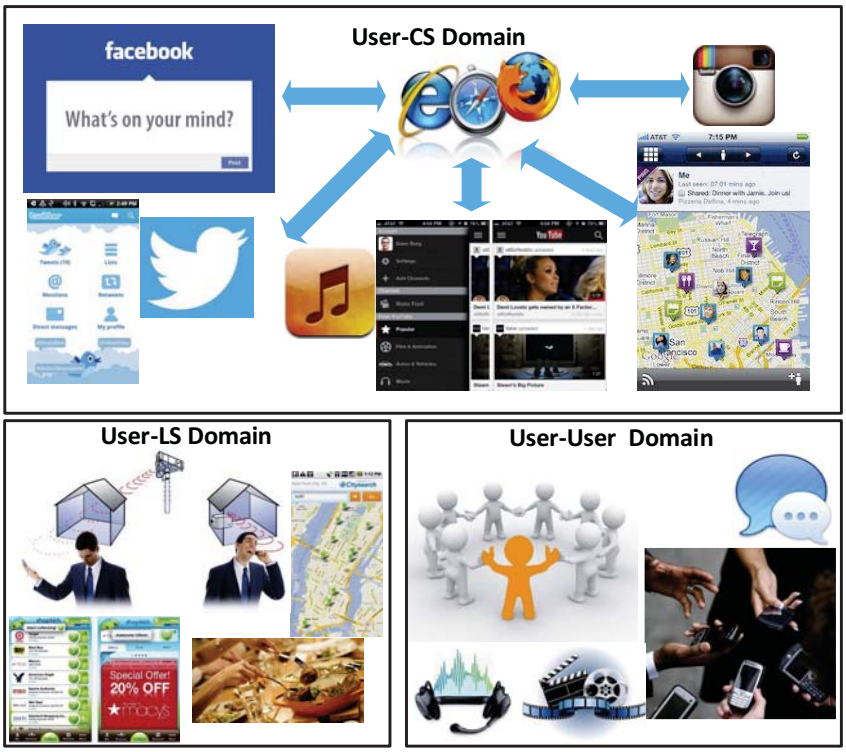


Figure 1.2: MSN domains: User-CS domain, User-LS domain, and User-User domain

In User-CS domain, users directly access the contents from the CS via either cellular networks (with the purchased mobile data plans) or WiFi access points, which are widely deployed in the residence area and public spots [7], such as campus and local stores. For example, Tim Hortons, one of the largest publicly-traded restaurant chains in North America, has rolled out the high-speed free WiFi service to customers since 2012. The communication range is dependent on the type of communications and network infrastructure. The connections in User-CS domain may be one-hop or multi-hop. The contents communicated within User-CS domain contain a broad range of multimedia all over the world since the CS holds diverse types of contents. In User-CS domain, users can browse the social media, photo galleries and online video; query the desirable contents over the Internet; and share their multimedia contents to others.

(2) *User-LS Domain*

In User-LS domain, the LS acts as not only a temporary local organizer equipped by the easy-to-setup and low-cost local wireless gateway or router, but also a mobile user participating in the mobile user's social interactions. The LS may have the capability to access the Internet, or establish local distributed or autonomous MSNs among neighboring users. The LS can also disseminate the contents to the near-by mobile users with a longer communication range compared with mobile users. Since the User-LS domain is featured by the local attributes, the LS provides multimedia services, local guidelines, advertisements, and local customers' reviews to help users better understand the features of the local area.

(3) *User-User Domain*

MSN users sometimes stay in the mobile local environment, where the continuous Internet services may not be guaranteed or users could directly exchange contents with each other in the physical proximity without the Internet. The User-User communications plays an uppermost role in such a scenario. For example, in a shopping mall, commercial street, etc., users with similar social preferences may want to share their multimedia contents to others. They can adopt opportunistic network, device-to-device communications, Bluetooth, etc., to establish the temporary connections to enable data sharing without the Internet and cellular network infrastructure. The communication range is usually from 1-200 meters in the local area, and multi-hop communications is applied according to users' demands. Most of contents in User-User domain are personal contents, such as personal status update, local information generated by users, etc.

1.1.3 Applications of MSNs

The main objective of MSN applications is to share information among mobile users by using wireless and mobile communications technology such that the closeness of social relationships is enhanced. In this section, we introduce some popular MSN applications.

Social Networks: From Desktop to Mobile

As the flourish of OSN applications, a large number of people are interested in exchanging their experiences with their friends over the Internet. OSNs, such as Facebook, Twitter, Linkedin and Wechat, gradually become an integral part of our lives. Knowing people and becoming friends are the primary motivations for these OSN services. According to a recent report from comScore, Instagram users in the United States spend 98% of time with their mobile devices instead of desktop, while this percentage for Twitter users is over 86% [8]. With the help of smartphones, people can start these applications at any time and

anywhere. The MSN services introduce the freedom of movement for users, and provide ease of use and seamless connection to social world. For example, a student can browse her Facebook friend updates and photos when taking a bus back home; a business man can process his emails when he is not in the office; a soccer fan can obtain the up-to-date game information when he is in a shopping mall. As a result, a promising tendency is to establish a social network in a mobile or distributed fashion where people use smartphones to communicate with the local neighbors for some shared interests, even though people might not know these neighbors in reality.

The applications in User-CS domain include traditional content query, data downloading, information exchanging, and social interaction for online social communities. With the available cellular data plan or widely applied WiFi access point, the Internet access is usually guaranteed. Therefore, it can support the wide range content query, large size data downloading, and real-time information exchanging.

Location-Based Services

Location-Based Service (LBS) is a popular social networking application with the assistance of GPS and some other sensors embedded in smartphones. The location information can be widely used and combined with user's social information to provide users a variety of contextual services and personalized searching services. LBSs can find the nearest restaurants, discover friends in the proximity, recommend information, such as social activities, location-based advertisements, games, etc. For example, Google Latitude [9], Loopt [10], and Foursquare are some of the popular location-based MSN services. Some of them are developed on the smartphone platform, such as iPhone, Android. Meanwhile, some applications, (for example, APPLAUS [11], secure top-k query [12]) enable LBS and preserve user's location privacy from directly disclosing to other untrusted entities in MSNs.

User-User applications

There are some autonomous MSN applications [13, 14] applied for the direct interactions among mobile users. The extension of the popular online services (e.g., Facebook, MySpace) to mobile devices accounts for wide popularity of mobile social networking services. For example, Dada [13] is an MSN application that enables users to update personal blogs with pictures and video, connect with and meet other local users in real time and stay in contact with all their friends even without the Internet. Carpool and ride sharing are popular MSN applications, which alleviate the heavy road traffic and offer profits to both

the driver and the passengers. With the temporal and geographic vicinity of the departure and arrival, the ride sharing provider (i.e., the driver) can save travel expense, such as fuel and parking costs, while the passengers pay little money for the journey. The similar travel plan is the common social preference of the ride sharing provider and passengers.

In addition, some wearable devices, such as Apple watch, Google glass [15], bracelets, Hug shirt [16], etc., can measure the environmental information or monitor the biomedical condition and health parameters. Users can interact with their body, for example, hugging and heating human body with Hug shirt. Therefore, the User-User applications lead to a new tendency of MSNs and provide diverse services.

1.2 Characteristics of MSNs

As discussed above, the main objective of MSNs is to enlarge users' social circle and enable their social interactions anytime and anywhere. Different from the traditional OSNs, MSNs are featured by: 1) dynamic user mobility and intermittent connections; 2) lack of central controller to manage the large scale network; 3) the limited resources, such as bandwidth of MSNs, user's computing, communication, and storage capabilities; 4) user's profiles are highly related to their privacy; and 5) security vulnerabilities. All these features dramatically complicate the design of application and network. In this section, we investigate several important characteristics of MSNs.

1.2.1 Social Characteristics

Social characteristics are of paramount importance in MSNs, since these social features may have impacts on user's mobility, networking connectivity, communication patterns, security, privacy and trustworthiness [17]. Social network analysis [18] has also attracted considerable attentions in many research fields such as anthropology, biology, communication studies, economics, information science, computer science and engineering. Regarding MSNs, we investigate several typical social characteristics, including community, centrality and similarity, friendship and selfishness.

Community represents a group of interacting users living in a common location or users who have common interests in certain aspects. It is widely studied at spatial and temporal scales. Community is introduced to Delay Tolerant Networks (DTNs), Vehicular Ad hoc NETworks (VANETs), and MSNs, in order to improve the network efficiency [19].

Centrality is a metric to measure the topological importance of a vertex within a social graph [20, 21]. Generally, the central node in the social graph has a stronger capability to reach/connect other nodes compared with that exist near the graph boundary. Centrality reflects the social importance of the node, and would be used to select the group or community leader.

Degree, as another metric of the social graph, is defined as the number of common neighbors between individuals in a social graph. It is extended to measure the social interests, location, etc.

Social-tie indicates the strength or characteristics of the link connecting two users. Usually, social-tie can be reflected by contact duration, frequency and some other factors that identify the connection strength of two users.

Friendship reflects the strong social-tie of a pair of users, including common interests, long-lasting and regular contacts, etc.

Selfishness is originated from sociology and economic. In MSNs, selfish users behave selfishly at individual level and aim to only maximize their own profits without considering the utility of the whole network. For example, some users behave selfishly and may not be willing to help forward other user's packets. Selfishness of mobile users may impact the cooperation and disrupt MSNs.

1.2.2 Network Characteristics

Data sharing is one of the most popular and significant applications in MSNs, depending on data forwarding performance of the network. Some traditional forwarding schemes usually rely on the shortest forwarding path, minimized the forwarding delay, energy consumption and some other defined metrics of forwarding. However, these types of forwarding schemes do not consider the social characteristics or relationships among MSN users. For example, during the social data sharing, the users with high social ties can share the data related to certain social attributes, while strangers may not be willing to help forward or share the data. The data forwarding is social-relationship-driven and towards certain social communities or a group of social friends. In addition, MSNs are sometimes deployed in local or urban area. On one hand, the Internet access may not be always available for all users due to the deployment of WiFi access points and environment condition; on the other hand, the high cost of data plan or roaming may hinder some mobile users to purchase. As a result, the Internet connections among mobile users or servers may be intermittent. Due to the intermittent connectivity, the communications in MSNs may be constrained and delay-tolerant, especially in the local area or during temporary social

events. To maintain the connections among mobile users, it is necessary to investigate user's social characteristics. Since the users with similar social interests or preferences would encounter at the same location, it is possible to explore opportunistic networking and take the advantages of mobile user's contact to improve the data forwarding efficiency in MSNs. Furthermore, users with similar social preferences can probably help their social friends to store-carry-and-forward the data and cooperatively share the data among social friends. In addition, some negative social factors, such as selfishness, should be addressed to improve the information sharing efficiency in MSNs. If the user's historic behaviors are collected to analyze and detect the selfish behavior, the user's privacy is easily disclosed. Therefore, it is challenging to balance the trade-off between the performance and privacy.

1.2.3 Security and Privacy in MSNs

Although MSN applications become popular in our daily life, security and privacy concerns still hinder the flourish of MSNs. In this section, we introduce several critical security threats and privacy leakage.

Security Threats

General security requirements, such as confidentiality, integrity, non-repudiation and access control, are applicable to MSNs. Besides, MSNs are vulnerable to a series of security threats, such as forgery, tampering, spam and Sybil attacks.

(1) *Forgery*: Malicious attackers may not only forge their identities, profiles and social relationships, but also generate fake information to misbehave or mislead other users. MSNs cannot effectively identify these misbehaviors. In addition, the network resources, such as bandwidth, storage and energy would be excessively consumed based on the forged information.

(2) *Tampering*: A tampering attacker could maliciously drop, delay or modify the transmitted data to disrupt MSNs and degrade the network efficiency. It is difficult to detect some tampering behaviors since the wireless channel condition and user mobility may also result in the transmission failure and delay [22].

(3) *Spam*: Spam data refer to the unwanted content, such as comments, chat, news and links, which are generated and spread by the attackers. The spam would result in the unnecessary network resource consumption, misleading social friends, and even privacy leakage.

(4) *Sybil*: Sybil attackers either manipulate fake identities or abuse pseudonyms in order to compromise and control the effectiveness of MSN. They could generate incorrect reports and social content such that users' opinions and options may be misled. In addition, Sybil attackers could link legitimate user's private information.

Privacy Leakage

As the MSN applications are highly related to user's social interests, relationships, and some other social preferences, directly revealing this information would violate user's privacy.

During social interactions, MSN users aim to share different types of data with other users. These shared data may contain identity, social interests, relationships and some other privacy-sensitive information, which is visible to the interacted users. Moreover, the unconscious disclosure of the private information may bring negative experiences to users and hinder the flourish of MSN, especially when users interact with strangers. In addition, during the packet forwarding, users may want to share their social information (or profiles) to acquire optimal relay candidates. These behaviors bring serious privacy problems and unconsciously disclose user's private information. Usually, users are not be willing to disclose their private information, such as preference and history activities, to those who do not have similar experiences. In other words, they would like to merely share such relevant profiles to others having common interests or events. Meanwhile, some malicious attackers might exist in MSNs. They could camouflage themselves with some fake profiles (e.g., camouflage/act as legitimate user's social friends) to interact with these legitimate users to steal their private information. These malicious users could also claim to have some social interests and acquire trust from legitimate users with the same social interests. Even worse, an attacker may collude with other attackers to reveal, link and infer legitimate user's information, even the user's information cannot be directly disclosed to the single attacker.

With these critical privacy concerns, users may not be willing and active in using MSNs for data sharing. In summary, it is necessary to develop privacy-preserving solutions in MSNs to prevent user's privacy-sensitive information from disclosing and improve the network efficiency simultaneously.

1.3 Research Motivations and Contributions

These emerging trends motivate our research in investigating the not-for-profit global initiative of security and privacy for MSNs. The research in this thesis focuses on developing

a set of security protection schemes to address the aforementioned security and privacy challenges in MSNs. Specifically, the main contributions are three-fold as follows.

- *Social Based Spam Filtering*: As the advertisements, rumors, and spams spread in MSNs, it is urgent to filter spams before they arrive at the recipients to reduce the network resource consumption, especially in autonomous MSNs (i.e., User-User domain). To this end, we propose a personalized fine-grained spam filtering scheme (PIF), which exploits social characteristics during message delivery. The PIF allows users to create personalized filters according to their social interests, and enables social friends to hold these filters, discarding the unwanted messages before delivery. In addition, the distributed filters may contain certain private information related to filter creators. If this private information embedded in filters is disclosed to others, it may violate the filter creator's privacy. To this end, we also propose privacy-preserving coarse-grained and fine-grained filtering schemes to not only enable the filtering but also protect users' private information included in the filters from disclosing to untrusted entities.

- *Mobile Sybil Detection*: Mobile Sybil attackers with a large number pseudonyms bring severe security threats in MSNs by frequently changing pseudonyms in a short period and repeatedly misbehave to (or cheat) normal users. It is difficult for mobile users to detect mobile Sybil attackers due to several limitations, including the lack of social graph information, user's dynamical-changing mobility, and limited detection capabilities. Moreover, mobile Sybil attackers sometimes act as normal users and merge into the normal user's crowd or social community, posing challenges for traditional Sybil detections. To address these challenges, we propose a social-based mobile Sybil detection scheme (SMSD). The SMSD detects mobile Sybil attackers according to the abnormal social contact and pseudonym changing behaviors in MSNs. As the volume of contact data from users keeps increasing, the local cloud servers (i.e., in User-LS domain) are adopted to store and process the users' contact data, alleviating the burden of mobile users. However, the untrusted cloud servers pose critical security and privacy concerns, such as data modification and deletion. The SMSD can prevent user's data from being modified and deleted by untrusted cloud servers. In addition, the SMSD detects the collusion attacks to degrade the attacker's capabilities and improve the detection accuracy.

- *Privacy-preserving Social Network Data Analysis*: We investigate privacy-preserving social network data analysis in a practical applications, i.e., infection spread analysis. We exploit social network associated with health data to analyze the instantaneous

infectivity during human-to-human contact. However, users' health and social data, such as infection status and social contact, are privacy-sensitive from the perspective of users, who are not willing to excessively reveal this private information to the untrusted or unauthorized entities. To preserve user's privacy, they may encrypt data and send the ciphertexts to cloud servers, limiting the data processing capability of cloud servers. In addition, social network data also contain privacy-sensitive information of infected and susceptible patients, such as identities and contact, which may be inferred by untrusted entities during the data sharing among different parties. To tackle these problems, we propose a privacy-preserving infection analysis approach (PIA), achieving the trade-off between data privacy and availability. The PIA enables the collaboration among different cloud servers (i.e., social network cloud server and health cloud server) in User-CS domain of MSNs. It employs a privacy-preserving data query method based on conditional oblivious transfer to enable data sharing among different entities. A privacy-preserving classification-based infection analysis method is also proposed to enable the health cloud server to infer infection spread and achieve data privacy.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 presents background and a comprehensive overview of security and privacy challenges in MSNs. Chapter 3 develops a social-based personalized filtering scheme with privacy preservation to resist spam in autonomous MSNs. Chapter 4 investigates users' social contacts and pseudonym changing behaviors to differentiate Sybil attackers from normal users. Chapter 5 exploits social network to enhance the infection spread analysis and protects privacy-sensitive information from disclosing to untrusted entities. Finally, Chapter 6 concludes the thesis, and points out our future research directions.

Chapter 2

Background

This chapter introduces the background of security and privacy for MSNs. We first present general security and privacy requirements. Then, we discuss several unique challenges of MSNs from the perspectives of security and privacy.

2.1 Security and Privacy Requirements

There are several general security and privacy requirements [23] should be satisfied in MSNs.

(1) *Integrity* should be ensured such that the data transmitted, shared, stored and processed over MSNs are accurate and complete representations of the intended information. These data should not be tampered in any way during any phase.

(2) *Confidentiality* should be guaranteed such that the data from users and service providers are invisible or unavailable to the unauthorized or untrusted entities. In other words, only the authorized MSN users can access the required data.

(3) *Availability* should be achieved when authorized users require certain data from MSNs. Ensuring availability also contains resisting denial-of-service attacks, jamming attacks, etc.

(4) *Authenticity* should be provided such that any involved entity requesting access in MSNs is valid and authentic. In MSNs, the information provided by the users or from social network service providers should be authenticated. Besides, each user and his identities should be verified as well. Any invalid information and user can be detected.

(5) *Privacy* ensures that any privacy-sensitive information, such as data, identity and location, should be prevented from being disclosed or inferred by any untrusted or illegal entity, including active and passive attackers.

(6) *Non-repudiation* resists the repudiation threats where attackers deny after performing certain behaviors in MSNs. For example, users who send spams may deny the spam sending behaviors; a local service provider may deny the offered services to customers. MSNs should be able to detect these repudiation threats.

(7) *Access Control* is to enforce access policies and ensure that only authorized users can have access to resources in MSNs. As users' private social network data are stored in the cloud sever, they should be able to define access policy.

(8) *Anonymity* guarantees that a user cannot be identified by unauthorized entities. The user's real identity should be anonymous when he stores his data on the LS or CS such that both LS and CS cannot learn anything about the identity.

(9) *Unlinkability* refers to the activities or use of MSN resources by a user without other users being able to interlink the activities and usage of these resources. Specifically, the information obtained from different flows over MSNs should not be sufficient to establish linkability by the unauthorized entities [24].

(10) *Auditing* ensures that all the data over MSNs are secure. All the data access activities are perceived and recognized by a trusted third party in MSNs.

2.2 Security and Privacy Challenges in MSNs

Besides the aforementioned general security and privacy requirements, several unique challenges in MSNs are crucial and require more research efforts to address.

2.2.1 Privacy Leakage During Social Interactions

Privacy leakage is a critical issue in MSNs when the privacy-sensitive information is involved in the data collection, transmission, processing and sharing. Without appropriate protections against privacy leakage, users may not be willing to expose their data visible to any untrusted entities. It may hinder the processing and sharing of users' social network data and their experiences. For example, a user suffering HIV/AIDS do not want other people to know his disease when using social network or forums. The inappropriate or unconscious information leakage may leave negative impacts on this user. Even worse,

users sometimes may not be aware of privacy leakage via MSNs, which would cause finance loss. In February 2016, a Newfoundland woman posted a picture of her "Roll Up the Rim" prize-winning Tim Hortons cup via Facebook. But one of her 900 Facebook friends stole the security code on the peeled-back rim of her cup in the online posted picture, and then claimed the $100 prize ahead of her. Another example is about training and fitness social applications. Soccer players in a game can wear a set of dedicated vests to measure the player's body condition and performance for coaches to determine the game strategy. Once the sensed player-related information is disclosed to the opponent, the opponent could change the strategy in advance. Therefore, privacy should be preserved in MSNs to provide user-friendly services.

In recent years, privacy preservation receives a lot of attentions in the research field. Several critical privacy threats in social related applications are introduced in [25, 26], where identity privacy, information leakage during transmission and location privacy are investigated. In [27], the privacy protection is applied between wearable devices and smartphones to protect wearable sensing data from disclosure in health-related social network applications. In [28], Ong et al. investigate the security services partitioned into various security levels to balance the trade-off between security and performance (with respect to computation, storage and communication overhead) preferences.

When the social network data are transmitted or shared in MSNs, privacy should be also taken into account when developing applications. Privacy-preserving aggregation is a promising way to gather the transmitted data. Shi et al. [29] propose a privacy-preserving aggregation scheme for time series data. The data are divided to mix them together and restrict the aggregator's decryption capability. In [29], the aggregator can only decrypt the summation of the aggregated data without learning anything about individual data. In [30], Lu et al. propose a multi-dimensional data aggregation scheme based on increasing sequence to reduce the computational and communication overhead during the aggregation. In [31], another privacy-preserving aggregation scheme is proposed to support a variety of statistical additive and non-additive aggregation functions. Moreover, this scheme is featured by resistance to the collusion attack during aggregation. To improve the robustness of privacy-preserving aggregation, Chan et al. [32] consider the fault tolerance problem during aggregation. A trusted authority assigns N capabilities to an aggregator, corresponding to the N users. By using a binary tree, this fault tolerant aggregation scheme forms several groups of users to improve the robustness. However, the communications between users and aggregator still consumes massive overheads.

2.2.2 Privacy Leakage During Data Processing

The volume of social network data keeps increasing such that the powerful cloud servers are involved in data storage and processing. When the social network data are outsourced to the cloud servers for processing and analysis, users may not want to reveal their raw data to the untrusted cloud servers. It is necessary to keep users' raw data invisible to the untrusted and unauthorized entities. In addition, users' private information, such as identities and personal profiles, should be kept anonymous to the untrusted and unauthorized entities according to users' privacy requirements.

A intuitive methodology to keep the private data from MSN users invisible to untrusted entities is to encrypt this private data before sharing. However, the encrypted data may hinder the processing or increase the overhead of cloud servers. To this end, several secure multi-party computation schemes (e.g., homomorphic encryption and functional encryption) have been proposed to preserve data privacy during operations, such as summation, comparison and aggregation [33, 34]. With various privacy requirements in MSN applications, the protections should be also correspondingly enhanced when the trusted entities conduct complicated operations to analyze inherent features over MSNs, e.g., data mining and machine learning [35]. Yuan et al. [36] propose a collaborative learning scheme, which enables each user to encrypt his data and upload the ciphertext to the cloud server. The cloud server performs most of the learning algorithms over these ciphertext without learning the plaintext. A variant of "doubly homomorphic encryption scheme for secure multi-party computation is adopted to perform flexible operations over the encrypted data. Bost et al. [35] develop a set of secure machine learning classification algorithms and propose a library of components, validating the feasibility of machine learning over encrypted data. When using homomorphic encryptions for machine learning and data mining, a large amount of computational and communication overheads is generated. It may also considerably reduce the battery lifetime of users' smartphones and wearable devices. The increasing computational and communication overheads can increase the delay of social network data analysis. Therefore, it is challenging to balance the trade-off between availability of data analysis and privacy preservation when performing machine learning and data mining over encrypted social network data.

2.2.3 Social Network Data Access Control

MSNs may take the advantages of the powerful storage and computation capabilities from the outsourced cloud servers such that security concerns related to these untrusted cloud servers [37] are raised. The data access policy should be clearly defined and applied to

authenticate the user's identity with access authority. For example, wearable devices can measure user's daily health condition, e.g., Electrocardiography (ECG), which is stored in the untrusted cloud server of health-oriented social networks. Only the authorized entity, such as doctors in the neurology subject, can access these data and the corresponding analysis results. Meanwhile, the data should be protected from being accessed by insurance company and any other untrusted entities [38]. Besides the general access control policies, it is still critical to ensure the fine-grained access in accordance to users' attributes. To this end, Yu et al. [39] propose fine-grained access control for cloud storage to prevent users' sensitive data from disclosure to other untrusted servers and unauthorized users. This fine-grained access control scheme is based on ABE technique which associates the data access policy with attributes. It delegates the majority of computations to the powerful cloud servers such that users' overheads are considerably reduced.

In MSNs, the dynamic access management is necessary to address the issues of users' attribute changing, revocation, new user's participation, etc. Delegation of access control is another important issue of access control in MSNs [4, 40]. For example, a user may obtain the access to a portion of data from data owner according to their similar social interests. This user can share the data owner's data to another user if all the users have similar social interests with the data owner. The access control should achieve both delegation and resist the collusion attack. The private information, such as unique/uncommon social interests, should be also protected in access control [24]. Least but not last, the computation overheads of access control schemes [41] should be considered for different MSN applications. With a large number of attributes used in attribute-based access control, the encryption and decryption overheads may correspondingly increase. Towards different access levels, the computation burden of users should be released.

2.2.4 Misbehaviors and Malicious Attacks

MSNs, as a type of promising paradigms expanding the traditional Internet to the ubiquitous network [42], connect wearable devices in the physical world, smartphones in the social circle and servers in the information world [43, 44]. From the perspectives of network and misbehaving characteristics, MSNs can be also divided into sensing domain, social domain and mobile domain as shown in Fig. 2.1. Furthermore, by integrating the sensing, communication and computation capabilities [45], MSNs can offer diverse intelligent services [46] to form smart home [47], smart community [48] and smart city [49, 50] as shown in Fig. 2.1.

However, the emerging MSN is vulnerable to a series of malicious attacks, such as Sybil attacks where fake identities are manipulated [51, 52] to compromise the effectiveness of

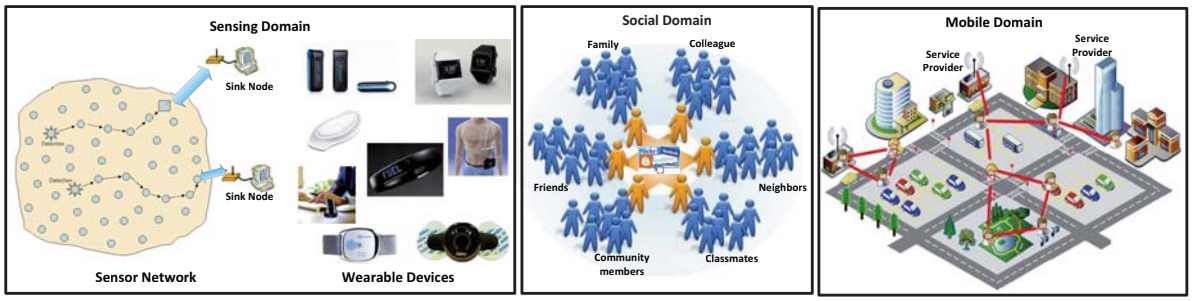


Figure 2.1: MSN domains: sensing domain, social domain, and mobile domain

MSNs. In the presence of Sybil attacks, incorrect information may be generated over MSNs, while users may receive spam and reveal their private information. An investigation report [53] in 2012 has shown that a substantial number of registered accounts are identified as fake or Sybil ones in OSNs. Around 76 million (8.7 percent) accounts are Sybil in Facebook, while 20 million newly-created fake accounts join Twitter during every week. These Sybil accounts may broadcast advertisements and spam, or even disseminate fishing websites and malware over OSNs in order to steal other users' private information. In MSNs, Sybil attackers also produce various biased information options with "legible" accounts or pseudonyms [54]. Without an effective detection scheme, the collected information over MSNs may be manipulated by Sybil attackers. Since most of Sybil attackers have similar behaviors as normal users, it is difficult to detect them.

We define Sybil attacks in three types. Before introducing each type of Sybil attacks, we present the social graph model, which is a useful tool to analyze social network in general. Consider an undirected social graph, which is denoted by $\mathcal{G}$ with n honest nodes (H) and m edges. Sybil nodes are denoted by S. Note that we use node in social graph to represent user, identity, or account in the real world. The edge connecting every pair of two nodes is weighted by their social relations or social-tie. An attack edge AG is the edge connecting an honest node and a Sybil one, i.e., red dashed line as shown in Fig. 2.2.

SA-1 Attacks

SA-1 attackers usually build connections within the Sybil community as shown in Fig. 2.2. In other words, Sybil nodes tightly connect with other Sybil nodes. However, they are not powerful to build many connections with honest nodes. Therefore, the number of social connections between Sybil and honest nodes is limited. As shown in Fig. 2.2, the number

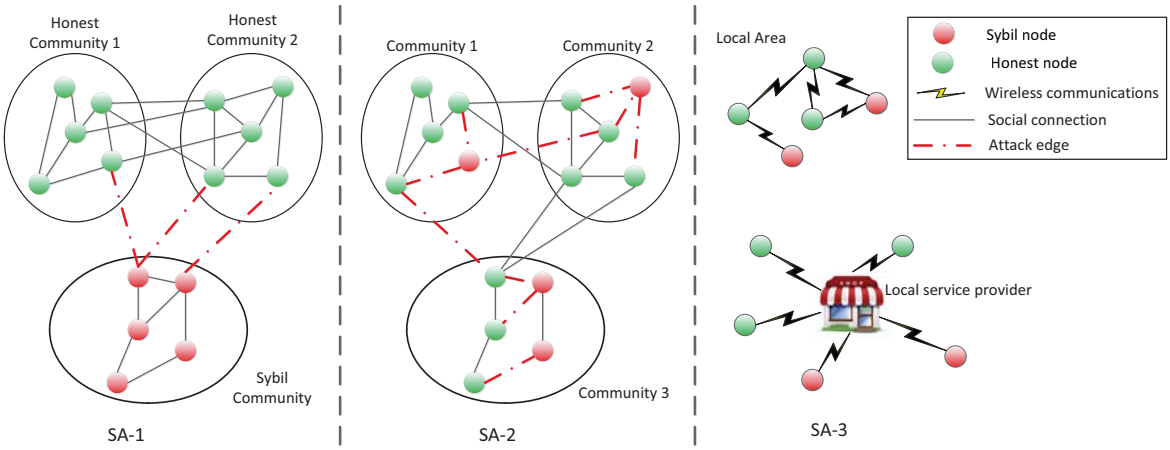


Figure 2.2: Three types of Sybil attacks: SA-1, SA-2 and SA-3.

of SA-1 attack edges is limited.

SA-1 attackers usually exist in sensing domain and social domain, i.e., OSN, voting [55] and mobile sensing systems [56]. The primary goal of SA-1 attacks is to manipulate the overall option or popularity. Specifically, in an online voting system, SA-1 attackers may illegally forge a massive number of identities to perform as normal users and submit the votes with biased options and preference. As a result, the final voting result could be manipulated by SA-1 attackers, provided a large portion of votes are from themselves. Similarly, in mobile sensing system, SA-1 attackers may forge the false sensing data and indirectly impact the aggregated data. In some cases, the behaviors of Sybil attackers are indistinguishable from the normal users.

SA-2 Attacks

SA-2 attackers usually exist in social domain. Different from SA-1, SA-2 attackers are able to build connections with both Sybil nodes and normal ones. In other words, the capability of SA-2 attackers is strong to mimic the normal user's social structures from the perspective of social graph. The number of attack edges is large.

The primary goal of SA-2 attack is to disseminate advertisements, spam and even malware; violate users' private information; and maliciously manipulate the reputation system. Specifically, in OSNs, SA-2 attackers may forge the profiles and friend list as normal users, but purposely spread spam, advertisements and malware. SA-2 attackers

Table 2.1: Three Types of Sybil Attacks

Categories of Sybil Attacks	Social Graph Features	Attack Goal	Behavior Discrimination	Mobility
SA-1	Sybil attacks exist in the same region or community, and the number of attack edges is limited.	Maliciously or purposely upload the biased reports or comments (positive or negative) to manipulate the overall option and dominate the whole system.	Perform as the normal users, and repeat specific behaviors frequently.	×
SA-2	Sybils may tightly connect with normal users, and generate more attack edges.	Disseminate spam and malware to launch some other attacks, camouflage as normal users, or violate other users' privacy.	Purposely repeat some specific behaviors in the high frequency.	×
SA-3	Sybils may tightly connect with normal users.	Manipulate the local popularity, disseminate spam in the mobile environment, or violate user's privacy.	Repeat specific behaviors frequently.	√

may also post plenty of biased review comments when evaluating MSN services such that either the advantages of services may be exaggerated or the services may be underestimated according to the negative comments to services. It can be observed that SA-2 attacks aim to repeat them in a high frequency [57].

SA-3 Attacks

SA-3 Sybil attackers exist in mobile domain. The primary goal of SA-3 is similar to that of SA-2. However, the impacts of SA-3 are usually effect in the local area and within a short period. Due to the dynamic mobility of MSNs [58, 59], mobile users cannot keep connections with others for a long time, and the connections are intermittent. Moreover, the centralized authority cannot exist in mobile domain at all the time. Different from online systems, MSN users lack strong social relationships such that global social structure, topology and historical behavior patterns in mobile domain is hard to be collected for the detection on SA-3 attackers. The limited knowledge of the global information and the dynamic mobility of mobile users pose challenges to detect SA-3 attacks compared with

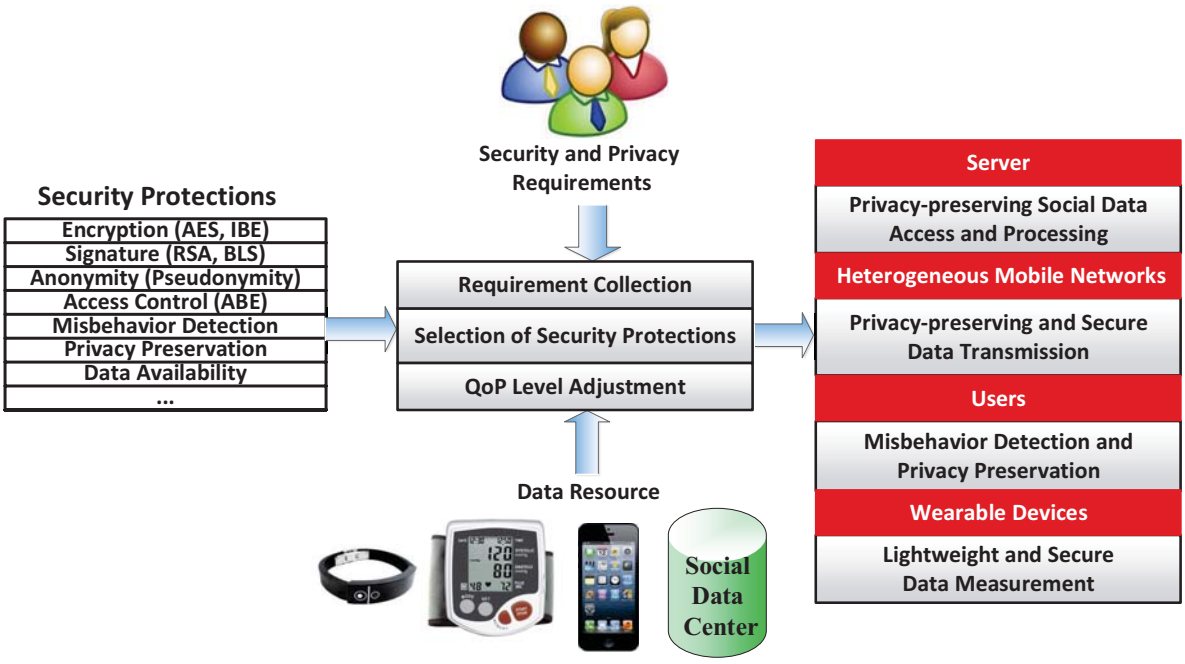


Figure 2.3: Quality-of-Protection in MSNs

the detections against SA-1 and SA-2 attacks. In Table 2.1, we compare different types of Sybil attacks.

According to the aforementioned discussion, Sybil attackers can misbehave in different patterns and mimic legitimate users such that Sybil detection becomes more challenging in MSNs.

2.2.5 Quality-of-Protection

With the main driver of user's experiences [60] and security requirements, Quality-of-Protection (QoP) has attracted extensive research attentions [61]. As an important security concept, QoP can provide multiple levels of security protections to satisfy various application requirements and user's demands [62, 63]. As shown in Fig. 2.3, MSNs with QoP can guarantee the confidentiality, integrity and non-repudiation via encryption and signature; achieve access privileges via authentication; ensure the copyright via watermarking; protect user's privacy via other cryptographic schemes (e.g., anonymity and obfuscation

techniques) [60]. With a set of security protections, QoP adjusts these tunable protection solutions, and is fueled by artifacts, human intelligence and involvements. A proper QoP construction can be offered by the characterization of QoP with security settings, where it expresses security constraints and attributes to customize protections for different applications [64]. In MSNs, to achieve a higher privacy level for users' data and profiles, security and privacy protections should be robust to resist the potential attacks and privacy leakage such that the computational overheads and latency are inevitably increased. Besides many off-the-shelf security protection solutions [64], other emerging schemes should be developed from QoP perspective to address critical security and privacy issues in MSNs.

2.3 Summary

In this chapter, we have discussed security and privacy challenges in MSNs. First, we have introduced general security and privacy requirements for MSNs, such as integrity, confidentiality, authenticity, availability, privacy, non-repudiation, access control, anonymity, unlinkability and auditing. Besides these general security requirements, we have discussed several challenging issues related to privacy, social data access control, privacy-preserving data processing, QoP, malicious attacks and misbehaviors in MSNs. In the following chapters, we will introduce several countermeasures to address the critical challenging issues.

Chapter 3

Social Based Spam Filtering

3.1 Introduction

MSN has become a promising social networking platform which enables group chat, media sharing, social gaming, and various pervasive social interactions, especially in a local area [26]. Users connect to each other through short range communication technologies (e.g, Bluetooth, WiFi and device-to-device communications), and establish a kind of opportunistic network for a temporary period (e.g., several hours) or a long span of years. This autonomous MSN (i.e., User-User domain) creates rich interaction opportunities for students in a campus area, residents in an urban neighborhood, customers in a shopping mall, tourists visiting a museum or scenic site, and businessmen attending a conference. In the User-User domain of MSNs, users' interactions are enabled either by the Internet or through opportunistic contacts among users to store-carry-and-forward data from source to destination. As we can imagine, users would have abundant and quality service experiences from MSNs [65], helping users obtain the desired and personalized information from others (e.g., crowdsourcing) rapidly, efficiently and ubiquitously.

MSN users exchange various types of information, such as newsletters, personal posts, rumors and advertisements, most of which are of immense values to users. As shown in Fig. 3.1, local stores or restaurants repeatedly disseminate their service information, flyers and advertisements to the nearby users in such an autonomous MSN. A saving mom may prefer the coupons, grocery sales and baby stuffs, while a tourist may be interested in handicrafts and tour instructions. In addition, the interests of users may vary over time. Although users could quickly exchange useful information in MSNs, they may still receive a portion of the unwanted or useless information, which is considered as spam [66]. Moreover,

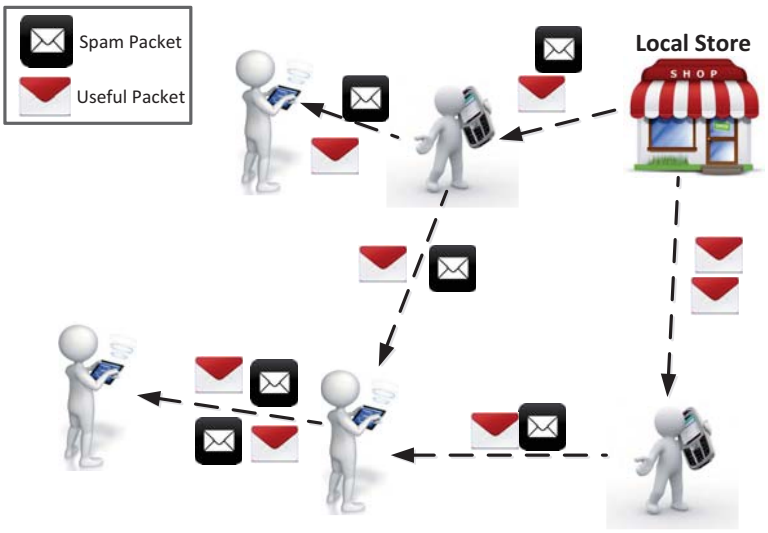


Figure 3.1: Information dissemination in MSNs

the communications among users relies on users' battery-constrained smartphones and happens during their opportunistic contact such that the communication overhead is very high. Therefore, it is crucial to make the communication efficient and meaningful in MSNs, i.e., exchange desired information to users and filter spam as early as possible.

According to an investigation by Nexgate, spam over social media has increased around 355% within the first six months of 2013 [67]. They are rapidly spread in social networks such that every 1 of 200 social network posts is identified as spam. Extensive industrial and research efforts have been put on filtering spam in various applications. Several schemes rely on blacklist [68] or whitelist to either block spam senders or admit legitimate senders. An alternative way of filtering is check the content by matching the keyword associated with the packet [69, 70] or using machine learning techniques [71] to detect spam. Social graph and relevant characteristics are also exploited to filter spam [72, 73]. Most of these schemes require the centralized server or trusted authority to perform spam filtering based on historical information. When spam senders shift to autonomous MSNs, they have more opportunities of going undetected [54] since autonomous MSNs do not rely on centralized and trusted servers and lack historical information. To tackle this problem, we propose a distributed filtering scheme to enable MSN users (filter creators) to personalize their spam filters. These filters are sent to some filter holders and allow these filter holders

to block spam when they are requested to forward packets to filter creators. However, several challenges may hinder the flourish of this type of filtering schemes. First of all, it is difficult to determine the filter holders who take filters for filter creators. If filter creators greedily distribute their filters to all the other users in MSNs, massive network bandwidth and resources would be consumed although it can benefit individual users. Therefore, how to distribute filters with the consideration of both distribution costs and filtering accuracy becomes a problem. Moreover, security and privacy concerns are raised when the distributed filters contain some privacy-sensitive information regarding filter creators. If this private information embedded in filters is directly disclosed to untrusted entities, filter creator's privacy, e.g., lifestyles, health condition and preferences, may be disclosed and inferred [74, 75]. In addition, MSNs are vulnerable to malicious attacks, which may illegally forge filters to block or delay the useful information transmitted in MSNs but bypass spam. The aforementioned challenging issues motivate us to efficiently filter spam in MSNs and preserve users' privacy at the same time.

In this chapter, we propose a Personalized fIne-grained spam Filtering scheme (PIF) with privacy preservation in autonomous MSNs. The PIF exploits personalized filters with social assisted filter distribution, privacy-preserving coarse-grained and fine-grained filterings, and efficient filter update. Specifically, the main contributions of this chapter are three-fold.

• Firstly, we develop a personalized filtering scheme. It allows the filter creator to personalize his filters in both coarse-grained and fine-grained ways. The keyword embedded in the coarse-grained filter enables filter holders to forward the packets containing the same keyword to the filter creator. The PIF also provides a fine-grained filtering scheme based on a variant of hidden vector encryption. Both schemes prevent keywords in the filters from directly disclosing to others, including filter holders.

• Secondly, we investigate the mobility and social relationship of MSN users. We also exploit the opportunistic contacts among users to analyze the packet delivery process in MSNs. According to this analysis, we propose a social assisted filter distribution scheme, which enables the filter creator to send filters to his social friends who have high probability to be the relay forwarding packets to him. As such, the PIF can reduce the filter distribution overhead and maintain the filtering accuracy.

• Thirdly, we conduct extensive simulations to show that the PIF can significantly reduce the storage and communication costs and deliver the useful packets in a low delay. Meanwhile, the security property analysis demonstrates that the PIF protect user's private keyword from directly disclosing to inside curious attackers and detect forged filters.

The remainder of this chapter is organized as follows. In Section 3.2, we review the

related works on spam filtering. We introduce the network and threat models with design goals in Section 3.3. Then, we propose the details of the PIF in Section 3.4, followed by the security discussions and the simulations in Sections 3.5 and 3.6, respectively. Finally, we conclude the chapter in Section 3.7.

3.2 Related Works

Extensive research efforts have been put on spam filtering [76, 77, 78]. Intuitively, some sophisticated filtering schemes exploit whitelist, blacklist [68] and graph [79] to bypass legitimate senders and block spam senders. In terms of blacklist based spam filtering, Soldo et al. [68] propose a predictive blacklisting scheme to forecast spam senders based on historical sender logs. With a multi-level prediction algorithm, an implicit recommendation system is formulated to resist spam. Using keyword to filter spam, Lu et al. [69] propose a relay-based keyword filtering scheme (PReFilter) in DTNs to detect the unwanted packets via keyword matching. The PReFilter enables relays to hold filters generated by other users such that it detects and block spam before it is transmitted to the receivers. Meanwhile, the filters with privacy-sensitive keywords are encrypted to protect user's privacy leakage. But the PReFilter does not consider the overhead of filter distribution and update.

In sociology theory, social network represents the social graph built by users in the network [80], which can be helpful to detect and filter spam. Lahmadi et al. [81] utilize social network to collaboratively filter the short message services based spam via the Bloom filters and content hashing filters. This collaborative filtering scheme also relies on a centralized server to build the social network among users. Hameed et al. [82] study the e-mail recipient's social network and mitigate spam outside of the social circle, which can also reduce the Internet bandwidth consumption by spam. To resist spam, malware and phishing via URLs, Thomas et al. [83] develop a real-time system, including URL aggregation, feature collection, feature extraction and classification. The proposed system visits every URL and collects its features, which are stored a centralized server for extraction in the training phase and real-time decision-making. Meanwhile, some social features, such as social interests, closeness, personal preferences and trust, are also adopted to facilitate the spam filtering. Li et al. [72] develop a social network based spam filtering framework. It can detect junk emails with the consideration of social features of users and network [84] such that the regular and junk emails can be differentiated. In [73], social trust is exploited to collaboratively filter spam. The spam reporter's trustworthiness is used to collect the correct spam reports and detect Sybil attacks at the same time. Li et al. [85] also exploit collaborative and privacy-preserving anti-spam system to resist a wide range

of camouflage attacks. The proposed ALPACAS framework controls the amount of shared information among the collaborated entities to achieve the confidentiality of e-mails.

In addition, Fan et al. [86] investigate the least cost rumor blocking problem to limit the negative rumor diffusion in social network. The community feature is utilized to minimize the total number of so-called rumor protectors and protect bridge ends, as known as the boundary individuals within the neighbor communities of rumor source. Based on a susceptible-infectious model, Shah et al. [87] propose a systematic framework to estimate and detect the rumor source. It is formulated as a maximum likelihood estimator for a class of graphs. Similarly, Wang et al. [88] detect the source of rumors with multiple observations based on the susceptible-infectious model. The multiple observations in a tree network are exploited to improve the rumor source detection. Different from most of existing filtering schemes, Stringhini et al. [89] propose a new approach to detect spam by looking at the way how emails are sent instead of content and origin of emails. For example, it can detect the IP address from which the message is sent, and the geographical distance between the sender and the receiver. They investigate the SMTP communication between the email sender and receiving mail server. The introduced concept of SMTP dialects capture small variations in the ways to carry out the SMTP protocol such that they can distinguish the between normal email senders and spam bots.

However, there are still many challenging issues for spam filtering in MSNs. Firstly, most of social network based filtering schemes are based on centralized trusted authority to perform the detection, which leaves a gap of filtering schemes between OSNs and MSNs. Secondly, the decentralized schemes, e.g. PreFilter [69] and SAFE [70], are limited due to the lack of knowledge about the packet recipients (i.e., filter creator). The SAFE offers spam filtering based on keyword matching, which is a coarse-grained approach. Filter creators only select the keyword from the keyword space of the network. The coarse-grained keyword filter may not reflect the sufficient features of the delivered packets. To this end, we propose a personalized fine-grained spam filtering scheme to allow the filter creators to generate filters with different features in multiple dimensions. The proposed PIF scheme can allow creators to personalize his filters. Both coarse-grained and fine-grained filtering schemes are integrated in the PIF.

3.3 System Model and Design Goals

In this section, we present the network model and design goals including efficiency of spam filtering and privacy preservation.

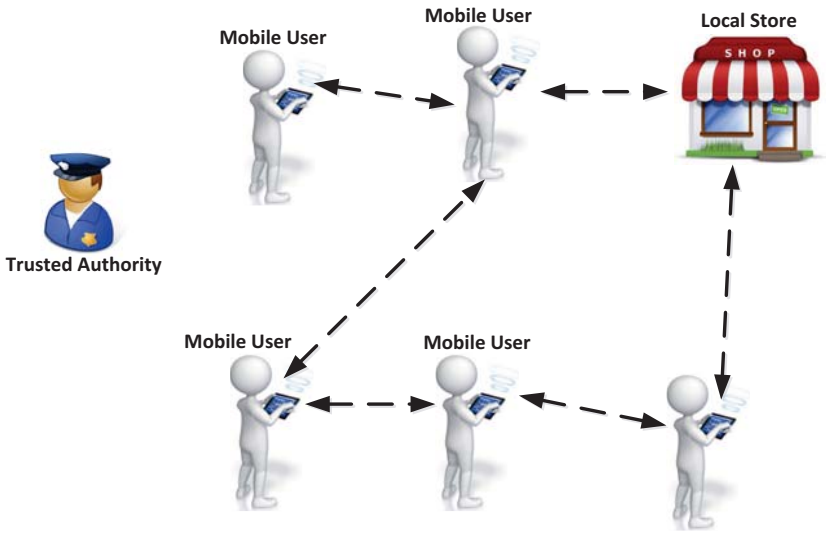


Figure 3.2: Network model

3.3.1 Network Model

We consider an MSN including a trust authority (TA) and N users (including mobile users and local stores) as shown in Fig. 3.2.

• *Trust Authority (TA)* is trusted by users, and bootstraps the whole system during the initialization phase. TA can generate secret master keys and receive the registration requests from legitimate users. Then, TA also issues certificates to legitimate users during registration. TA does not participate in user's communication and filtering.

• *Users* include mobile users and local stores having smartphones or wearable devices to communicate with each other in the local area. They are denoted by $\mathbb{U} = \{u_1, u_2, ..., u_N\}$. The power and storage occupancy of each user's smartphone are limited. Each legitimate user first registers to the TA to build user's profiles and obtain key materials, e.g., unique identity, certificate and secret keys which should be securely kept for session key generation. In packet delivery and spam filtering phases, users can authenticate their identities and filters, and verify other user's information.

3.3.2 Threat Model

Several threats may occur in MSNs to violate user's privacy during the phases of packet delivery and spam filtering. We consider two types of threats: inside curious user and forgery attack.

First, some of the filter holders may be curious about other user's preferences and personal profiles included in the distributed filters. These inside curious user may violate filter holder's privacy-sensitive information during filter distribution, storage, packet delivery and filtering phases. In addition, these filter holders can honestly follow the protocols.

Secondly, some malicious users may forge other user's filters to benefit themselves or degrade the performance of MSNs. Either the useful packets may be blocked, or spam may be normally delivered in MSNs. A large number of communication and storage overheads would be consumed if these malicious users exist in MSNs.

3.3.3 Design Goals

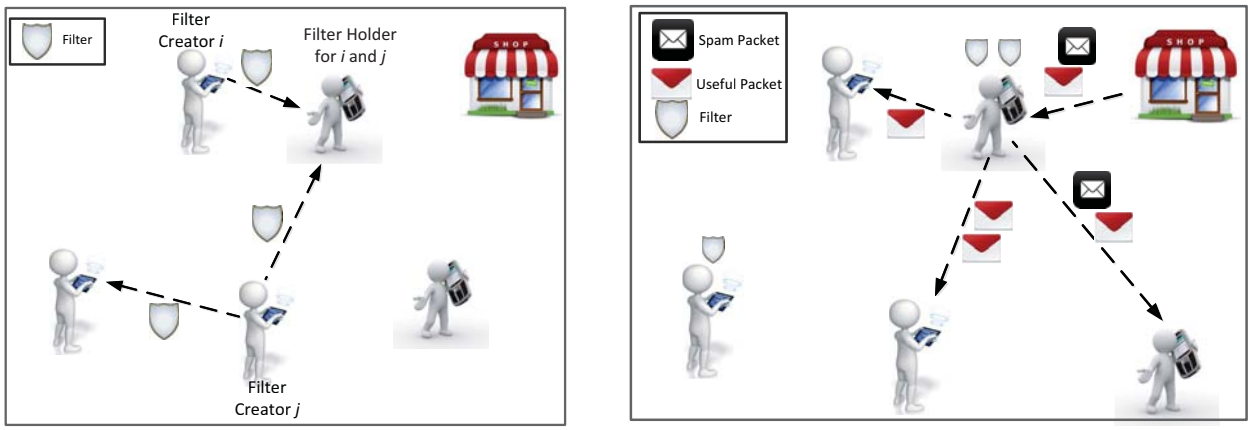
In this chapter, our design goal is to develop a personalized fine-grained filtering scheme with user's privacy preservation.

Efficiency goals

Due to the opportunistic contact (i.e., intermitted end-to-end connectivity) and limited smartphone battery, our goal is to develop an efficient spam filtering scheme to detect and block the spam in MSNs as early as possible. The proposed scheme should efficiently filter the spam and cost few extra storage, communication and computational overheads. Meanwhile, it should be able to bypass useful packets without any delayed delivery of them. In addition, the distributed filters should be personalized by filter creators and updated timely.

Security goals

Our security goal is to preserve user's privacy against inside curious users and detect the forged filters. First, the proposed spam filtering scheme should be able to preserve the filter creator's privacy from directly disclosing. The keyword included in the distributed filters cannot appear in plaintext to others. During the filtering, the keyword should be



(a) Filter distribution

(b) Filtering

Figure 3.3: PIF scheme (In filter distribution phase, filter creator sends his filters to his social friends. In filtering phase, filter holders block spam to the filter creator with his filters.)

also invisible to others and kept in the ciphertext. Secondly, the proposed scheme should be able to prevent malicious users from forging legitimate user's filters. If any filter is forged, the filter creator and other users are able to detect it efficiently.

3.4 Proposed PIF Scheme

In this section, we propose the PIF scheme as shown in Fig. 3.3. Firstly, users (i.e., filter creators) build their personalized filters embedding the keywords and degree. Then, the filter creator sends his filters to his social friends (i.e., filter holders). When meeting a sender who wants to send a packet to the filter creator, filter holders use these filters to check if this packet if desired by the filter creator, and block spam in the early stage of the packet delivery. The PIF consists of social assisted filter distribution, coarse-grained and fine-grained filters, and Merkle Hash tree based filter authentication and update.

Table 3.1: Frequently Used Notations

Notation	Description
$\lambda_{i,j}$	Contact rate between u_i and u_j
$C_{i,j}$	Contact between u_i and u_j within period T
$\overline{C_{i,j}}$	Expectation of Contact Times between u_i and u_j within period T
$P_i(t \leqslant T)$	Probability that u_i meets another user in T
$P^r_{s,d}(t \leqslant T)$	Forwarding probability that u_i forwards a packet to u_d through a relay u_r within T
$W_{i,x}$	u_i's x-th keyword
$\mathcal{F}_i$	u_i's keyword filter set
TH	Number of common communities

3.4.1 Social Based Filtering Distribution

To efficiently distribute filters, we first formulate the packet delivery process to understand the effective way (or relay selection) of packet forwarding in MSNs. Some frequent used notations are listed in Table 3.1.

The packet delivery in MSNs relies on users' opportunistic contacts. According to [90, 91], the contact between two users u_i and u_j follows a Poisson distribution with the pairwise contact rate $\lambda_{i,j}$. A binary random variable $C_{i,j}$ is defined as

$$C_{i,j} = \begin{cases} 1, & \text{if } u_i \text{ and } u_j \text{ meet within time period } T; \\ 0, & \text{otherwise.} \end{cases}$$

Let λ_i be the average contact rate that u_i meets any other user. We have

$$\overline{C_{i,j}} = 1 \cdot \int_0^T \lambda_i e^{-\lambda_i t} dt + 0 \cdot \int_T^\infty \lambda_i e^{-\lambda_i t} dt. \tag{3.1}$$

Therefore, $C_{i,j}$ follows Bernoulli distribution. As the contacts between each two users are independent [91], the probability that u_i meets another user in T is

$$\begin{aligned} P_i(t \leqslant T) &= 1 - \prod_{\substack{u_j \in \mathbb{U} \\ j \neq i}} (1 - \overline{C_{i,j}}) \\ &= 1 - e^{-\sum\limits_{\substack{u_j \in \mathbb{U} \\ j \neq i}} \lambda_{i,j} T}. \end{aligned} \tag{3.2}$$

Let $\lambda_i = \sum_{\substack{u_j \in \mathbb{U} \\ j \neq i}} \lambda_{i,j}$, then $P_i(t \leqslant T) = 1 - e^{-\lambda_i T}$. Thus, t follows power-law distribution. The PDF (probability distribution function) is $f_i(t) = \lambda_i e^{-\lambda_i t}$ $(t \geqslant 0)$. We have the average contact interval of u_i as

$$\overline{E_i(t)} = \int_0^\infty t f_i(t) dt = \int_0^\infty t \lambda_i e^{-\lambda_i t} dt = \frac{1}{\lambda_i} \tag{3.3}$$

According to [20, 92, 93], users in the same social community may have a higher probability to meet each other since social community indicates users' personal interests. Consider the packet delivery within one community (u_s, u_r and u_d are in the same community), if the sender u_s meets a relay u_r at t_1 and u_r meets the destination u_d at t_2, the forwarding probability $P^r_{s,d}(t = t_1 + t_2 \leqslant T)$ is

$$\begin{aligned} P^r_{s,d}(t \leqslant T) &= \int_0^{t_1} \lambda_{s,r} e^{-\lambda_{s,r} t} dt \cdot \int_{t_1}^{T} \lambda_{r,d} e^{-\lambda_{r,d} t} dt \\ &= \int_0^T f_{s,r}(t) \otimes f_{r,d}(t) dt \\ &= \int_{t=0}^{T} \left(\int_{\tau=0}^{t} f_{s,r}(\tau) \cdot f_{r,d}(t-\tau) d\tau \right) dt. \end{aligned} \tag{3.4}$$

Note that $\otimes$ is the convolution. Because u_r knows $t_{s,r}$,

$$P^r_{s,d}(t = t_1 + t_2 \leqslant T) \geqslant P_r(t_1 \leqslant t_{s,r}) \cdot P_r(t_2 \leqslant t_{s,r}). \tag{3.5}$$

Thus, we have

$$\begin{aligned} P^r_{s,d}(t \leqslant T) &= \int_{t=0}^{T} \left(\int_{\tau=0}^{t} f_{s,r}(\tau) \cdot f_{r,d}(t-\tau) d\tau \right) dt \\ &\geqslant \int_{\tau_1=0}^{t_{s,r}} f_{s,r}(\tau_1) d\tau_1 \cdot \int_{\tau_2=0}^{T-t_{s,r}} f_{r,d}(\tau_2) d\tau_2 \\ &= \left(1 - e^{-\lambda_{s,r} t_{s,r}}\right) \cdot \left(1 - e^{-\lambda_{r,d}(T - t_{s,r})}\right). \end{aligned} \tag{3.6}$$

With the consideration of both direct and indirect contacts between u_s and u_d [94], the probability of forwarding a packet from u_s to u_d is

$$p_{s,d}(t \leqslant T) = 1 - (1 - P_{s,d}(t \leqslant T)) \prod_{\substack{u_r \in \mathbb{U} \\ r \neq s,d}} \left(1 - P^r_{s,d}(t \leqslant T)\right). \tag{3.7}$$

Then, we have

$$p_{s,d}(t \leqslant T) \geqslant 1 - e^{-\lambda_{s,d}T} \cdot \prod_{\substack{u_r \in \mathbb{U} \\ r \neq s,d}} \left(1 - p_{s,d}^{r}\right) \tag{3.8}$$

where $p_{s,d}^{r} = \left(1 - e^{-\lambda_{s,r}t_{s,r}}\right) \cdot \left(1 - e^{-\lambda_{r,d}(T - t_{s,r})}\right)$. Since $0 \leqslant 1 - p_{s,d}^{r} \leqslant 1$ where $u_r \in \mathbb{U}$ and $r \neq s, d$, $p_{s,d}$ becomes smaller when multiplied by more items such as $1 - p_{s,d}^{r}$.

If multiple relay users are selected for the packet forwarding, the probability of multi-hop packet delivery in time period T can be

$$P_{s,d}^{r \cdots r'}(t \leqslant T) = \int_{0}^{T} f_{s,r}(t) \otimes \cdots \otimes f_{r',d}(t) dt. \tag{3.9}$$

With multiple communities, the probability that u_s forwards the packet to u_d can be calculated as

$$\begin{aligned} \mathcal{P}_{s,d}(t \leqslant T) &= 1 - \prod_{i \in \mathbb{CC}_{s,d}} \left(1 - p_{s,d}(t \leqslant T, i)\right) \\ &\geqslant \max_{i \in \mathbb{CC}_{s,d}} \{p_{s,d}(t \leqslant T, i)\}. \end{aligned} \tag{3.10}$$

It is larger than the probability that u_s forwards the packets within only one community. Therefore, the PIF selects the filter holders as the users who have large number of common communities with the filter creator.

3.4.2 Coarse-grained Filtering

To achieve the security goals, the coarse-grained filtering for PIF consists of initialization, filter generation, filter distribution, and filtering as follows.

• ***Initialization***: TA bootstraps the system and assigns secret keys to individual users. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two additive cyclic groups. They have the same order q, and $\mathbb{G}$'s generator is P. Note that q a large prime. A bilinear pairing [95] exists between $\mathbb{G}$ and $\mathbb{G}_T$ is e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. We have $e(xP, yP) = e(P, P)^{xy}$, where x and y are randomly selected from $\mathbb{Z}_q^*$. A key generation algorithm $\mathcal{G}$ takes as input a security parameter $\mathbf{k}$, and outputs (q, $\mathbb{G}$, $\mathbb{G}_T$, P, e, H_1), where H_1 is a trapdoor hash function $\mathsf{H}_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$. Then, u_i randomly picks $x_i \in \mathbb{Z}_q^*$ to compute his public key $\mathsf{PK}_i = \frac{1}{x_i}P$ and secret key $\mathsf{SK}_i = x_i$.

• ***Filter Generation***: The filter creator u_i runs an algorithm $\mathsf{FilGen}(W_{i,x}, \mathsf{SK}_i) \longrightarrow \mathcal{F}_{i,x}$ to generate filters, where $W_{i,x}$ is the keyword, and $\mathcal{F}_{i,x}$ is the generated filter for $W_{i,x}$.

u_i selects his keywords $W_{i,1}, \ldots, W_{i,K}$ where $1 \leqslant k \leqslant K$, and establishes a keyword list $\mathcal{W}_i$. Note that $K \subseteq \mathbf{K}$ which is the keyword space of the whole MSN. Every keyword in $\mathbf{K}$ is semantically defined by the TA. Each user selects his keyword according to his social interests. For a specific keyword $W_{i,k}$ (e.g., "Health"), the filter $\mathcal{F}_{i,k} = \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})} P$. The keyword filter set for u_i is $\mathcal{F}_i = (\mathcal{F}_{i,1}, \cdots, \mathcal{F}_{i,k})$.

• ***Filter Distribution***: If u_i meets another user u_j, they first authenticate each other and privately compare with their profiles to determine the number of their common communities (as discussed in Section 3.4.1). We adopt privacy-preserving profile matching scheme in [96] to enable users to learn their common communities. If the number of their common communities is larger than a threshold TH, u_i can send his filter $\mathcal{F}_i$ to u_j as the filter holder.

• ***Filtering***: A packet sender u_s wants to deliver a packet including keywords $(W_{s,1}, \cdots, W_{s,x})$ to u_i. When u_s meets u_j, u_j helps u_i to determine whether the packet from u_s can be delivered or not.

First, u_s runs an algorithm $\mathsf{Packet}(\mathsf{PK}_i, W_{s,x}) \longrightarrow \varphi_s$ to generate keyword for a packet to u_i. For the keyword $W_{s,x}$, u_s computes $\varphi_1 = r(\frac{1}{\mathsf{H}_1(W_{s,x})} P + \mathsf{PK}_i)$. Then, u_s randomly selects $r \in \mathbb{Z}_q^*$ and computes $\varphi_s =< \varphi_0, \varphi_1 >$, where $\varphi_0 = e(P, r\mathsf{PK}_i)$. u_s sends φ_s to u_j.

Then, u_j runs an algorithm $\mathsf{Filter}(\mathcal{F}_i, \varphi_s) \longrightarrow \{0, 1\}$ to perform filtering. For every received $\mathcal{F}_{i,k} \in \mathcal{F}_i$ from u_i, u_j checks whether $\varphi_0 = e(\varphi_1, \mathcal{F}_{i,k})$ or not. If it holds, $\mathsf{Filter}(\mathcal{F}_i, \varphi_s)$ outputs 1, indicating that the keyword $W_{s,x}$ matches u_i's filter. This packet should be forwarded. Otherwise, $\mathsf{Filter}(\mathcal{F}_i, \varphi_s)$ outputs 0 and the packet is discarded by u_j. When there are multiple keywords in the packet from u_s to u_i, u_j discards u_s's packet if none of the keywords associated with u_s matches u_i's filter.

The above four steps enable the filter holders to check the packet's keyword matching in a coarse-grained manner (i.e., coarse-grained filtering). The details of the coarse-grained filtering scheme are illustrated in Algorithm 1.

The correctness of the coarse-grained filtering scheme is guaranteed. We have $e(P, r\mathsf{PK}_i) = e(P, \frac{r}{x_i} P) = e(P, P)^{\frac{r}{x_i}}$, and $e(\varphi_1, \mathcal{F}_{i,k}) = e(r(\frac{1}{\mathsf{H}_1(W_{s,x})} P + \mathsf{PK}_i), \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})} P)$ as follows.

Algorithm 1: Social-assisted Coarse-grained Filtering

Procedure: Social-assisted Filtering
u_s wants to send a packet including keyword $W_{s,x}$ via u_j to u_i
if u_j has u_i's filters **then**
    u_j checks if the keyword in the packet is valid or not
    u_s sends $\varphi_s =< \varphi_0, \varphi_1 >$ to u_j
    for All $\mathcal{F}_{i,k} \in \mathcal{F}_i$ **do**
        u_j computes $e(\varphi_1, \mathcal{F}_{i,k})$
        if $e(\varphi_1, \mathcal{F}_{i,k}) = \varphi_0$ **then**
            u_s forwards the packet to u_j;
            Abort.
        end if
    end for
    u_j discards u_s's the packet, and informs u_s
else
    u_s forwards this packet to u_j
end if
end procedure

$$
\begin{aligned}
e(\varphi_1, \mathcal{F}_{i,k}) &= e(r(\frac{1}{\mathsf{H}_1(W_{s,x})}P + \mathsf{PK}_i), \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P) \\
&= e(r(\frac{1}{\mathsf{H}_1(W_{s,x})}P + \frac{1}{x_i}P), \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P) \\
&= e(\frac{r(x_i + \mathsf{H}_1(W_{s,x}))}{x_i \mathsf{H}_1(W_{s,x})}P, \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})}P) \\
&= \begin{cases} e(P,P)^{\frac{r}{x_i}}, & \text{If } W_{i,k} = W_{s,x}; \\ random, & \text{otherwise.} \end{cases}
\end{aligned}
$$

When two keywords match, $\varphi_0 = e(\varphi_1, \mathcal{F}_{i,k})$. If the keywords are not the same, $e(\varphi_1, \mathcal{F}_{i,k})$ is random.

3.4.3 Fine-grained Filtering

Although the coarse-grained keyword-based filter can block a portion of packets when matching keywords, users may want to personalize their filters due to their own preferences. It is necessary to provide a fine-grained filtering solution. The filter creator can define

various levels of his interests corresponding to the specific keyword, and allow the filter holders to fine-grained filter the packets. To this end, we develop a variant of hidden vector encryption technique [97, 98] in the PIF scheme to achieve the fine-grained [99] spam filtering.

The filter creator u_i generates his fine-grained keyword filter as a vector $\mathbf{w} = (w_1, \cdots, w_l) \in \{1, \cdots, n\}^l$ to indicate his interest degree in certain keyword. A high w_l means that u_i is likely interested in the l-th keyword. Denote $\sigma^*(\mathbf{w}) = \sigma^*_{a,b} \in \{1, *\}^{nl}$ as

$$\sigma^*_{a,b} = \begin{cases} 1, & \text{if } w_a = b \\ *, & \text{otherwise.} \end{cases}$$

Let $f(\sigma^*(\mathbf{w}))$ be the set of all index k such that $\sigma^*_k \neq *$ where $k \in \{1, \cdots, nl\}$.

When the sender u_s wants to send a packet with keyword W', u_s builds the encryption vector $\sigma(\mathbf{w}') = \sigma_{a,b} \in \{0, 1\}^{nl}$ for $\mathbf{w}' = (w'_1, \cdots, w'_l) \in \{1, \cdots, n\}^l$ as

$$\sigma_{a,b} = \begin{cases} 1, & \text{if } w'_a \geqslant b, \\ 0, & \text{otherwise.} \end{cases}$$

Here, $a \in \{1, \cdots, l\}$ and $b \in \{1, \cdots, n\}$. For example, let $l = 3, n = 4$, and $\mathbf{w} = (1, 3, 1)$. The vector $\sigma^*(\mathbf{w}) = (1 * **, * * 1*, 1 * **)$, indicating that the matching condition with another vector $\mathbf{w}'$ is $P = (w'_1 \geqslant 1) \bigwedge (w'_2 \geqslant 3) \bigwedge (w'_3 \geqslant 1)$. When the encryption vector $\mathbf{w}' = (2, 3, 1)$. $\sigma(\mathbf{w}') = (1100, 1110, 1000)$. Therefore, the two vectors are matched.

Define a predicate function

$$P(\sigma^*(\mathbf{w}), \sigma(\mathbf{w}')) = \begin{cases} 1, & \text{if for all } i \in f(\sigma^*(v)),\ \sigma^*(w_a) = \sigma(w'_a) \\ 0, & \text{otherwise.} \end{cases}$$

If $P(\sigma^*(\mathbf{w}), \sigma(\mathbf{w}')) = 1$, u_j can forward the packet to u_s. We consider "$\geqslant$" predicate in this chapter. The proposed scheme can be extended to "$\leqslant$" and some other predicates. The combination of multiple predicates is also feasible. Having the predicate, we propose a fine-grained filtering scheme to not only enable the filtering but also preserve the privacy of sender's keyword vector.

• ***Initialization***: Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two multiplicative cyclic groups. Both $\mathbb{G}_1$ and $\mathbb{G}_2$ have the same order q, where q is a large prime. $\mathbb{G}_1$'s generator is g. Let e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing, if it satisfies that $e(g^a, g^b) = e(g, g)^{ab}$ for any random numbers $a, b \in \mathbb{Z}^*_q$. A bilinear key generation algorithm $\mathcal{G}$ takes as input the security parameter $\mathbf{k}$, and outputs (q, $\mathbb{G}_1$, $\mathbb{G}_2$, g, e).

TA randomly selects elements g_1, g_2, (h_1, u_1, ψ_1), $\cdots$, $(h_{nl}, u_{nl}, \psi_{nl}) \in \mathbb{G}_1$, and picks random numbers $y_1, y_2, v_1, \cdots, v_{nl}, t_1, \cdots, t_{nl} \in \mathbb{Z}^*_q$. TA computes $Y_1 = g^{y_1}$, $Y_2 = g^{y_2}$,

$\Gamma = e(g_1, Y_1)e(g_2, Y_2) \in \mathbb{G}_2$, $V_k = g^{v_k} \in \mathbb{G}_1$ and $T_k = g^{t_k} \in \mathbb{G}_1$ for $t, k \in (1, \cdots, nl)$. The public key PK and secret key SK are

$$PK = (g, Y_1, Y_2, (h_1, u_1, \psi_1, V_1, T_1), \cdots, (h_{nl}, u_{nl}, \psi_{nl}, V_{nl}, T_{nl})) \tag{3.11}$$

$$SK = (g_1, g_2, y_1, y_2, v_1 \cdots, v_{nl}, t_1 \cdots, t_{nl}).$$

• ***Filter Generation***:

u_i builds his fine-grained filter $\mathbf{w} = (w_1, \cdots, w_l) \in \{1, \cdots, n\}^l$, where $\mathbf{w}$ is determined by u_i. Denote u_i's interest degree as $D_{i,x} \in [1, \rho]$, and ρ indicates the highest interest degree. u_i randomly selects each $w_x \in N(D_{i,x}, \sigma)$, where $N(D_{i,x}, \sigma)$ is a Gaussian distribution and σ is determined by u_i. By selecting a proper σ, the real interest degree $D_{i,x}$ can be protected from directly disclosing to filter holders. Then, u_i maps it to vector $\sigma^*(\mathbf{w})$. Then, $\sigma^*(\mathbf{w})$ is sent to u_j with the encryption of AES, when they are encountered.

u_j decrypts $\sigma^*(\mathbf{w})$ from u_i and secretly keeps it. Then, u_j selects two random numbers $\alpha, \beta \in \mathbb{Z}_q^*$, and picks random tuples $< \mu_a, \phi_a, \theta_a, \delta_a > \in \mathbb{Z}_q^*$ such that $\mu_a y_1 + \phi_a y_2 = \alpha$ and $\theta_a y_1 + \delta_a y_2 = \beta$ for all $a \in f(\sigma^*(\mathbf{w}))$.

Then, u_j computes the filter $\mathsf{F}(\sigma^*(\mathbf{w}))$ as

$$\begin{aligned}
F_1 &= g_1 \prod_{a \in f(\sigma^*(\mathbf{w}))} (h_i u_i^{\sigma^*(w_a)})^{\mu_a} \psi^{\theta_a}, \\
F_2 &= g_2 \prod_{a \in f(\sigma^*(\mathbf{w}))} (h_i u_i^{\sigma^*(w_a)})^{\phi_a} \psi^{\delta_a}, \\
F_3 &= g^{\alpha}, F_4 = g^{\beta}, F_5 = g^{-\sum\limits_{a \in f(\sigma^*(\mathbf{w}))} (v_i \alpha + t_i \beta)}.
\end{aligned}$$

• ***Filtering***:

u_s first generates ciphertext with the keyword related vector $\sigma(\mathbf{w}'_s)$. Then, u_s encrypts $\sigma(\mathbf{w}'_s)$ by using u_j's public key PK. u_s also picks two random numbers ρ_1 and $\rho_2 \in \mathbb{Z}_q^*$, and

sends the ciphertext CT=$(C_1, C_2, C_{3,1}, \cdots, C_{3,nl}, C_{4,1}, C_{4,nl}, C_5, C_6)$ where

$$\begin{aligned}
&C_1 = Y_1^{\rho_1}, C_2 = Y_2^{\rho_1} \\
&C_{3,1} = (h_1 u_1^{w'_a})^{\rho_1} V_1^{\rho_2} \\
&\cdots, \\
&C_{3,nl} = (h_{nl} u_{nl}^{w'_a})^{\rho_1} V_{nl}^{\rho_2} \\
&C_{4,1} = \psi_1^{\rho_1} T_1^{\rho_2} \\
&\cdots, \\
&C_{4,nl} = \psi_{nl}^{\rho_1} T_{nl}^{\rho_2} \\
&C_5 = g^{\rho_2}, C_6 = \Gamma^{\rho_1}.
\end{aligned}$$

Having CT from u_s, u_j aggregates $C'_3 = \prod_{a \in f(\sigma^*(\mathbf{w}))} C_{3,a}$ and $C'_4 = \prod_{a \in f(\sigma^*(\mathbf{w}))} C_{4,a}$. u_j collects the indexes of keyword passed the coarse-grained keyword filtering, and checks

$$\frac{e(F_1, C_1) e(F_2, C_2)}{C_6} \stackrel{?}{=} e(F_3, C'_3) e(F_4, C'_4) e(F_5, C_5) \tag{3.12}$$

If Equation 3.12 holds, u_j forwards the packet to u_i; otherwise, u_j discards it.

The correctness of fine-grained filtering is as follows.

$$\begin{aligned}
& e(K_1, C_1) e(K_2, C_2) \\
= \; & e\left(g_1 \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\mu_a} \psi^{\theta_a}, g^{y_1 \rho_1}\right) e\left(g_2 \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\phi_a} \psi^{\delta_a}, g^{y_2 \rho_1}\right) \\
= \; & \Gamma^{\rho_1} \prod_{a \in f(\sigma*(\mathbf{w}))} \left[e((h_a u_a^{w_a})^{\mu_a}, g^{y_1 \rho_1}) e((h_a u_a^{w_a})^{\phi_a}, g^{y_2 \rho_1})\right] \prod_{a \in f(\sigma*(\mathbf{w}))} \left[e((\psi^{\theta_a}, g^{y_1 \rho_1}) e(\psi^{\delta_a}, g^{y_2 \rho_1})\right] \\
= \; & \Gamma^{\rho_1} \prod_{a \in f(\sigma*(\mathbf{w}))} e((h_a u_a^{w_a})^{\rho_1}, g^{\mu_a y_1 + \phi_a y_2}) \prod_{a \in f(\sigma*(\mathbf{w}))} e(\psi^{\rho_1}, g^{\theta_a y_1 + \delta_a y_2}) \\
= \; & \Gamma^{\rho_1} e\left(\prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a})^{\rho_1}, g^{\alpha}\right) e\left(\prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}, g^{\beta}\right)
\end{aligned}$$

$$
\begin{aligned}
& e(K_3, C_3')e(K_4, C_4')e(K_5, C_5) \\
= \; & e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a'})^{\rho_1} g^{v_a \rho_2}\right) e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1} g^{t_a \rho_2}\right) e\left(g^{-\sum_{a \in f(\sigma*(\mathbf{w}))} (v_a \alpha + t_a \beta)}, g^{\rho_2}\right) \\
= \; & e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a'})^{\rho_1}\right) e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}\right) \\
& e(g^{\rho_2}, \prod_{a \in f(\sigma*(\mathbf{w}))} g^{v_a \alpha + t_a \beta}) e\left(g^{-\sum_{a \in f(\sigma*(\mathbf{w}))} (v_a \alpha + t_a \beta)}, g^{\rho_2}\right) \\
= \; & \begin{cases} e\left(g^{\alpha}, \prod_{a \in f(\sigma*(\mathbf{w}))} (h_a u_a^{w_a'})^{\rho_1}\right) e\left(g^{\beta}, \prod_{a \in f(\sigma*(\mathbf{w}))} \psi^{\rho_1}\right), \text{if } w_i = w_i' \text{ for all } a \in f(\sigma * (\mathbf{w})); \\ \\ \perp, \text{Otherwise.} \end{cases}
\end{aligned}
$$

Note that $C_6 = \Gamma^{\rho_1}$. If $\mathbf{w}$ matches $\mathbf{w}'$, it passes the fine-grained filtering such that the packet from u_s is forwarded to u_i.

3.4.4 Filter Authentication and Update Scheme

After filtering, the filter holders should authenticate filters to the packet senders to verify that the blocked or forwarded packet is determined by the filter creator. We exploit Merkle Hash tree [100] (i.e., a tree structure of cryptographic Hash functions) to authenticate each filter. We propose the construction of Hash tree for filters with the filter authentication.

Merkle Hash tree has a typical binary tree structure including 2^{N-1} leaf nodes. The depth of Merkle tree is N [101]. A parent node $p_{i-j} = \mathsf{H}(ch_i||ch_j)$ is computed by a one-way Hash function with the input as the children nodes. In Fig. 3.4, given the leaf nodes ch_1 and ch_2, the parent node $p_{1-2} = \mathsf{H}(ch_1||ch_2)$ as shown. Similarly, p_{1-4} is computed by concatenating p_{1-2} and p_{3-4}. The root node $r_{1-8} = \mathsf{H}(p_{1-4}||p_{5-8})$. Let $\mathcal{PH}_1 = \{ch_2, p_{3-4}, p_{5-8}\}$ be the path from the leaf node ch_1 to the root r_{1-8}. $\mathcal{PH}_1$ can be used to authenticate the leaf node ch_1.

In the PIF, the filter creator u_i builds his keyword list $\mathcal{W}_i = \{W_{i,1}, \ldots, W_{i,K}\}$, $1 \leqslant k \leqslant K$. Each keyword is located in the leaf of Merkle Hash tree $\mathcal{FR}_{u_i}$. In the authentication, the path $\mathcal{PH}_k$ of $W_{i,k}$ is the certificate of the keyword $W_{i,k}$. The verifier checks if the concatenated hash value of $\mathcal{PH}_k$ equals the root R_i or not. If not, the keyword is forged.

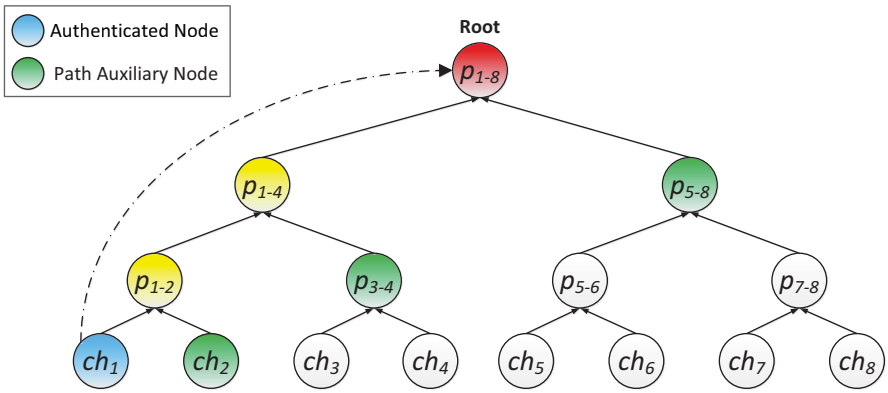


Figure 3.4: Merkle Hash tree based filter authentication

In addition, the filter creator generates a signature on $\mathsf{H}(R_{u_i} \| ID_{u_i})$. The root value and the path connecting each leaf node to the root are verifiable. Suppose there are 2^N leaf nodes in a Merkle Hash tree. Users perform N Hash operations to verify each keyword (leaf node). The size of filter's signature is $N \times L$. Note that L denotes the length of each Hash value. For example, in SHA-256, L is 256 bits.

The properties of Merkle Hash tree is exploited to check the filter's version. We propose a filter update scheme based on this property. As we presented above, the root of Merkle Hash tree changes if any leaf node varies. We do not need to check every leaf node (i.e., keyword) of the distributed filter. The filter creator u_i checks the root value R_{u_i} from his filter holder u_j for filter tree $\mathcal{FR}_{u_i}$. If the root is an existing root value, u_i sends the updated filter tree $\mathcal{FR}'_{u_i}$ to u_j as illustrated in Algorithm 2. The PIF improves the efficiency of filter search during the filter update. The Merkle Hash tree can be also extended to fine-grained filter where each value in the vector is assigned as leaf node.

3.5 Security Discussions

In this section, we discuss security properties of the PIF. We analyze the resistance to the threats introduced in section 3.3.

Algorithm 2: Filter Update Check

Procedure: Filter Update Check
u_i changes his keyword $W_{i,k}$,
and constructs a new filter tree $\mathcal{FR}'_{u_i}$ with the root node R'_{u_i}.
u_i meets his filter holder u_j.
if u_j has u_i's keyword $W_{i,k}$ **then**
    u_j sends R_{u_i} to u_i for the authentication.
    if R_{u_i} is valid **then**
        if $R_{u_i} \neq R'_{u_i}$ **then**
            u_j searches the changed leaf nodes.
            u_i sends the updated $\mathcal{FR}'_{u_i}$ to u_j.
            u_j updates u_i's filter as $\mathcal{FR}'_{u_i}$.
        end if
    else
        u_i reports u_j to the TA since u_j forges u_i's filter.
    end if
end if
end procedure

3.5.1 Resistance to Inside Curious Users

We discuss the semantic security of the filtering scheme as follows. The coarse-grained filtering scheme allows each filter holder to encrypt his filters before distribution. We say the filtering scheme is semantically secure if no inside curious user $\mathcal{A}$ has a non-negligible advantage in polynomial time against the Challenger in the following game.

- ▷ *Setup*: The challenger takes a security parameter $\mathbf{k}$ and runs $\mathcal{G}$ algorithm. It gives the inside curious user $\mathcal{A}$ system parameters.
- ▷ *Query*: $\mathcal{A}$ issues queries. Each query q_i is related to W_i. The challenger responds by running $\mathsf{FilGen}(W_i, \mathsf{SK}_i) \longrightarrow \mathcal{F}_i$.
- ▷ *Challenge*: $\mathcal{A}$ outputs $W_0, W_1 \in \mathbf{K}$, which is the keyword space. W_0 and W_1 are not queried in the Query phase. The challenger selects a random bit $b \in \{0, 1\}$ and runs $\mathsf{Packet}(\mathsf{PK}_i, W_b) \longrightarrow \varphi_s$. φ_s is sent to $\mathcal{A}$.
- ▷ *Guess*: $\mathcal{A}$ outputs a guess $b^* \in \{0, 1\}$. It wins the game if $b^* = b$.

The security of the PIF can be reduced to several sub-problems, i.e., Bilinear Diffie-Hellman Problem in ($\mathbb{G}$, $\mathbb{G}_T$, e), the discrete logarithm problem in $\mathbb{G}$, and the security of

collision-resistant hash function. Specifically, Bilinear Diffie-Hellman Problem is computational difficult in ($\mathbb{G}$, $\mathbb{G}_T$, e). Given (P, xP, yP, zP) with x, y and z randomly selected from $\mathbb{Z}_q^*$, it is computationally infeasible to compute $e(P,P)^{xyz} \in \mathbb{G}_T$ [95]. The secret key SK_i is securely kept in the filter. Secondly, under the honest-but-curious model, the keyword is computationally indistinguishable in $\mathcal{F}_{i,k} = \frac{\mathsf{H}_1(W_{i,k})}{x_i + \mathsf{H}_1(W_{i,k})} P$ due to the assumption that the discrete logarithm problem in $\mathbb{G}$ is difficult. Finally, due to the security properties of one-way hash function, it is difficult to compute $W_{i,k}$ from $\mathsf{H}_1(W_{i,k})$. Therefore, $\mathsf{Prob}[\mathcal{A}(b^* = b)] = \frac{1}{2} + \epsilon_1$, where ϵ_1 is negligible. The PIF can achieve semantic security and resist inside curious users.

In addition, the filter holder u_j can efficiently check if the keyword in the packet matches any keyword in u_i's filter without disclosing $W_{i,k}$. u_j only forwards the packet with appropriate keywords to u_i. The keyword index is defined by each filter creator. Different creators randomly sort the filters. If the keyword space is not large enough, u_j can take much time to exhaustively search every keyword in the keyword space. In $\mathbb{G}$, the collusion attack algorithm with k traitors (k-CAA) is difficult [95]. Specifically, for an integer k and randomly selected $x \in \mathbb{Z}_q^*$, $P \in \mathbb{G}$, given $(P, Q = xP, h_1, h_k \in \mathbb{Z}_q^*, \frac{1}{h_1+x}P, \cdots, \frac{1}{h_k+x}P)$, there is negligible probability to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \cdots, h_k\}$. Therefore, it is hard for filter holders to guess the keyword inside the filters.

Moreover, an expired time can be added into the filter, and the filter creator can update his filters timely. u_j can only guess the keyword before this expired time. After this expired time, the filter is not valid. The guess on an invalid filter cannot match any keyword within the filter since the time stamp inside the hash function would change the output of hash value. The long guess-time can limit the inside curious user's attacking capability. Furthermore, the filters for different holders are set with different expired time and keyword index, i.e., $\mathcal{F}_{i,k} = \frac{\mathsf{H}_1(W_{i,k}||\mathsf{Time})}{x_i + \mathsf{H}_1(W_{i,k})} P$, where Time indicates the expiry time of certain social activity. The sender and filter creator are supposed to know Time since they have same social interests to exchange information.

In the fine-grained filtering, the keyword vector from sender is invisible to the filter holders u_j. Assume the augmented Decision Linear Problem [97] is computationally infeasible, u_s's private vector $\mathbf{w}'$ cannot be guessed by u_j under the selective security model. The fine-grained vector from the filter holder u_i is visible to u_j. It becomes a trade-off between fine-grained privacy of the creator and the filtering capability of holder. Fortunately, u_i can personalize his vector $\mathbf{w} = (w_1, \cdots, w_l) \in \{1, \cdots, n\}^l$. Take $n = 5$ as an example, u_i has interests in "Health" with the fine-grained degree $(1, 3, 2)$ in different dimensions. In the vector, u_i can change his original fine-grained degree to build a fuzzy searching vector and distribute this fuzzy vector to a specific filter holder. Since the keyword is invisible to

the filter holders, they cannot link the fuzzy fine-grained degree with a specific keyword. In addition, the packet is also encrypted by using the filter creator's public key (i.e., the destination of the packet). The filter holder cannot infer the keyword from the forwarded packet.

3.5.2 Resistance to Filter Forgery

The PIF can detect the forged filters. With Merkle Hash tree, the root value is concatenated from its children nodes. Having the path information from the leaf nodes to the root, each leaf node (i.e., keyword) has a unique certificate generated by the filter creator u_i. The path information is verifiable by others. If the existing filters are changed by u_i, the new certificate is updated. But before the filter update at u_j, the former certificate is still valid. The resistance to filter forgery relies on the security level of hash function used to construct the Merkle tree.

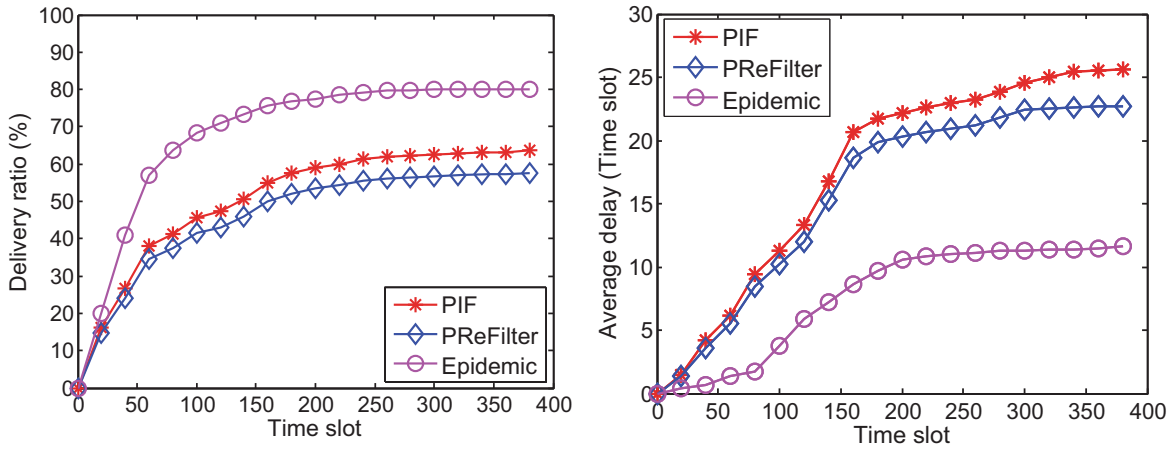
According to the above analysis, the PIF can preserve user's privacy from directly disclosing to inside curious users and resist the forgery attack. Note that the encountered users need to match their profiles to determine the common communities. We follow the security solution from [96] to guarantee the security and privacy requirements during profile matching. In addition, TA can receive the forgery reports from users and revoke the malicious attackers, but does not participate in the communications. Therefore, the PIF operates in a decentralized manner from the perspective of spam filtering and security protections.

3.6 Performance Evaluation

To evaluate the performance of the PIF scheme, we conduct the extensive simulation through Infocom06 trace [102].

3.6.1 Simulation Setup

The Infocom06 trace [102] consists of 78 mobile users during a four-day conference. Every mobile user takes a dedicated portable device to discover the nearby Bluetooth devices every 120 seconds. The system log records mobile users' mobility and contact information. Totally, there are $128,979$ contacts available for the simulation. We then divide the data



(a) Delivery ratio

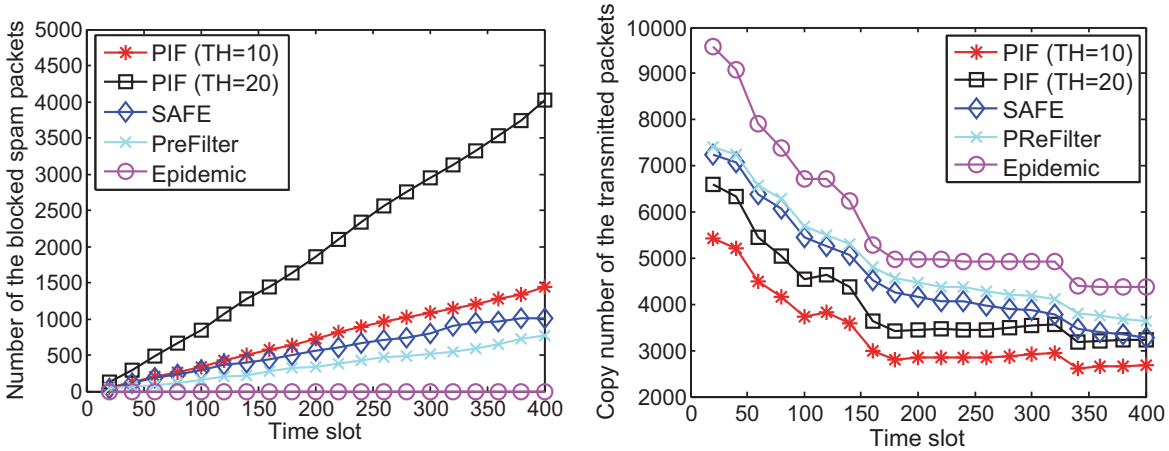
(b) Average delay

Figure 3.5: Packet delivery comparison among different schemes

set into two parts: the training set including one third of the data to produce users' social relations (e.g., communities), and the simulation set including the other two third of the data. We also leverages maximal clique to assign each user's communities. Finally, 100 communities are selected. Every community consists of a sufficient number of users, while the sum of all the edges within the community is large. In every community, there are at least 28 users. On average, every mobile user participates in 38 communities. In the simulation, the time is divided into time slots, and each time slot represents 90 seconds. At the beginning of simulation, we define 100 keywords according to communities where each user selects keywords which are associated with fine-grained interest values from $[1, 100]$ defined by users. Then, each user generates 78 packets with random keywords and interest values to different destination users every 10 time slots.

3.6.2 Simulation Results

We compare the PIF with SAFE [70], PReFilter [69] and Epidemic schemes. The PIF and SAFE have the same delivery ratio and delay, since they do not block any useful packets. Compared with PReFilter, the PIF achieves higher delivery ratio with a reasonable delay as shown in Figure 3.5(a) and 3.5(b). Epidemic scheme allows each user to send his packets to any encountered user such that it achieves the highest delivery ratio with lowest delay. However, it costs many network resources, such as communication and storage. Note that



(a) Number of the blocked spam

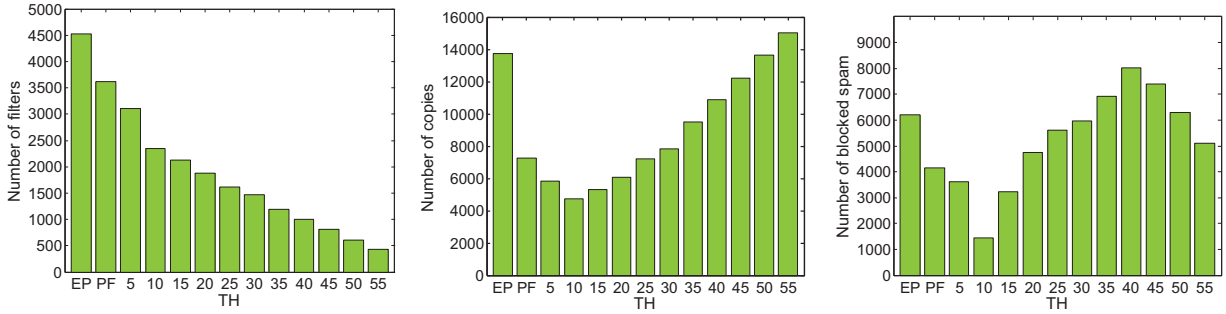
(b) Number of copies

Figure 3.6: Filtering comparison among different schemes

the PIF achieves the same delivery ratio and delay with different THs (i.e., the number of common communities that both encountered users have). It is because the PIF forwards packets based on the common communities with the destination. Only the number of distributed filters is impacted by TH. Therefore, the useful packets can pass the filter check and be forwarded.

In Figure 3.6, we compare the PIF with SAFE and PreFilter in terms of filtering performance. From Figure 3.6(a), the PIF blocks more spam compared with SAFE and PreFilter schemes since the PIF employs find-grained filtering to effectively block the useless packets according to filter creator's defined keyword and fine-grained interests. Meanwhile, the PIF ($TH = 20$) filters more spam compared with the PIF ($TH = 10$). In Figure 3.6(b), the PIF ($TH = 10$) significantly reduces the communication overheads. Although the PIF ($TH = 20$) blocks more spam as shown in Figure 3.6(a), it still produces many copies. It is because the fewer filters are distributed in the network when $TH = 20$, and more users without filters may help to carry-and-forward spam. The PIF ($TH = 10$) can balance the trade-off between the number of copies and the number of blocked spam packets compared with other schemes and settings.

In Figure 3.7(a), when TH increases, the number of distributed filters decreases. During the filter distribution, a smaller TH leads to a larger number qualified users to hold filters. The PReFilter and Epidemic filtering schemes (i.e., PF and Ep in Figure 3.7(a)) distribute too many filters to users. In the PIF, the filter creators purposely distribute their filters



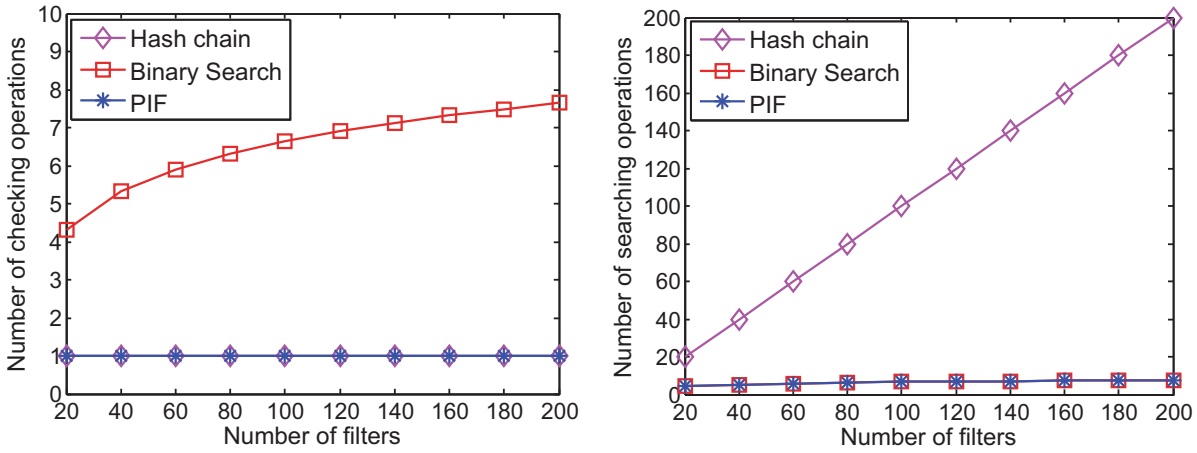
(a) Number of filters vs. TH (b) Number of copies vs. TH (c) Number of blocked packets vs. TH

Figure 3.7: Performance comparison of PIF with different THs

to the users who have more than TH common communities with the filter creators. In Figure 3.7(b), a higher TH causes more copies during the packet delivery. Since the higher threshold decrease the number of distributed filters in the network, the smaller number of filters cannot effectively filter spam. From Figure 3.7(c), we can see that the PIF with an increased TH can block more spam. When TH is small (e.g., 10 or 15), a sufficient number of users hold filters such that they do not duplicate spam. Under this circumstance, spam is filtered at sender's side. When TH increases, fewer users hold filters. The number of produced spam increases, but the number of blocked spam is also increased. With a larger TH (e.g., 45), fewer users hold filters. The spam keeps increasing, but the filtering capability is degraded. In other words, the further increased TH leads to a decreasing number of blocked spam when $TH > 40$. In summary, the PIF ($TH = 10$) achieves the better performance to balance the number of distributed filters and copies (i.e., communication overhead), and efficiently blocks spam packets.

3.6.3 Computational Overhead

In this section, we evaluate the PIF in terms of computational complexity. Denote C_H as a Hash operation ($\{0,1\} \longrightarrow \mathbb{Z}_q^*$), C_M as a multiplication operation in $\mathbb{G}_1$ and C_p as a pairing operation. In the coarse-grained filtering scheme, the filter generation has $1 \cdot C_H + 1 \cdot C_M + 1 \cdot C_p$ operations; the filter holder checks packet sender's keyword with one pairing operation and packet sender only has one multiplication operation to protect his keyword from direct disclosing to the filter holder. For the fine-grained filtering scheme, we do not calculate the time of multiplication operations since exponential operations take much more time than multiplication operations. Denote C_e as an exponential operation



(a) Checking comparison

(b) Searching comparison

Figure 3.8: Update comparison among different schemes

in $\mathbb{G}_1$, and $C_{e'}$ as an exponential operation in $\mathbb{G}_2$. The filter generation has $(6nl + 3) \cdot C_e$ operations. The packet sender has $(5nl + 1) \cdot C_e$ and $1 \cdot C_{e'}$ operations. Finally, the filter holder has 5 paring operations to check if the sender's keyword matches the filter creator's filters.

We compare the filter update complexity as shown in Figure 3.8. Filter update includes two steps: 1) check if the filters need to be updated; and 2) search the out-of-date filter. We compare the PIF with a binary search scheme and a Hash chain scheme (i.e., computing every leaf node's Hash value and checking the concatenation of all these Hash values). From Figure 3.8(a), both the PIF and Hash chain schemes achieve $O(1)$ checking complexity to find if any filter should be updated. The reason is that the Merkle Hash tree based update check only needs to check the root of the distributed filters. The binary search scheme requires an increasing number of operations when more filters are distributed, i.e., $O(log(N))$ where N is the total number of filters. During the searching step, Hash chain scheme requires $O(N)$ searching operations, while both the PIF and binary search schemes only have $O(log(N))$ searching complexity as shown in Figure 3.8(b). Therefore, the PIF can efficiently update the distributed filters.

3.7 Summary

In this chapter, we have proposed a personalized fine-grained spam filtering scheme with privacy preservation in MSNs. Firstly, we have developed a filter distribution scheme based on users' common communities to efficiently distribute filters and block spam. Then, we have proposed coarse-grained and fine-grained filtering schemes with privacy preservation to enable filter creator to personalize his filters. We have also proposed a Merkle Hash tree based filter structure, which can not only authenticate the validity of filters but also update the filters to satisfy user's various demands. The security property analysis demonstrates that filter creator's private information included in his filters can be protected from direct disclosing. In addition, we have conducted the extensive simulations to show that the PIF cannot only reduce the delay as well as the communication and storage overhead but also achieve a high filtering accuracy and efficiency.

Chapter 4

Social Based Mobile Sybil Detection

MSNs are vulnerable to misbehaviors and a series of malicious attacks, which may degrade the network performance or even disrupt MSNs. For example, attackers could forge social profiles to snatch other legitimate users' private information during information exchanging. Attackers may also push and broadcast some biased service/product recommendations and spam over MSNs [70]. Furthermore, if attackers misbehave, e.g., not following network protocols, launching Denial-of-Service (DoS) attacks or maliciously occupying a large amount of network resources, the primary goal of MSNs would not be achieved. Existing misbehavior detection schemes [103] may resist certain types of attacks to some extent. However, how to adjust the security protection against the smart and powerful attacks (e.g., Sybil attacks) becomes a challenging issue in MSNs, especially in mobile environments (User-LS or User-User domains of MSNs). In addition, the cost of misbehavior detection also increases due to the skyrocketing attacking capabilities of Sybil attackers. To support MSNs from Quality-of-Protection perspective, we should consider misbehaviors or attackers from different levels (with different attacking capabilities) when designing detection schemes. In this chapter, we propose a social based mobile Sybil detection scheme to differentiate Sybil attackers from normal users in MSNs with the consideration of multiple levels of attacking capabilities.

4.1 Introduction

Sybil attackers manipulate a large number of identities (or pseudonyms) to profit from services without offering sufficient contribution [51, 26]. Such misbehaviors can compromise the effectiveness of MSNs [104]. For example, in MSN applications, such as WeChat,

Fon11, FireChat and Groovr, users directly exchange or share information via smartphones in the local area or among the crowd. Sybil attackers could maliciously mislead the overall popularity in a voting system, spread spam, or steal legitimate user's private information through forging a large number of fake identities (or pseudonyms). Moreover, Sybil attackers can frequently change their pseudonyms to repeatedly broadcast the same/similar information, e.g., social recommendation and traffic condition. From the perspective of the nearby users, all the same/similar information seems to be sent from different senders such that these legitimate users' opinions and preferences might be misled by Sybil attackers. In addition, mobile Sybil attackers may merge into the crowd or rapidly move with unpredictable trajectories, it is intractable to detect them in MSNs, especially in User-LS or User-User domains.

Extensive research efforts [105, 106, 107] have been put on Sybil detection by using social graph or community detection. Some related works [108, 109, 104] investigate the network characteristics (e.g., wireless channel characteristics), or develop cryptographic mechanisms to detect Sybil attackers. However, MSN users may not easily detect Sybil attackers in User-LS or User-User domains due to the lack of strong social relationships (i.e., social graph), dynamic user mobility and limited detection capabilities. Firstly, users cannot obtain the overall knowledge of the whole network and build social graphs for all users. Moreover, users may not have strong or tight social relationships with each other in the local area, since user's dynamic mobility limits the maintenance of stable social connections for a long time. Without a stable social graph, some traditional social-graph based Sybil detection schemes cannot be directly applied in MSNs. Secondly, Sybil attackers are smart and able to mimic normal users such that they may merge into the normal user crowd or social community. It would disrupt the traditional community-based Sybil detection schemes. Thirdly, users have limited detection capabilities, such as storage and computation. To alleviate their resource consumption during Sybil detection, one possible solution is to leverage the cloud server to assist data storage and computation. However, the cloud server may be untrusted by users, posing critical security and privacy concerns at the same time. In addition, the collusion of users augments Sybil attacker's capabilities and dramatically reduces the detection accuracy. Therefore, it is crucial to take these challenges into account when developing mobile Sybil detection in MSNs.

In this chapter, we propose a Social-based Mobile Sybil Detection (SMSD) scheme to detect Sybil attackers according to their abnormal contact and pseudonym changing behaviors in User-LS and User-User domains of MSNs. Intuitively, since Sybil attackers frequently change their pseudonyms to cheat legitimate users, we investigate the number of contacts and the used pseudonyms. Sybil attackers can be detected by comparing the number of contacts associated with pseudonyms from normal users. Due to user's limited

storage and computation capabilities, the cloud server (as an LS) is involved to store and process the large volume of user's contact information, alleviating users' burden. In addition, the SMSD resists the collusion attacks and data modification when employing the cloud server for mobile Sybil detection. Specifically, the main contributions of this chapter are three-fold.

• Firstly, we investigate the characteristics of user's mobile social behaviors, including pseudonym changing and social contact. We identify four levels of Sybil attackers, i.e., general Sybil attackers, Sybil attackers with forged contact, Sybil attackers with collusion of mobile attackers, and Sybil attackers with collusion of cloud servers according to various attacking capabilities. Then, we propose a social-based mobile Sybil detection scheme to detect mobile Sybil attackers based on their abnormal pseudonym changing behaviors.

• Secondly, we exploit the cloud server to store and process the user's contact data, alleviating users' storage and computing burden. With powerful storage and computational capabilities, the cloud servers can assist to detect the Sybil attackers such that user's storage and computation overhead is significantly reduced.

• Thirdly, we propose a learning assisted SMSD scheme (LSMSD), i.e., semi-supervised learning with hidden Markov model, to resist the collusion of mobile attackers. The LSMSD utilizes a small number of labeled data for training and adapts to the variation of unlabeled data. In addition, a ring structure is built to collect users' contact signatures associated with the bi-directional Hash chain [34], protecting user's contact data (including encountered users, contact time and time order) from being modified by untrusted cloud servers.

The remainder of this chapter is organized as follows. The system model, attacker model and design goals are introduced in Section 4.2. Then, we present the details of the proposed Sybil detection scheme in Section 4.3, followed by security analysis and performance evaluation in Sections 4.4 and 4.5, respectively. We review the related works in Section 4.6 and summarize this chapter in Section 4.7.

4.2 System Model and Design Goals

In this section, we first introduce the system model and security model. Then, we identify the design goals of Sybil detection.

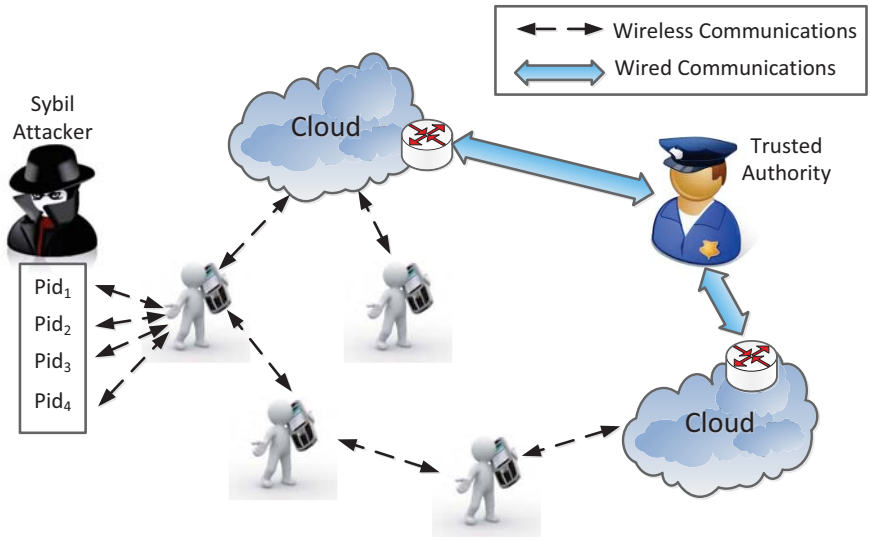


Figure 4.1: System model

4.2.1 System Model

In this chapter, we consider an MSN consisting of three entities: trusted authority, mobile users and cloud servers as shown in Figure 4.1.

• **Trusted Authority (TA)** bootstraps the whole system and generates certificates for mobile users. The TA also audits the mobile users' data that are stored in cloud servers. When a Sybil attacker is detected, the TA revokes the attacker's identity and update the revocation list.

• **Mobile Users** take smart phones or portable communication devices (Bluetooth module) to bi-directionally communicate with other users. A user u_i should first register to the TA for identity and certificates. Then, u_i generates session keys, pseudonyms, and signatures during the social interactions. Note that pseudonyms are used to prevent u_i's real identity from being exposed.

• **Cloud Server (CS)** has powerful storage and computing capabilities. It is an untrusted entity deployed in the local area. The CS can also communicate with mobile users and collect their data.

4.2.2 Security Model

According to the Sybil attacker's capabilities, we define Sybil attackers in four levels.

(1) ***General Sybil Attackers*** (Level-1)

A Sybil attacker (denoted as $\mathcal{A}_s$) exists in User-LS or User-User domain of MSNs to compromise the normal users and launch Sybil attacks to maliciously generate biased information to other users [110]. Having multiple pseudonyms to hide real identity (u_s), $\mathcal{A}_s$'s attacking capability is to repeatedly send the same/similar information and spam to normal users. As a result, the normal user u_i may consider all the received information are from different senders such that u_i's preference may be manipulated by $\mathcal{A}_s$. In other words, given $k' \ll TH$ contacts with $\mathcal{A}_s$, $\mathcal{A}_s$ changes his pseudonym; while, the normal user u_i changes his pseudonym given TH contacts.

(2) ***Sybil Attackers with Illegally Claimed Contact*** (Level-2)

The goal of Sybil attackers in Level-2 is to illegally claim a large number of contacts as the evidences of valid pseudonyms changing. A Sybil attacker $\mathcal{A}_s$ would maliciously claim an extensive number of social contacts associated with his pseudonyms to increase the pseudonym changing frequency. In other words, given $k' \ll TH$ contacts with $\mathcal{A}_s$, $\mathcal{A}_s$ changes his pseudonym and claims to have TH contacts. By claiming this social contact information, $\mathcal{A}_s$ may prevent himself from being detected.

(3) ***Sybil Attackers with Mobile Attacker's Collusion*** (Level-3)

The attacking capability of mobile attackers is to collude with each other and forge inexistent contact in a "legal" way compared with Level-2 attackers. The colluded attackers can mimic as normal users and generate valid proof or signatures for the inexistent contact, although they have not met each other. In other words, given $k' \ll TH$ contacts with $\mathcal{A}_s$, $\mathcal{A}_s$ changes his pseudonym and claims $TH - k'$ inexistent contacts with $\mathcal{A}'_s$. Each inexistent contact is validated by $\mathcal{A}'_s$. From the detector's view, Level-3 attackers have the "reasonable" number of contacts to change their pseudonyms.

(4) ***Sybil Attackers with Collusion of Cloud Servers*** (Level-4)

The CS is involved in Sybil detection and helps mobile users to store their contact data. It is an untrusted entity, although it may honestly follow the protocols. If the CS is compromised or colludes with the Sybil attacker $\mathcal{A}_s$, the CS may either add some fake contact information for $\mathcal{A}_s$, or modify and delete the normal user's contact information to increase the false detection rate of Sybil detection.

4.2.3 Design Goals

To detect Sybil attackers in MSNs, we have the following design goals.

(1) *General Mobile Sybil Detection*

The proposed scheme should be able to detect Level-1 Sybil attacker who maliciously changes his pseudonyms without honestly following the pseudonym changing rules.

(2) *Unforgeability*

The proposed scheme should be able to prevent attackers from forging their social contacts. The encountered users should exchange unforgeable information (e.g., signatures of the contact) to the other user, and keep the integrity of contact information.

(3) *Resistance to Collusion of Mobile Attackers*

The proposed scheme should be able to resist the collusion of mobile attackers and detect the forged inexistent contact when they collude.

(4) *Resistance to Collusion of Cloud Servers*

The data stored at the cloud server should not be maliciously added, modified or deleted by Level-4 attackers. The modified data should be detected by trusted third party or the TA.

4.3 The SMSD Scheme

In this section, we propose the SMSD scheme to detect the identified four levels of Sybil attackers in MSNs. In the SMSD, each user collects the contact information (including contact signatures) from every encountered user. This contact information is used to support the user's pseudonym changing. Collecting the contact information from mobile users, the detector distinguishes Sybil attackers from the normal users by monitoring the abnormal pseudonym changing and contact behaviors. The CS helps mobile users store their contact signatures to reduces mobile user's resource consumption. We also propose a novel Sybil detection scheme by exploiting semi-supervised learning with Hidden Markov Model (HMM) to distinguish the distribution of abnormal contacts (forged by Level-3 attackers) and detect the colluded mobile attackers, as shown in Figure 4.2.

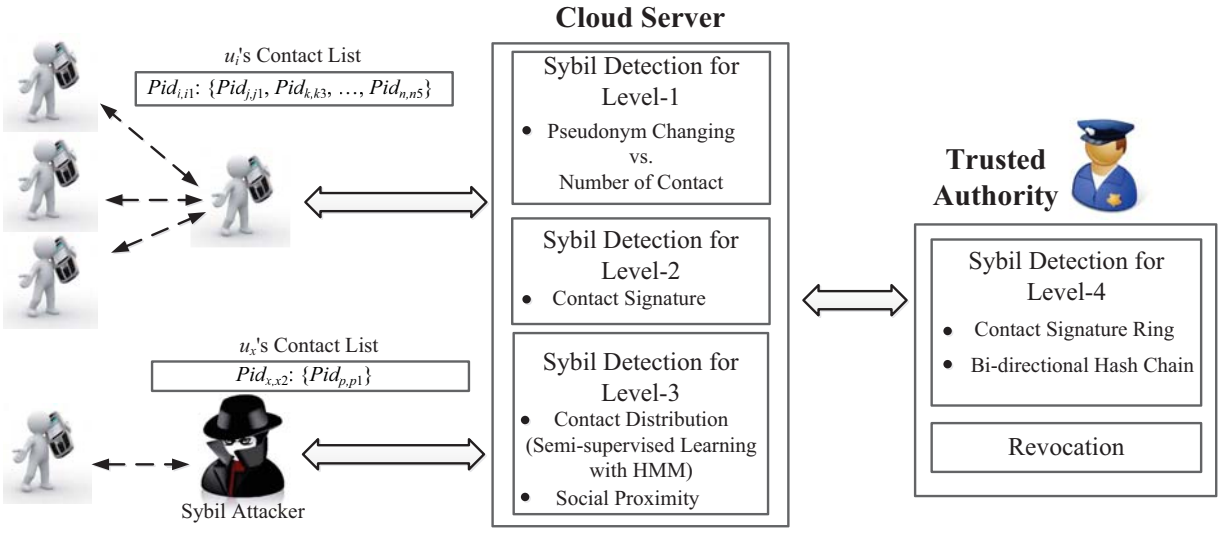
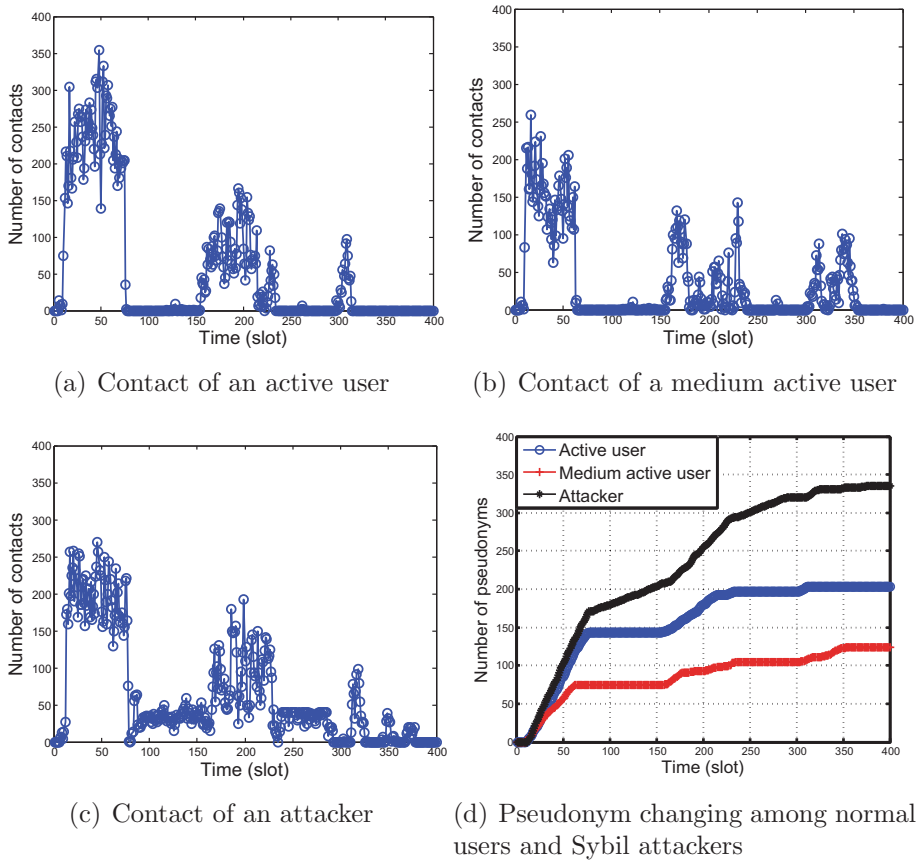


Figure 4.2: Overview of SMSD

4.3.1 Social-based Mobile Sybil Detection

Pseudonym techniques [96] have been widely applied to protect user's real identity and guarantee the anonymity. But the use of pseudonyms may degrade Sybil detections, since mobile users cannot easily link the Sybil attackers' identities given only their pseudonyms. A Sybil attacker $\mathcal{A}_s$ aims to maliciously generate the biased information and convince the normal users. If $\mathcal{A}_s$ uses the same pseudonym to send the same information to a user u_i for multiple times, u_i can easily detect it as spam. However, if $\mathcal{A}_s$ rapidly changes his pseudonyms and sends the same information to u_i with different pseudonyms, u_i may consider the received information is originated from different users. As a result, u_i's preference or decision would be impacted by $\mathcal{A}_s$. It is of utmost importance to ensure mobile users to honestly change pseudonyms only when they are encountered with a certain number of users.

Mobile users usually adopt two types of pseudonym changing strategies: Period Based Pseudonym Changing strategy (PBPC) and k-anonymity based Pseudonym Changing strategy (kPC) to achieve the anonymity. In the PBPC, a normal user u_i can change his pseudonym after a required period (or time window) $\mathbb{T}_s$. When using a pseudonym pid_{i,i_p} with a longer duration than $\mathbb{T}_s$, u_i changes pid_{i,i_p} since it is exposed to the public for a long time and may be linked by others. Normal users cannot frequently or rapidly change their pseudonyms if $\mathbb{T}_s$ is properly defined. The drawback of the PBPC is that u_i cannot adjust



(a) Contact of an active user

(b) Contact of a medium active user

(c) Contact of an attacker

(d) Pseudonym changing among normal users and Sybil attackers

Figure 4.3: Observations on contact and pseudonym changing between normal users and Sybil attackers

the defined period according to the number of contact and environment changes. Alternatively, the kPC enables a normal user u_i to change his pseudonym when k-anonymity [96] is violated. For example, after pid_{i,i_p} is used more than TH times, pid_{i,i_p} should be changed. Here, TH is a pre-defined threshold. Note that it is possible for u_i to change his pseudonym pid_{i,i_p} in a high frequency when pid_{i,i_p} meets many users (more than TH users) within a short period. However, u_i would not always change pseudonyms in such a high frequency in reality.

To understand the relation between contact and pseudonym changing behaviors, we investigate the Infocom06 trace [102], which is a real human trace with 78 mobile users attending a conference within four days. Time is divided into small time slots (10 minutes

for each). We collect the contact and pseudonym changing behaviors from active user (with the highest number of contacts), medium active user (with the average number of contacts), and Sybil attacker in Figure 4.3. Similar to [104], we randomly select users from the trace as attackers, and randomly set the pseudonym changing rate $\frac{1}{k'} \gg \frac{1}{TH}$. In Figure 4.3(d), Sybil attackers adopt multiple pseudonyms under the similar mobility (and the number of contacts) from normal users as shown in Figure 4.3(a) and 4.3(b). Normal users change their pseudonyms following the rule, i.e., changing after TH contacts. In contrast, Sybil attackers may sometimes normally change their pseudonyms to act as normal users, and abnormally change their pseudonyms when launching attacks.

The SMSD exploits users' contact information, i.e., the encountered user's pseudonym and the number of contacts, as the evidence to support their pseudonym changing behaviors. Specifically, the contact between two users with pseudonyms pid_{i,i_p} and pid_{j,j_q} at time t is denoted by $\mathbb{C}_{i_p,j_q} = (pid_{i,i_p}, pid_{j,j_q}, t)$. The kPC is adopted in the SMSD for users to change pseudonyms. The detailed Sybil detection steps are illustrated in Alg. 3. After the Level-1 Sybil detection, the detector reports the Sybil attacker $\mathcal{A}_s$'s pseudonym and the corresponding contact list to the TA.

4.3.2 Contact Signature with Aggregate Verification

According to Level-1 Sybil detection of the SMSD, a pseudonym pid_{i,i_p} with few contacts ($k' \ll TH$) can be detected as a Sybil attacker. To disrupt Level-1 Sybil detection, the Level-2 Sybil attacker $\mathcal{A}_s$ may illegally claim his contact amount to the detector such that $\mathcal{A}_s$ seems to have a "reasonable" number of contacts (TH) to change his pseudonyms. To resist Level-2 Sybil attacks, we propose a contact signature scheme in the SMSD. A contact signature is generated by each pair of the encountered users, and is used as the evidence of the contact. We also develop a variant of aggregate signature [33] to reduce the overall signature size and the verification overhead. This scheme consists of initialization, contact signature, verification, and aggregation authentication as follows.

• ***Initialization***: Let $\mathbb{G}$ and $\mathbb{G}_1$ be additive cyclic groups with the same prime order q, and P is the generator of $\mathbb{G}$. $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$, and $\mathsf{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ are two cryptographic hash functions. Let e be a bilinear pairing [111], where e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ [33] between $\mathbb{G}$ and $\mathbb{G}_1$ exists under two conditions: 1) for any random numbers $a, b \in \mathbb{Z}_q^*$, $e(aP, bP) = e(P, P)^{ab}$; 2) $e(P, P) \neq 1$. Taking a security parameter κ as input, a probabilistic algorithm outputs a tuple (q, $\mathbb{G}_1$, $\mathbb{G}$, e, P, H, H_1) as the system parameters to the public.

u_i receives a series of pseudonyms pid_{i,i_1}, pid_{i,i_2}, ..., pid_{i,i_n} from the TA. Each pseudonym pid_{i,i_p} is assigned with the corresponding secret key pair $\mathsf{SK}_{i,i_p} = (sk_{i_p,0}, sk_{i_p,1})$, where

Algorithm 3: SMSD

Input: user u_i with pseudonym pid_{i,i_p},
an initialized contact list $\mathcal{CL}_{i,i_p}$, and pseudonym changing threshold TH
Output: $\mathcal{CL}_{i,i_p}$ and Sybil detection
while $|\mathcal{CL}_{i,i_p}| < TH$ ($|\mathcal{CL}_{i,i_p}|$ denotes the number of items) **do**
    if pid_{i,i_p} is encountered with another user pid_{j,j_q} **then**
        They generate $\mathbb{C}_{i_p,j_q} = (pid_{i,i_p}, pid_{j,j_q}, t)$.
        pid_{i,i_p} adds $\mathbb{C}_{i_p,j_q}$ into $\mathcal{CL}_{i,i_p}$.
    end if
end while
u_i changes pid_{i,i_p} to $pid_{i,i_{p+1}}$.
Sybil Detection:
Having $\mathcal{CL}_{i,i_p}$, the detector first checks if (1) $|\mathcal{CL}_{i,i_p}| < TH$ and (2) $\mathbb{T}_{p-1} < t_1 < \cdots < t_j < \cdots < t_n < \mathbb{T}_p$. Here, $\mathbb{T}_{p-1}$ and $\mathbb{T}_p$ are starting and ending time of pseudonym pid_{i,i_p}
if Both (1) and (2) are not guaranteed at the same time **then**
    pid_{i,i_p} is maliciously used. u_i is a Sybil attacker.
else
    pid_{i,i_p} is legitimately used.
end if

$sk_{i_p,0} = s_{i,i_p}\mathsf{H}(pid_{i,i_p}||0)$, and $sk_{i_p,1} = s_{i,i_p}\mathsf{H}(pid_{i,i_p}||1)$. $s_{i,i_p} \in \mathbb{Z}_q^*$ is selected by u_i. The public key is $\mathsf{PK}_{i,i_p} = s_{i,i_p}P$.

• ***Contact Signature***: When two users pid_{i,i_p} (from u_i) and pid_{j,j_q} (from u_j) are encountered, pid_{i,i_p} generates the contact as $\mathbb{C}_{i_p,j_q} = \{pid_{i,i_p}, pid_{j,j_q}, t\}$. pid_{i,i_p}'s signature of the contact between pid_{i,i_p} and pid_{j,j_q} at time t is

$$\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,j_q}) = (pid_{j,j_q}, \omega_{i_p}, \theta_{i_p}) \tag{4.1}$$

$$\begin{cases} \omega_{i_p} = r_{i_p}\mathsf{H}(pid_{j,j_q}) + sk_{i_p,0} + c_{i_p}sk_{i_p,1} \\ \theta_{i_p} = r_{i_p}P \end{cases} \tag{4.2}$$

where $c_{i_p} = \mathsf{H}_1(t||pid_{i,i_p}||pid_{j,j_q})$, and $r_{i_p} \in \mathbb{Z}_q^*$ is a random number. Finally, pid_{i,i_p} sends $\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,j_q})$ to pid_{j,j_q} as the unforgeable signature to prove the contact $\mathbb{C}_{i_p,j_q}$.

• ***Verification***: After receiving the contact signature from the encountered user, pid_{j,j_q} verifies its authenticity as

$$e(\omega_{i_p}, P) \stackrel{?}{=} e(\theta_{i_p}, \mathsf{H}(pid_{j,j_q})) \cdot e(\mathsf{H}(pid_{i,i_n}||0) + c_{i_p}\mathsf{H}(pid_{i,i_n}||1), \mathsf{PK}_{i,i_p}). \tag{4.3}$$

If Equation 4.3 holds, the received signature is valid; otherwise, it is invalid. Then, pid_{j,j_q} replies $\mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q}) = (pid_{i,i_p}, \omega_{j_q}, \theta_{j_q})$ to pid_{i,i_p}. These signatures can be stored and used as the evidence of user's pseudonym changing behaviors.

Table 4.1: Comparison of Computation Complexity

	Sign	Verification
S	$C_{H_p} + 3C_M$	$3C_{H_p} + 3C_p + 2C_M$
S_{agg}	$N \cdot C_{H_p} + 3N \cdot C_M$	$(2N+1) \cdot C_{H_p} + (N+2) \cdot C_p + (N+1)C_M$

• ***Aggregate Authentication***: When u_j changes his pseudonym from pid_{j,j_q} to $pid_{j,j_{q+1}}$, u_j collects all the contact signatures related to pid_{j,j_q} and sends them to the CS. As the ever-growing number of encountered users, the volume of signatures increases correspondingly. We develop an aggregate authentication scheme to reduce the communication and computation overhead of authentication. First, u_j aggregates the signatures $\mathsf{Sign}_{agg} = (\Omega_{agg}, \Theta_{agg}, pid_{j,j_q})$ of $(pid_{1,1_a}||pid_{2,2_b}|| \dots ||pid_{i,i_p}|| \dots ||pid_{n,n_x}, t_1||t_2|| \dots ||t_i|| \dots ||t_n, pid_{j,j_q})$ where

$$\Omega_{agg} = \sum_{i=1}^{n} \omega_{i_p}, \Theta_{agg} = \sum_{i=1}^{n} \theta_{i_p}. \tag{4.4}$$

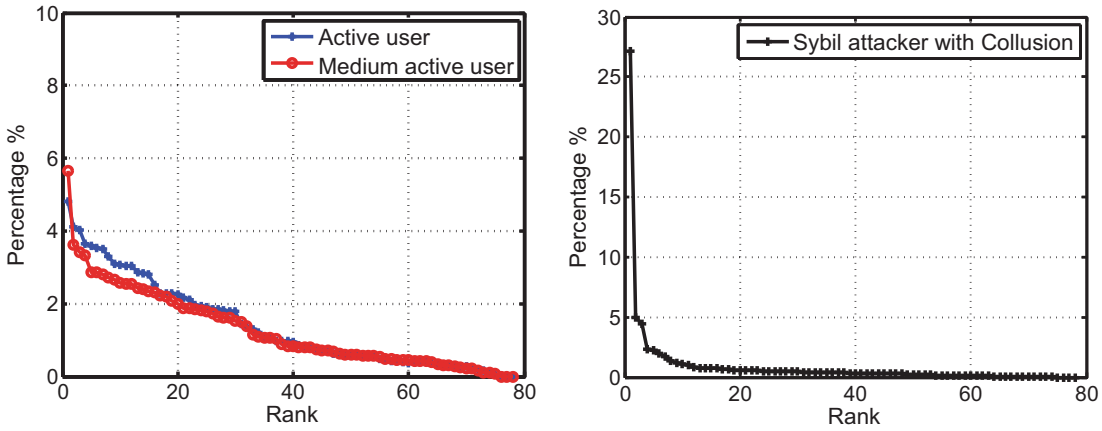
Then, u_j sends the aggregate signature Sign_{agg} to the CS for authentication. To verify pid_{j,j_q}'s aggregate signature, the CS checks

$$e(\Omega_{agg}, P) \stackrel{?}{=} e(\Theta_{agg}, \mathsf{H}(pid_{j,j_q})) \cdot \prod_{i=1}^{N} e(\mathsf{H}(pid_{i,i_p}||0) + c_{i_p}\mathsf{H}(pid_{i,i_p}||1), \mathsf{PK}_{i,i_p}).$$

If it does not hold, some of pid_{j,j_q}'s contact signatures are invalid by pid_{j,j_q} or other users. Note that during each contact, pid_{j,j_q} should check the validity of the received signatures at the beginning. In other words, each stored contact signature by pid_{j,j_q} should be valid by pid_{j,j_q}'s verification. The invalid signatures would be forged by pid_{j,j_q}. Therefore, the CS could directly detect the Level-2 Sybil attacker.

As the contact signature inevitably increases the communication, computation and storage overhead, we adopt the cloud server to replace mobile users as the detector. We show the computation complexity in Table 4.1, where C_{H_p} is map-to-point Hash operation, C_M is multiplication, and C_p is pairing operation. From Table 4.1, our aggregate signature scheme can significantly reduce the verification overhead.

If u_i holds all the contact signatures, u_i should provide his historic contact signatures for other user's detection. It would directly expose his past pseudonyms, while the authentication overhead exponentially increases as u_i meets more users. In the SMSD, the CS takes u_i's contact signatures and verifies once for each pseudonym. Then, the CS signs a receipt for the successful detection to u_i. Thus, u_i can adopt this verified receipt to



(a) Contact rate distribution of normal users (b) Contact rate distribution of Sybil attackers

Figure 4.4: Comparison of contact rate distribution between normal users and Sybil attacker

prove his validity instead of authenticating his past pseudonyms associated with contacts to every individual user.

4.3.3 Learning Assisted Mobile Sybil Detection

Level-3 Sybil attacker $\mathcal{A}_s$ may collude with other mobile attackers to disrupt the Level-1 and Level-2 Sybil detections by generating valid signatures for inexistent contacts with $\mathcal{A}_s$. The inexistent contact between $\mathcal{A}_s$ and the colluded attackers may increase the total number of $\mathcal{A}_s$'s contact which makes his abnormal pseudonym changing "legal". To tackle this problem, we propose a Learning assisted SMSD scheme (LSMSD) to detect Level-3 Sybil attackers. Specifically, the LSMSD consists of three steps: contact rate distribution, semi-supervised learning with HMM, and social proximity evaluation.

Contact Rate Distribution

To detect the collusion of mobile attackers, we first analyze the contact rate distribution of each user. When two users are frequently encountered, they are expected to stay in the physical proximity. If two users are colluded, they would have a very high contact rate with each other to compromise mobile Sybil detection. Meanwhile, they likely have a limited number of contacts with other users. We extract several samples from the real world trace

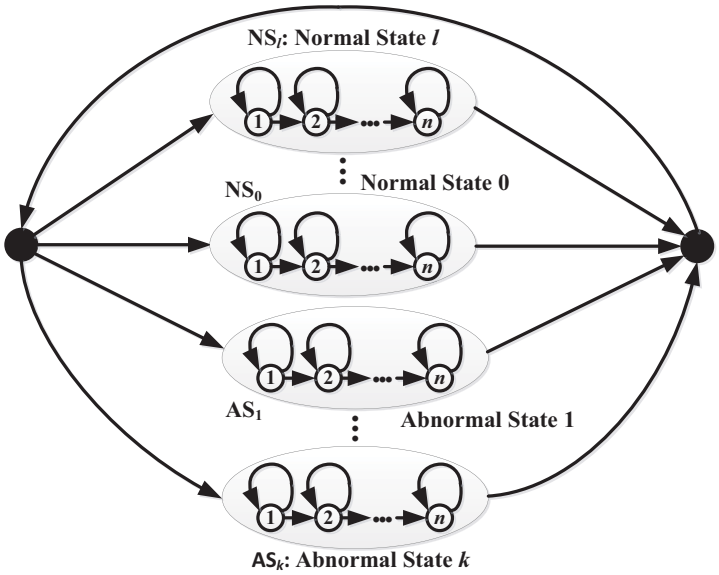


Figure 4.5: Hidden Markov model

as shown in Figure 4.4. The percentage in y-axis represents the contact number for each pair of encountered users. It equals the contact number of individual user over the total contact number. The contact rate distribution could be approximated to an exponential distribution. The detector (e.g., the CS, the TA or other trusted party) can form sequences to represent the contact distribution for classification.

Semi-supervised Learning with HMM

We propose a semi-supervised learning scheme with HMM to detect the collusion of mobile attackers. First, we form an ergodic k-class HMM to analyze the abnormal contact distribution, where k is the amount of abnormal states in HMM and there exist multiple normal states. In the initialization, there is only one normal central state NS_0 in HMM as shown in Figure 4.5. Then, l normal states and k abnormal states are generated based on training and labeling. NS_0 can be obtained by training from a certain number of contact distribution samples. In the ground truth data set, we select user's contact distribution during daytime and night time from Infocom06 trace [102] to adjust different users' mobilities and social behaviors.

A set of parameters θ^* of normal state HMM is obtained by maximizing the likelihood

Algorithm 4: LSMSD

1: **Input**: A set of N_l labeled contact distributions $\mathbb{C}^{(l)} = \{c_1^{(l)}, \cdots, c_{N_l}^{(l)}\}$, N_u unlabeled contact distributions $\mathbb{C}^{(u)} = \{c_1^{(u)}, \cdots, c_{N_u}^{(u)}\}$.
2: **Output**: Trained HMM $\Theta = \{\theta_1, \cdots, \theta_K\}$ where $K = l + k$. The classification of the contact distributions $\mathbb{NS} = \{NS_1, \cdots, NS_l\}$ and $\mathbb{AS} = \{AS_1, \cdots, AS_k\}$.
3: Step 1 (**Supervised training**): Given the ground truth data, estimate the central state NS_0 as $\theta^* = \arg\max_{\theta} \prod_{j=1}^{N_l} P(c_j^{(l)}|\theta)$.
4: **while** The number of iterations is small than that of abnormal states **do**
5: Step 2 (**Outlier**): Having a sliding window ω, split the contact distribution into Ω segments with overlapping.
Select the outlier with the smallest likelihood as $s^* = \arg\min_{s} \left(\arg\max_{s} \prod_{j=1}^{\Omega} P(c_j^{(l)}|s) \right)$.
Label this contact distribution as an abnormal state.
6: Step 3 (**Adaptation**): A new abnormal state model AS_i is adapted from the general model via the abnormal detection. The normal state model is adapted from the general model by using the other segments.
7: Step 4 (**Boundary**): Determine the boundary of states
8: Step 5 (**New Outlier**): Select a new state model with the smallest likelihood in the adaptive normal state model as an outlier.
9: **end while**

of the ground truth sequences (contact distribution) $\{c_1^{(l)}, \cdots, c_{N_l}^{(l)}\}$ as

$$\theta^* = \arg\max_{\theta} \prod_{j=1}^{N_l} P(c_j^{(l)}|\theta). \tag{4.5}$$

We assume that each HMM state follows the Gaussian Mixture Model (GMM), which can be estimated by standard Expectation-Maximization (EM) algorithm [112]. The concrete steps of semi-supervised learning algorithm with HMM are stated in Alg. 4.

In the adaptation phase, Maximum A Posteriori (MAP) [113] scheme is adopted to adjust the normal state model to a certain abnormal state model for training on the abnormal state model. The original normal state model is also trained by adapting the non-outlier segments. θ^* is selected to maximize posterior probability density as

$$\theta^* = \arg\max_{\theta} P(\theta|C) = \arg\max_{\theta} P(C|\theta)P(\theta). \tag{4.6}$$

Having GMM, the model is adapted according to the new weight, mean and variance, denoted by w_i', μ_i' and σ_i' according to Equation 4.7.

$$
\begin{aligned}
w_i' &= \frac{1}{L}\sum_{j=1}^{L} P(i|c_j,\theta) \\
\mu_i' &= \frac{\sum_{j=1}^{L} c_j P(i|c_j,\theta)}{\sum_{j=1}^{L} P(i|c_j,\theta)} \\
\sigma_i' &= \frac{\sum_{j=1}^{N} P(i|c_j,\theta)(c_j-\mu_i')(c_j-\mu_i')^T}{\sum_{j=1}^{L} P(i|c_j,\theta)}.
\end{aligned} \tag{4.7}
$$

The adaptive parameters can be updated as

$$
\begin{aligned}
\widehat{w}_i =& \beta \cdot w_i + (1-\beta)\cdot w_i' \\
\widehat{\mu}_i =& \beta \cdot \mu_i + (1-\beta)\cdot \mu_i' \\
\widehat{\sigma}_i =& \beta \cdot (\sigma_i + (\widehat{\mu}_i-\mu_i)(\widehat{\mu}_i-\mu_i)^T) + (1-\beta)\cdot(\sigma_i' + (\widehat{\mu}_i-\mu_i')(\widehat{\mu}_i-\mu_i')^T).
\end{aligned} \tag{4.8}
$$

Here, w_i, μ_i and σ_i are the previous weight, mean and variance, respectively. β is the adaption factor to balance the new parameters and the previous ones. When β becomes larger, the new parameters contributes more in the adapted model. In step (4) of Alg. 4, we determine the boundary of states based on Viterbi decoding [112].

The LSMSD only needs a small amount of ground truth data in the training phase, which considerably reduces the training overhead and is suitable for mobile Sybil detection. In HMM, we also establish l normal states which leverages from the active users to inactive ones; and from the daytime to the night time. With the adaption on the HMM, the LSMSD can improve the detection accuracy even given a large amount of unlabeled data.

Social Proximity Evaluation

Although the LSMSD can detect and identify the abnormal contact distribution, it may generate false detection when normal users always stay together. To solve this problem, we investigate the social community and enhance the LSMSD by using this social feature. In reality, if two users frequently meet each other, they should have certain social relationships, such as colleagues, social friends and neighbors. We extract these social features and

form social communities in MSNs. Let each user u_i maintain a social community vector $\overrightarrow{SC_i} = [1, 0, 0, \ldots, 1, 0]$. We define the social proximity $SP_{i,j}$ between u_i and u_j as

$$SP_{i,j} = \frac{|\overrightarrow{SC_i} \bigcap \overrightarrow{SC_j}|}{|\overrightarrow{SC_i} \bigcup \overrightarrow{SC_j}|} \in [0, 1]. \tag{4.9}$$

We define a social proximity threshold SP as the reasonable value of social proximity that normal friends have according to the investigation of human trace. Given the detection results from the LSMSD, the user pair (u_i, u_j) with the social proximity $SP_{i,j} < SP$ is labeled as the Level-3 Sybil attackers. The LSMSD is thus enhanced with the comparison of social characteristics.

4.3.4 Ring Structure of Contact Signature

In the aforementioned sections, Level-1, Level-2 and Level-3 Sybil detections are proposed by exploiting the relation between contact and pseudonym changing behaviors, aggregate signatures of contact, and the contact rate distribution, respectively. In reality, the CS is the untrusted entity as indicated in Section 4.2, and is possibly compromised. To resist Level-4 Sybil attacks, we develop a ring structure of contact signatures to prevent users' data from be deleted or modified. Before sending the contact list to the CS, each user builds his contact list in a ring structure.

1) u_i first initializes the contact list $\mathcal{CL}_{i_p}$ for pid_{i,i_p}. When u_i begins to use a pseudonym pid_{i,i_p} at time t_0, the contact list is $\mathcal{CL}_{i_p} = \{\mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p})\}$, where $\mathbb{C}_{i_p,i_p} = (pid_{i_p,i_p}, pid_{i_p,i_p}, t_0)$.

2) When pid_{i,i_p} meets pid_{j,j_q} at t_1, u_i obtains the contact signature $\mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q})$, and updates the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, \mathsf{Sign}_{\mathsf{SK}_{j,j_q}}(\mathbb{C}_{i_p,j_q})\}$, where $R_1 = (pid_{i,i_p}, t_0, \mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p}))$.

Similarly, when another user pid_{l,l_r} is encountered with pid_{i,i_p} at t_2, pid_{l,l_r} sends the contact signature $\mathsf{Sign}_{\mathsf{SK}_{l,l_r}}(\mathbb{C}_{i_p,l_r})$ to pid_{i,i_p}. pid_{i,i_p} then updates the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, R_2, \mathsf{Sign}_{\mathsf{SK}_{l,l_r}}(\mathbb{C}_{i_p,l_r})\}$, where $R_2 = (pid_{j,j_q}, t_1, \mathsf{Sign}_{\mathsf{SK}_{i,i_p}}(\mathbb{C}_{i_p,i_p}))$.

3) pid_{i,i_p} recursively builds the ring structure following step 2). When u_i changes pseudonym pid_{i,i_p} at t_N, u_i finalizes the contact signature ring as $\mathcal{CL}_{i_p} = \{R_1, R_2, \ldots, R_N, \mathsf{Sign}_{\mathsf{SK}_{i_p}}(\mathbb{C}'_{i_p,i_p})\}$, where $\mathbb{C}'_{i_p,i_p} = (pid_{i_p,i_p}, pid_{i_p,i_p}, t^*)$ and $t^* = \mathsf{H}_1(t_0||t_1||\ldots||t_N)$.

In addition, u_i generates a contact order list $\mathcal{CO}_{i,i_p} = \{CO_0, CO_1, \cdots, CO_N\}$. Let H_2 and H_3: $\{0,1\}^* \rightarrow \mathbb{Z}_q^*$ be two cryptographic hash functions. u_i adopts pid_{i,i_p} in the

duration $[t_0, t_N]$, and has contact at $\mathbb{T} = \{t_1, t_2, \cdots, t_{N-1}\}$. For the n-th contact, $CO_n = \mathsf{H}_1(h_n||pid_{j,j_q})$, where pid_{j,j_q} is the encountered user. Here, $h_n = \mathsf{H}_1(\mathsf{H}_2(h_{n+1}||t_{n+1}) \oplus \mathsf{H}_3(h_{n-1}||t_{n-1}))$ where $n \in [1, N-1]$. As such, a bi-directional Hash chain is established, where the forward seed is $h_0 = \mathsf{H}_1(t_0)$ and backward seed is $h_N = \mathsf{H}_1(t_N)$.

The contact signatures form a closed ring, while the established bi-directional Hash chain guarantees the order of every contact time. The contact list should be synchronized with the contact order list to ensure the integrity of the contact information provided by mobile users.

4.4 Security Analysis

In this section, we discuss the security properties of the SMSD scheme according to the defined attacker model in Section 4.2.

4.4.1 General Mobile Sybil Detection (Level-1)

The SMSD scheme can detect general Sybil attack when user's contact and pseudonym changing behaviors are correlated. If a Level-1 attacker $\mathcal{A}_s$ rapidly or frequently changes his pseudonyms, $\mathcal{A}_s$ hardly collects sufficient contacts to validate his pseudonym changing behaviors within a very short period. In other words, $\mathcal{A}_s$ changes his pseudonym when having $k' \ll TH$ contacts. By collecting contacts related to each pseudonym, the SMSD can identify the behavior difference between normal users and Level-1 attackers This kind of difference can directly reflect their primary purposes of participating in MSNs.

Although $\mathcal{A}_s$ sometimes mimics normal users and does not rapidly or frequently change his pseudonyms, the pseudonyms changed within the short period or with few contacts can also be easily detected. The SMSD can cause a higher resource consumption and reduce Level-1 Sybil attack's attacking capabilities.

4.4.2 Contact Unforgeability of Mobile User (Level-2)

Theorem 1 *The SMSD can resist Level-2 attackers from illegally claming contacts via contact signatures.*

Proof. Given $k' \ll TH$ contacts of pid_{j,j_q} (from $\mathcal{A}_s$), $\mathcal{A}_s$ changes his pseudonym pid_{j,j_q} and claims to have TH contacts related to pid_{j,j_q}. However, during the usage period of

pid_{j,j_q}, when pid_{j,j_q} meet another user pid_{i,i_p}, they sign on the contact event, including the encountered pseudonyms and the time, by using secret keys. Moreover, $sk_{i_p,0}$, $sk_{i_p,1}$ and r_{i_p} are selected and secretly kept by pid_{i,i_p}. The forgeability of contact signature $\omega_{i_p} = r_{i_p}\mathsf{H}(pid_{j,j_q}) + sk_{i_p,0} + c_{i_p}sk_{i_p,1}$ and $\theta_{i_p} = r_{i_p}P$ can be reduced to the computational Diffie-Hellman problem (CDH) in $\mathbb{G}$, i.e., given $P, aP, bP \in \mathbb{G}$ where $a, b \in \mathbb{Z}_q^*$, to compute abP. Since the CDH problem in $\mathbb{G}$ is computational difficult for a polynomial-time adversary [33], the proposed contact signature scheme is unforgeable under the defined attacker model. As a result, $\mathcal{A}_s$ only collects k' valid contact signatures. The illegally claimed $TH - k'$ contacts can be detected. Therefore, the contact signature can validate the authenticity of the contact event and prevent Level-2 attackers from illegally claiming contacts. □

4.4.3 Resistance to Collusion of Mobile Attackers (Level-3)

The LSMSD can resist the collusion of mobile attackers through semi-supervised learning with HMM on the contact rate distribution and social proximity comparison. When u_i colludes with $\mathcal{A}_s$, u_i can generate "valid" signatures for some inexistent contact with $\mathcal{A}_s$, such that $\mathcal{A}_s$ can change his pseudonyms prior to the normal changing time point. $\mathcal{A}_s$ and u_i would have a large number of contact, reflecting a high contact rate in their contact distribution. However, $\mathcal{A}_s$ may not meet other users frequently, such that $\mathcal{A}_s$ may have lower contact rates with other users. As shown in Figure 4.4, normal users and Level-3 attackers have different contact distribution. Therefore, the proposed semi-supervised learning with HMM can classify the normal contact distribution and the abnormal one due to the collusion. The detection accuracy will be presented in Section 4.5.

As an enhancement of the LSMSD, social proximity is explored to assist the contact distribution. Since the colluded attackers may not have strong social connections with $\mathcal{A}_s$ but frequently contact with $\mathcal{A}_s$, the colluded attackers can be identified according to their social relationships.

4.4.4 Resistance to Collusion of Cloud Server (Level-4)

Theorem 2 *The CS cannot add, modify and remove the user's contacts due to the contact signature ring and bi-directional Hash chain of contact order.*

Proof. Suppose the CS deletes the contact between pid_{i,i_p} and pid_{j,j_q} at t_j. $h_{j-1} \neq \mathsf{H}_1(\mathsf{H}_2(h_{j+1}||t_{j+1}) \oplus \mathsf{H}_3(h_{j-2}||t_{j-2}))$. Similarly, h_{j+1} cannot be recovered as well. As a result, the whole contact order list is invalid. Due to the forward and backward secrecy,

the contact order list cannot be forged. If the CS modifies or adds contact signatures for any user, the detectors can find out the CS's malicious operations due to Theorem 1.

In the contact signature ring, if R_2 (e.g., from pid_{j,j_q}) is deleted, t^* cannot be calculated without t_2. Similarly, if the CS adds R_j^* into $\mathcal{CL}_{i_p}$, the contact signature ring cannot be synchronized with the order list. Therefore, the proposed contact signature ring and bi-directional Hash chain can protect the stored contact information from addition, modification and deletion by the CS. □

In summary, the SMSD scheme can resist the four levels of Sybil attackers considered in Section 4.2.

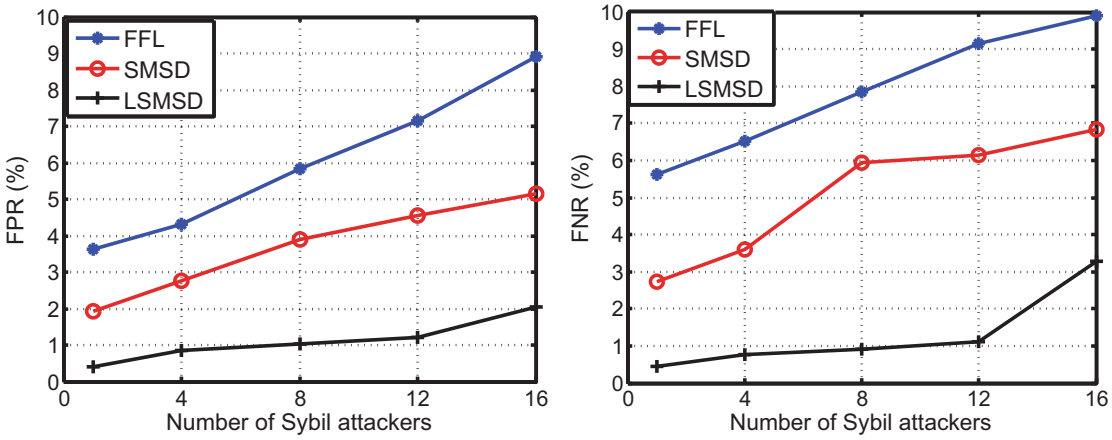
4.5 Performance Evaluation

Based on a trace-driven simulation, we evaluate the performance of the SMSD compared with other mobile Sybil detection schemes in terms of detection accuracy .

4.5.1 Simulation Setup

We conduct the simulation on a real world human trace (Infocom06 trace) [102]. This trace contains 78 mobile users attending a four-day conference. Mobile users carry the dedicated Bluetooth modules, discovering the surrounding users via Bluetooth. There are totally $128,979$ recorded contacts in this trace. We separate the entire data set into two parts: 20% of data are the training set to produce mobile users' profiles (e.g., social communities), and the remaining data are used for the simulation.

We assign users with social communities according to the sociology theory [80]. A complete graph G is built up, where each edge $E(u_i, u_j)$ weighted by the total number of contacts between two vertex u_i and u_j. We then refine the graph G with 78 vertices and $2,863$ edges by removing the edges weighted smaller than 100. We adopt Bron-Kerbosch algorithm [114] to extract maximal cliques in G. Each clique is a complete subgraph where every edge is high-weighted. We select 100 social communities (i.e., cliques) based on the weight of each maximal clique. The selected communities are used for simulation comparison on social connections.



(a) False positive rate vs. the number of Sybil attackers

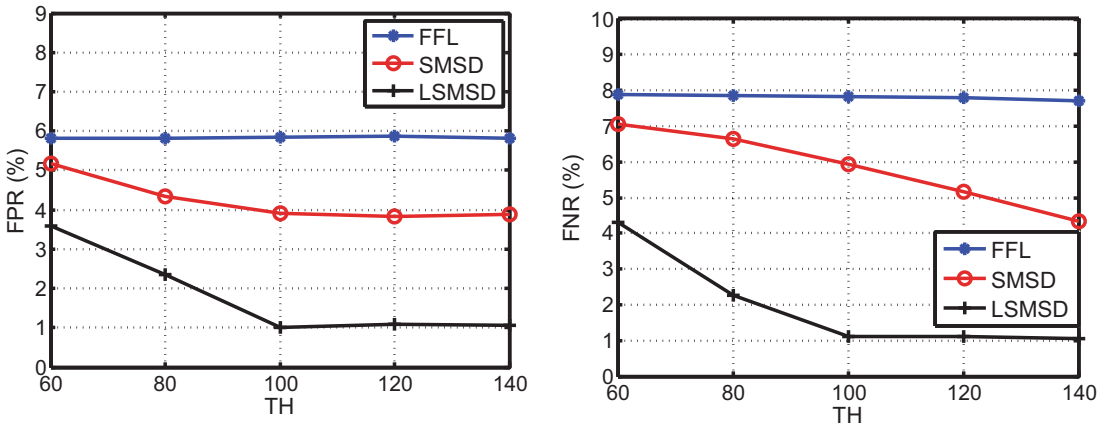
(b) False negative rate vs. the number of Sybil attackers

Figure 4.6: The impacts of the number of Sybil attackers

4.5.2 Simulation Results

To demonstrate the advantages of SMSD and LSMSD schemes, we compare the detection accuracy with FFL (Friend and Foe list) [104]. In FFL, mobile users detect attackers based on checking their social friend list. Similarly to [104], we randomly select 1, 4, 8, 12, and 16 users as the attackers in our simulation. To quantify the detection accuracy, we utilize false positive rate (FPR) and false negative rate (FNR) as the metrics to evaluate the Sybil detection accuracy. A false positive detection results in a normal user being detected as a Sybil attacker, while a false negative indicates that a Sybil attacker is labeled as a normal user. $\mathsf{FPR} = P_f/(P_f+N)\times 100\%$, where P_f denotes the number of false positive detections and N is the total number of Sybil attackers. Similarly, $\mathsf{FNR} = N_f/(N_f+P)\times 100\%$, where N_f denotes the number of false negative detections, P is the total number of normal users. We conduct the simulations of different schemes within 400 time slots. For the LSMSD, we set adapter factor $\beta = 0.5$ and select 100 contact distributions from different users as the training data.

As shown in Figure 4.6, we compare the FPRs and FNRs of FFL, SMSD and LSMSD when the number of Sybil attackers increases. Note that $TH = 80$ and $SP = 0.3$. The number of Sybil attackers has a greater impact on FFL compared with that on SMSD and LSMSD. The increasing number of attackers when using FFL (friend and foe lists) increases the number of contacts between normal users and Sybil attackers such that the detection error increases. This number can also affect SMSD and LSMSD since a large



(a) False positive rate vs. TH

(b) False negative rate vs. TH

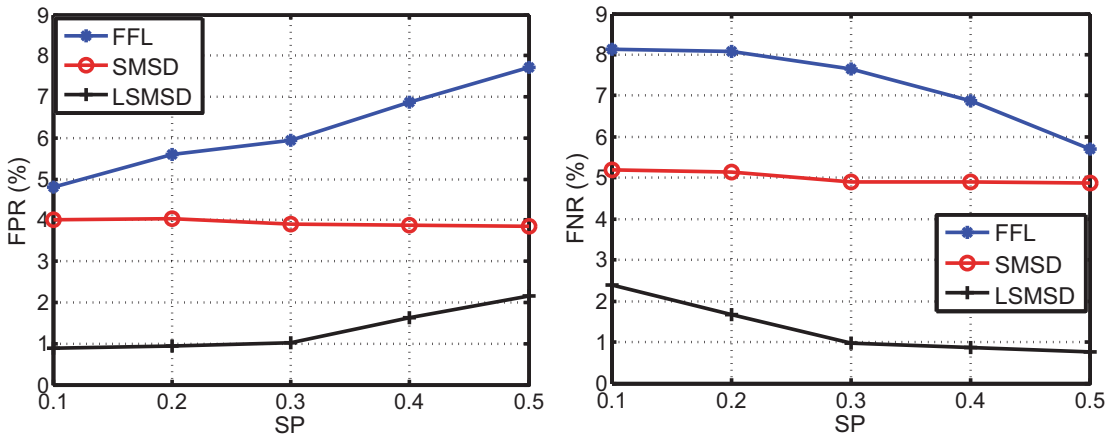
Figure 4.7: The impacts of TH (i.e., every user changes pseudonyms when the pseudonym meets more than TH users)

number of Sybil attackers can launch strong collusion attack and forge contact signatures to disrupt the SMSD and LSMSD. In the following results, we set 8 Sybil attackers in the network.

We compare FPRs and FNRs of different schemes by changing the threshold (number of contacts) for changing pseudonyms. We set $SP = 0.3$ for the LSMSD. As shown in Figure 4.7, when TH is small, e.g., $TH = 60$, the FPRs and FNRs of SMSD and LSMSD are not high. The reason is that Sybil attackers can mimic normal users. A smaller TH results in a smaller gap with the number of contacts that an attacker has. When TH increases, the gap becomes larger such that the FPRs and FNRs of SMSD and LSMSD are dramatically reduced.

As shown in Figure 4.8, the social proximity threshold SP can only impact on LSMSD which detects Level-3 attackers. We set $TH = 80$. For FFL, we adopt SP as the threshold to befriend with others. When SP is large, the Sybil attackers would not befriend with normal users such that the FNR reduces. Meanwhile, a large SP prevents some normal users from befriending with others. Therefore, they may be detected as attackers, which increases the FPR.

In terms of the users with high contact rate, the LSMSD (by exploiting social proximity) can detect whether their contacts are forged or not. If SP is small, the FNR increases, since the colluded attackers may also have certain social connections. It is easy to achieve such that both normal users and Sybil attackers with high contact rate are likely detected



(a) False positive rate vs. SP

(b) False negative rate vs. SP

Figure 4.8: The impacts of SP

as normal users. By increasing SP, the FNR drops, while the FPR increases. The reason is that the colluded users with high contact rate can hardly build very strong social connections with each other as $SP > 0.3$.

In summary, LSMSD performs better than SMSD since the Level-3 attackers can be detected by semi-supervised learning with HMM, which balances the training overhead and detection accuracy. Having the appropriate parameters, e.g., $TH = 100$, $SP = 0.3$, the four levels of Sybil attackers could be detected.

4.6 Related Works

In this section, we introduce some exiting works of Sybil detection. We categorize them into four types, i.e., social network based Sybil detection, social community based Sybil detection, behavior classification based Sybil detection and mobile Sybil detection.

4.6.1 Social Network Based Sybil Detection

Social network based Sybil detection (SNSD) relies on the concept of "social network" from sociology theory [80], which is a social structure linking social relationships among nodes in the social graph. In this section, the term "social network" indicates the user's

social graph and structure, which can reflect user's social relationships and the social trustworthiness [115, 116] among users. Leveraging the "social network" structure, Yu et al. [105] propose an SNSD scheme, named SybilGuard, by using random walk algorithm for detection [117, 118]. Before the explanation of the detailed SybilGuard, we introduce an assumption as follows.

Assumption 1: Although the Sybil nodes can tightly connect with other Sybil ones, the number of social connections among Sybil nodes and honest ones is limited.

SybilGuard relies on Assumption 1, and each node detects the Sybil one in a distributed manner. Specifically, a node with degree R generates totally R random routes starting from itself along its edges with a fixed length L. If a route reaches a known honest node, it is verified as honest by this known honest node. Particularly, a Sybil node S may be accepted as a verified one (i.e., the route from S to H is called verifier) if one of the routes from S reaches the known honest node V. Given a threshold $T \leqslant R$, S is accepted as an honest node when more than T routes from S are verified. According to Assumption 1, the limited number of attack edges makes sure that the number of verifiers cannot be greater than T, where T is properly selected. If there are totally X attack edges, the number of Sybil groups is bounded by X. [119] proves that $T = \Theta(\sqrt{n}\log n)$ is sufficiently large for the honest nodes passing the random walk detection. In addition, security schemes [110] are adopted to ensure the authenticity of the nodes and the routes. Every pair of directly connected nodes (or one-hop neighbors) negotiate a shared key on the connecting edge. Message authentication code (MAC) can be adopted for each node to verify the other one. Furthermore, every generated random route should be registered with an unforgeable token (or witness table) containing all L nodes on the route such that Sybil attackers cannot deny the connections and forge the route information.

The correctness of SybilGuard relies on the fast-mixing property of the social graph [110]. The mixing time t of a social graph indicates how fast the ending point of random walk algorithm achieves the stationary distribution. If the ending point distribution is independent on the starting point as $L \longrightarrow \infty$, it is the stationary distribution [105]. If the mixing time is $\Theta(t)$, the graph is fast mixing. When a random walk with the length $L = \Theta(\sqrt{n}\log n)$, there exist $\Theta(\sqrt{n})$ samples independent on the starting point. The probability that a Sybil node is accepted/verified as honest by the known honest node (i.e., both the Sybil node and the honest one select the same attack edge in the random route) follows the Birthday Paradox [120]. The collision probability is

$$\mathsf{Prob}(\text{Collision}) = 1 - \left(1 - \frac{1}{\sqrt{m}}\right)^{\sqrt{m}}. \tag{4.10}$$

Therefore, SybilGuard has a high probability to detect SA-1 (presented in Section 2) based on random walk.

To improve the detection accuracy of SybilGuard and guarantee the near-optimal, another SNSD scheme, named SybilLimit, [106] is proposed. SybilLimit enables each node to generate $R = \Theta(\sqrt{m})$ random routes with length $L = \Theta(\log n)$. Similar to SybilGuard, the Sybil or honest nodes are labeled by using random walk algorithm [121]. Different from SybilGuard, SybilLimit leverages the intersections on edges instead of vertex (node), and performs short random routes with multiple independent instances of random walk. SybilLimit accepts $O(\log n)$ Sybil nodes per attack edge. This number in SybilGuard is $O(\sqrt{n}\log n)$ [106, 122]. Note that both SybilGuard and SybilLimit rely on Assumption 1.

To understand the characteristics of social structure, Alvisi et al. [123] study several structural properties of social graphs, such as small world property [80], popularity distribution [124], clustering coefficient [125] and conductance [126]. Popularity distribution among the nodes follows a power-law or lognormal distribution. Small world property indicates that the distance between any two nodes is small. Clustering coefficient is a parameter that reflects the closeness of nodes within a social network. The conductance $C(S)$ reflects the mixing time, which indicates the minimum length of a random walk. $C(S) = \frac{S_{out}}{S_{in}}$, where S_{out} denotes the number of edges that are out from S and S_{in} denotes the number of edges within S. According to [123], the conductance (related to the mixing time of a random walk) is more resilient in Sybil detection compared with other characteristics. The mixing time is high, when the conductance is low. [123] also proves that for the first three properties, the number of edges that Sybil attackers need to generate to launch Sybil attacks is 0 or 1, while this number for the property of conductance is $\frac{C(S)m}{\log(C(S))}$. Sybil attackers have to consume more resources to compete with the conductance based Sybil detection schemes. The effectiveness of SybilLimit [106] can be validated since SybilLimit adopts conductance to detect Sybil nodes. In addition, a concept of *perfect attack* is introduced to explain an undetectable attack that draws some honest nodes in the social network into Sybil region, without any impact on the whole social network. In other words, when a Sybil node joins the social network and sets up many connections with the honest nodes, it is not easy to detect such an attacker as well. The attack edge is a metric to evaluate the attacker's capability to launch a perfect attack. To resist the strong Sybil attacks, [123] exploits conductance to build a white-list for honest users and proposes a so-called SoK scheme. This white-list contains a set of nodes ranked associated with these honest users' trustworthiness. The SoK is more robust compared with other SNSD schemes, such as SybilGuard and SybilLimit.

There are also several promising SNSD schemes. For example, Cao et al. [127] propose SybilRank, which utilizes a centralized online server to rank nodes according to their

Table 4.2: Comparison on Social Graph Based Sybil Detection

Sybil defense scheme	Preliminaries	Social graph	Centralized	Trustworthiness
SybilGuard and SybilLimit	Random walk	Undirected	×	×
SumUp	Adaptive max flow	Undirected	√	Credit network
Gatekeeper	Random walk	Undirected		Trust
SybilDefender	Community detection	Directed	√	Trust
SybilShield	Community detection	Undirected	×	Trust
VoteTrust	Community detection	Directed	×	Asymmetric trust

perceived likelihood of being Sybils. The goal of SybilRank is to achieve the scalability of the Sybil detection in a large scale OSN and reduce the computation overhead. By exploiting a probabilistic model of honest node's social network, Danezis el al. [128] propose a Bayesian inference scheme to divide the whole social graph into Sybil and honest regions. In addition, the principle of privilege attenuation [129] is developed for SNSD to prevent malicious Sybil attackers from arbitrarily adding or removing edges in the social graph without employing social engineering, especially for collusion attack [130]. To further improve SybilLimit, Tran et al. [131] propose a "Gatekeeper" scheme to optimize the case of $O(1)$ attack edges and guarantee only $O(1)$ Sybil identities. Gatekeeper is integrated with a multi-source ticket distribution algorithm for node admission control.

Exploiting trust to build social graph becomes a state-of-the-art idea of SNSD. Cao et al. [132] propose SybilFence, leveraging users' negative feedbacks on Sybil attackers and adjusting the weight of each edge in social graph. Specifically, if a user u_i receives negative comments from others, u_i's edge weights are correspondingly reduced. With the directed social graph, Sybil attackers can be well detected. Tran et al. [55] adopt credit network [115, 133] and propose SumUp to solve the vote aggregation problem in an online rating system. SumUp leverages online user's voting history to restrict Sybil attacker's voting capability if this attacker continuously misbehaves. In SumUp, a trusted node computes a set of max-flow paths on the trusted graph and then aggregates the votes. It allows the votes from the trusted users to be effectively aggregated, while limits the

votes from untrusted users. Canal [134] is similar to SumUp. With a credit payment mechanism in a large scale network, Canal enhances the establishment of social graph and is compatible to the existing SNSD.users . In [135], Mohaisen et al. form a trust-based social graph based on the observation that nodes trust themselves more than they trust others. They also observe that the trustworthiness of other nodes is not uniformly equal. Differential trust in the social graph is developed to filter weak trust edges and model trustworthiness by biasing random walks. Delaviz et al. [136] propose a trust and credit based Sybil detection scheme, named SybilRes. SybilRes utilizes a local subjective weighted directed graph to indicate user's data transfer activities. When a user u_i uploads data, the edge weight on the path from u_i to the downloaders is reduced. To maintain the edge weight of honest users, after downloading, the downloaders increases the weights of the edges on the paths from the uploader u_i to itself. Then, Sybil users could be detected by using the sophisticated SNSD. Unlike the basic SNSD [105, 106], these trust based SNSD schemes [136, 135] leverage trustworthiness to build a directed social graph rather than the original undirected social graph for random walk Sybil detection. Since this enhancement relies on a practical assumption that the honest nodes would not provide high trust on the unknown (or Sybil) nodes, the attack edges could be filtered to guarantee the SNSD accuracy. Therefore, the credit and trustworthiness enhance the traditional social graph and restrict Sybil attackers to build connections with normal users such that the detection accuracy is improved.

4.6.2 Social Community Based Sybil Detection

Social community based Sybil detection (SCSD) develops social community detection algorithms to facilitate Sybil detection. SCSD explores social community detection to facilitate Sybil detection. The possibility of using social community detection algorithms to detect SA-1 is validated in [137]. Viswanath et al. [137] first analyze the SNSD schemes and summarizes them into a ranking problem. Recall that the SNSD schemes usually partition Sybil nodes and honest ones into two parts: Sybil region and honest region. It is possible to formulate as a graph partitioning problem. For SNSD schemes, each unknown node is ranked according to its social connections with the known trusted nodes. By selecting different parameters (i.e., thresholds), the social graph can be divided into two partitions. These parameters determine the boundary of the partition, i.e., "cutoff". The ranking of nodes is towards the direction of reduced conductance. In other words, the nodes tightly connected with the known trusted ones (e.g., lower conductance) would score higher in the ranking. Furthermore, the ranking algorithms significantly impact on the ranking results and the Sybil partition. Meanwhile, another problem comes out. If a node weakly

connects with the current known trusted nodes, it is more likely to be detected as a Sybil node no matter how tightly it connects with other unknown trusted nodes. In other words, when there are multiple social communities in the graph, it is inefficient and ineffective to detect Sybil nodes only through social network partition. Therefore, leveraging community detection to detect Sybil nodes becomes promising and enhances the traditional Sybil detections.

SybilDefender [138] is a typical SCSD scheme. It performs a limited number of random walks for Sybil identification and community detection. Sybil identification can detect whether a node is Sybil or not, similar to the existing SNSD schemes. Then, a community detection algorithm is adopted to detect the neighboring Sybil nodes around the detected Sybil nodes. Moreover, an efficient combination of Sybil identification and community detection facilitates SybilDefender to consume few computation overheads. Due to the observation that a portion of relationships among OSN users are untrusted [124], SybilDefender develops a mechanism to limit the number of attack edges. This attack edge limiting mechanism enables users to rate their friend's relationships as "Friend" or "Stranger". The attack edges could be further removed since Sybil attackers are probably "Stranger" from the view of normal users. Note that SybilShield relies on Assumption 1.

By using multi-community social network structure, Shi et al. [139] propose SybilShield, an agent-aided SCSD scheme. SybilShield also leverages trust relationships among users to form the social graph. Since two honest nodes from two different social communities may not tightly connect with each other, SybilShield exploits the agents and ensures the honest nodes tightly connect with other honest ones. Similar to SybilGuard (the first random walk based Sybil detection) [139], some agents of a verifier are selected to run a second round of random walk, called agent walk, where the agents traverse all of the verifier's edges to confirm the suspect nodes. SybilShield relies on Assumption 2.

Assumption 2: Sybil nodes cannot tightly connect with honest nodes in the multiple honest communities since honest nodes would not trust Sybil ones. Honest nodes can tightly inter-connect with others in the honest community.

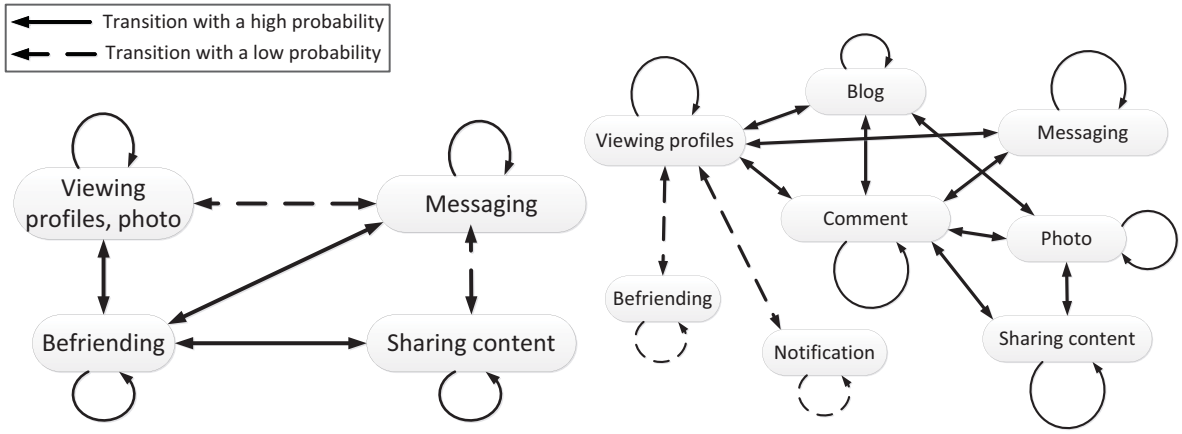
Given Assumption 2, Cai et al. [140] leverage the latent community model and utilize a machine learning algorithm to detect Sybil attacks. According to [140], the tightly interconnected communities are connected more closely than the one loosely connected. Although several communities are compromised by Sybil attackers, the attack communities can be also detected via the transitivity of the latent community model. Based on user's befriending interactions (invite, or accept friends), an interesting SCSD scheme, named VoteTrust [141], builds a friend invitation graph and leverages a trust-based vote assignment as well as global vote aggregation to estimate the probability of a Sybil at-

tacker. VoteTrust integrates the social graph and user's social behaviors (i.e., feedback of accepting or rejecting friend requests in OSNs) to establish a directed graph. It relies on an assumption that the Sybil users cannot receive more than a certain number of friend requests from normal users. The global aggregation of the votes for every node can be used to estimate its global rating. With this two-way (voting and feedback) mechanism in a directed graph, Sybil detection can be more effective compared others schemes. In Table 4.2, we compare the SGSD schemes with respect to preliminary techniques, assumption, decentralized properties, etc. A tendency is to explore trustworthiness to facilitate the Sybil detection to SA-1.

4.6.3 Behavior Classification Based Sybil Detection

Users' behaviors can be used to classify Sybil attackers, i.e., behavior classification based Sybil detection (BCSD). In [107, 57], Sybil users in RenRen, a Chinese OSN, can generate an exponential number of social connections with the normal (or honest) users. Jiang et al. [142] demonstrate that the smarter Sybil attackers rarely establish social connections with other Sybil attackers in RenRen. As a result, the SGSD schemes may not always effectively detect these smarter Sybil attacks once Assumptions 1 and 2 do not hold. Therefore, some novel Sybil detection schemes are desirable and should exploit some promising features of Sybil attackers.

Based on the observation on abnormal OSN behaviors, Wang et al. [57] develop Sybil attackers by comparing OSN users' browsing and clicking habits (as known as "online" habits). According to the data obtained from RenRen [57], the basic OSN activities of users are summarized as follows. 1) *Befriending*: send, accept or reject friend requests; 2) *Photo*: upload photos, tag friends in the pictures, browse photos, and comment on the photos; 3) *Profile*: browse profiles of other users; 4) *Share*: share multimedia contents, including video, photo, audio, text contents and website links; 5) *Messaging*: update status, wall posts, send or receive instant messages; 6) *Blog*: post blogs, browse blogs, and comment on the blogs. According to the statistics, the primary activities of Sybil users are friending (especially, sending friend requests), viewing photos and profiles of others, and sharing contents with others. On the contrary, the normal users spend a large portion of online time to view photo, and perform other activities, like viewing profiles, sending messages, sharing contents with a similar frequency. Both Sybil attackers and normal users share content or send messages at similar frequencies. Note that sharing content or sending messages are the common approaches for Sybil attacks to disseminate spam in OSNs. This observation indicates that the traditional spam detection schemes cannot simply leverage numeric thresholds to resist spam.



(a) State transitions for a Sybil user

(b) State transitions for a normal user

Figure 4.9: Online social networking behaviors and transition probabilities of Sybil attackers and normal users.

As shown in Figure 4.9, the click transitions could be modeled by Markov chain with each state as a click pattern. Normal users usually perform diverse OSN behaviors, and the transitions among states are really complicated. By contrast, the Sybil users are involved in some specific activities in a high frequency. To distinguish the BOSAs, support vector machine (SVM) [143, 144] can be adopted according to the session features, such as average clicks per session, average session length, average inter-arrival time between two clicks, and average sessions every day, and the click features. The preliminary results show that the Sybil detection accuracy is high. In [57], three models (click sequence model, time based model, hybrid model), which can cluster similar click patterns, are built for the behavior classification. According to some specific similarity metrics, the sequence similarity graph can be established. Through graph clustering, the Sybil users can be detected. The SVM based scheme is supervised learning tool which requires a long term training period. To address this issue, an unsupervised learning scheme is proposed, where only a small portion of click patterns of given normal users as "seeds". They color normal clusters which contain a seed sequence; otherwise, the uncolored clusters are Sybil ones.

Assisted by crowdsourcing and social Turing tests, Wang et al. [145] propose a distributed Sybil detection scheme which significantly improves the detection accuracy. For a Sybil attacker, he cannot pass "social Turing test" with different attack strategies. Furthermore, crowdsourcing provides an adaptive platform for normal users (e.g., "turker")

to complete the Sybil profile detection with a reasonable cost. From the experiments in [145], the accuracy of crowdsourcing Sybil detection under the reasonable burden is almost as high as that performed by "experts". Some key factors (e.g., demographic factors, temporal factors and survey fatigue, turker selection, and profile difficulty) that may impact on the crowdsourcing Sybil detection are provided. Obviously, the cost of a crowdsourcing workforce is low. In addition, some BCSD schemes [146] are proposed based on behavior classification. For example, DSybil [146] exploits the heavy-tail distribution of the classical voting behavior from the honest users to detect Sybil identities. In summary, these BCSD schemes can detect Sybil attacks according to the user's behavior learning and classification in different applications.

In reality, smart and strong Sybil attackers penetrate into the social graph and generate plenty of social connections with normal users. It breaks the assumption for the SGSD schemes. If Sybil attackers are familiar with normal user's click patterns or habits, Sybil attackers may mimic normal users such that BCSD schemes may not be effective. However, it is obvious that Sybil attackers have to consume a large portion of time or resources to mimic normal users. Consequently, their attack behaviors are partially limited.

4.6.4 Mobile Sybil Defense

Due to the dynamic mobility of MSNs and the incomplete global social graph information, Sybil detection is quite difficult compared with that in OSNs. In [104], Quercia et al. propose a matching based scheme for Mobile Sybil Detection (MSD). This scheme allows mobile user's to match common communities and label the users from the Sybil community as Sybil attackers. The assumption of [104] is that each user maintains a friend list containing the trusted mobile users, and a foe list with the untrusted users in it. When two users meet with each other, they fist match their communities by using profile matching [94]. The users outside the trusted communities are identified as Sybil attackers. In addition, Chang et al. [147] propose another MSD scheme based on the assumption that Sybil attackers and normal users are from different communities. Under this assumption, community matching is conducted when users are encountered for detections. Leveraging friendship is an effective methodology to identify Sybil attackers. However, this type of friend relationship based MSD (FR-MSD) schemes need mobile users to maintain the trusted community information in advance.

Cryptography is another promising tool in MSD, and can restrict Sybil attacker's malicious behaviors to some extent. We introduce several cryptography based MSD (crypto-MSD) schemes as follows. When Sybil attacks are launched in VANET, a challenging issue

of detecting them is vehicle's dynamic mobility, making it increasingly difficult to tie an attacker to the location. To address this issue, Lin [109] proposes a Local Sybil Resistance scheme (LSR) based on group signature [148] to detect local Sybil attackers and mitigate zero-day Sybil vulnerability in sparse VANET. As the vehicle users hardly detect local Sybil attackers, LSR [109] enables a user u_i to generate event signatures, where the event is posted by u_i. According to the features of group signature, if a user signs on the same event for multiple times (i.e., posting this event for several times), these signatures would be linked and invalid. Users are able to detect these local Sybil attackers. In [109], the delay of Sybil reports from vehicle users is also analyzed. Two-layer and multi-layer reporting are proposed to track the Sybil attacker's real identity for TA's revocation. In addition, some costly resources are exploited for Sybil detection, which aim to limit Sybil attacker's capabilities. Secure hardware [149] is also utilized to validate every user's authenticity. Sybil attackers can only authenticate themselves with a limited number of times. Although this secure hardware based Sybil detection scheme may effectively resist Sybil attacks, its high cost hinders the widely usage. It is usually applied when the highest security level is required. Reddy et al. [56] propose an identity fee based Sybil defense scheme by increasing the cost of identity maintenance. Sybil attackers have to spend more fees/resources to launch attacks. Similarly, a resource testing scheme [150] detects the overloaded users as Sybil attackers. The resource testing relies on the observation that the each user or attacker works on a single or limited number of machines/devices. If a Sybil attacker exists, it might consume the dramatic amount of resources (e.g., computation, communication, storage, and network bandwidths) to maintain the created fake identities. In [151], Li et al. propose an admission and retainment control mechanism to enforce nodes to periodically solve computational puzzles. Although these dedicated resources can support legitimate nodes, Sybil attackers are not able to obtain sufficient recourses to launch attacks. The attacker's capabilities are limited to some extent. These Sybil detection schemes provide some challenges, such as hardware, device resource, and reputation, to limit the Sybil attacker's behaviors.

In MSNs, pseudonym techniques are widely applied. Indeed, pseudonyms protect legitimate user's real identity from being identified and linked. However, pseudonymous identities hinders MSD since the detection schemes hardly trace the Sybil attacker's identity given only his pseudonyms. Similarly, in [152, 153], a malicious user pretending to be other vehicles can be detected in a distributed way through passive overhearing by a set of fixed nodes (i.e., road-side boxes). Such a Sybil detection does not disclose any vehicle's identity during the detection. Vehicle users' privacy can be preserved at the same time. In [154], Triki et al. utilize the embedded RFID tags on the vehicles and the short lifetime certificates from RSUs to verify user's authenticity. Some observers (e.g., RSUs,

or vehicles) are involved in monitoring the sensitive events to mitigate the false negative detection. Moreover, vehicles change their identities when they switch to another RSU's area. The unlinkability and privacy can be achieved.

Sybil attackers could compromise the popular service review applications, especially in User-User and User-LS domains of MSNs [108, 54]. Usually, MSN users query the special offers of products, services and social activities, and browse the reviews or service evaluations from experienced users. Alternatively, the LS can gather the users' comments and post them to the nearby users in User-LS domain of MSNs. No trusted authority is always available to maintain trustworthiness between LSs and users. Sybil attackers in the local area may forge some positive reviews, delete or modify the negative ones. They may maliciously manipulate the system and degrade the quality of MSN applications. To detect Sybil attacks during service evaluation, Liang et al. [108] exploit trustworthiness and propose a Trustworthiness Service Evaluation scheme (TSE) to gurantee the legitimate service review submissions and limit Sybil attackers' capabilities. In the TSE, local service providers (LSPs) generate many tokens to synchronize users' review submissions. A user u_i ties his reviews with signatures to only one token after receiving a token from either LSPs or other users who have similar profiles or preferences. The trust relationships are established based on users' similar profiles and preferences in a local area. Afterwards, these tokens are circulated among mobile users for cooperative review submissions. A time stamp is included in the review signature to prevent any user from modifying and deleting the submitted reviews. In addition, every user adopts pseudonym when submitting reviews. All pseudonyms for the reviews in the same token are stored in a list for the traceability purpose [155]. If u_i submits multiple reviews with multiple pseudonyms, both LSP and other users can easily identify it according to the features of group signature. Moreover, u_i's real identity can be linked given the revealed multiple pseudonyms that u_i uses. After publishing a token, the LSP cannot omit this token once some reviews are negative to the LSPs. In each token the length of the review chain can bound the LSP's modification capability. For example, the LSP has to be stronger to modify the existing review chain with a longer review chain. With different token structures, such as ring, chain, tree, it is difficult for SA-3 to modify the posted reviews. It is because the established structure would be destroyed if any modification is made on this structure. Besides the basic cryptography solutions, in [108], if a user generates a massive number of reviews with the same pseudonym in a short period, i.e., one time slot, other users can easily detect his behavior.

Some other mobile network features, such as channel characteristics [156, 157] and mobility features, are also investigated to differentiate Sybil attacks and normal users. In [158], channel characteristics are investigated, where the spatial variability of radio channels is typical in indoor and urban environments with rich scattering. [158] also

develop an enhanced physical-layer authentication. The integration of channel features and authentication jointly detects Sybil attacks. This proposed scheme is feasible featured by efficient channel estimations. In addition, the received signal strength (RSS) [159, 160] is utilized for Sybil detection in a static wireless network, e.g., sensing domain as discussed in Section 2. If a node receives the packets with similar RSS for many times, the sender is likely a Sybil attacker [161]. Several other MSD schemes leverage mobile network features to defend Sybil attacks. In [162], Geutte el al. estimate the amount of cheated nodes to measure the success rate of Sybil attacks. They evaluate the impact of transmission power tuning from senders, and analyze the impact of bi-directional antenna over omni-directional antenna for the receiver. By comparing the transmission signal differences, they quantify the effects of different security assumptions on Sybil attackers and the impact of antennas on the Sybil detection accuracy. In [163], Yu et al. analyze vehicles' communication signal strength distribution and exploit a statistical method to cooperatively identify the location where a vehicle comes from. Since the neighbors cooperatively measure the signal strength of the specific vehicle, the location estimation accuracy can be significantly improved. Abbas et al. [164] propose a lightweight RSS-based Sybil detection scheme in mobile ad hoc network, without centralized authority and dedicated hardware (e.g., directional antenna or GPS). This lightweight detection scheme relies on the node mobility, and sets the threshold to differentiate the node's moving speed. If any node moves much faster than the pre-set threshold, it may be Sybil attackers. In summary, by investigating normal user's and Sybil attacker's behaviors related to channel conditions, Sybil attackers can be identified.

With the consideration of user's mobility, Piro et al. [165] observe that in Mobile Ad hoc NETwork (MANET), the Sybil identities related to a single Sybil attacker are bound to a single physical node. In other words, an effective detector should be able to find that many Sybil identities move together. By monitoring the user's motility, Sybil identities can be detected. In [166], Mutaz et al. leverage the mobility characteristics of vehicle platoon to detect the Sybil attacks in VANETs. Park et al. [167] investigate the mobility of vehicle and detect Sybil attackers based on the fact that the two vehicles rarely pass multiple roadside units (RSUs) always at the same time. Correlating the vehicles and RSUs in both temporal and spatial domains, Sybil attackers can be detected. Defending Sybil attackers through investigating the system features is a promising approach where the challenge is how to obtain the sufficient knowledge or features.

In Table 4.3, we summarize the existing Sybil defense schemes with respect to some design principles. Sybil defense should leverage different features to classify, detect, and resist Sybil attacks towards different scenarios and networks. In summary, some existing mobile Sybil detection schemes either rely on the pre-defined communities among users,

Table 4.3: Sybil Detection: A Comparison

Sybil defense scheme	The type of Sybil attacks	Preliminary technique	Base or Assumption	Decentralization
SNSD	SA-1	Social graph partition, Random walk	Assumption 1	Centralized
SCSD	SA-1	Community detection	Assumption 2	Centralized and decentralized
BCSD	SA-2	Behavior classification	Behavior difference	Centralized and decentralized
FR-MSD	SA-3	Community detection, or profile matching	Trusted community features	Decentralized
Feature-MSD	SA-3	Channel estimation, feature classification	Wireless channel characteristics, mobility features	Decentralized
Cypto-MSD	SA-3	Cryptography	Security of cryptography	Decentralized

or adopt cryptography techniques to restrict Sybil attackers. However, Sybil attackers may act similarly as normal users to disrupt these traditional mobile Sybil detections. Furthermore, some online Sybil detection schemes relying on centralized authority cannot be directly applied in the mobile network. To this end, we study the relation between mobile user's contacts and pseudonym changing behaviors and propose the mobile Sybil detection scheme balancing the trade-off between the detection accuracy and overhead.

4.7 Summary

In this chapter, we have proposed a social-based mobile Sybil detection scheme to detect four levels of Sybil attackers with different attacking capabilities. We have investigated mobile user's pseudonym changing behaviors associated with their social contact to differentiate Sybil attackers from normal users. The security analysis demonstrates the effectiveness of the SMSD in terms of detecting four levels of Sybil attacks. The extensive trace based simulation validates the detection accuracy of the SMSD. The proposed SMSD scheme is a novel paradigm of detecting Sybil attacks in MSNs, which takes the advantages of powerful storage and computing capabilities from the cloud server. It also initiates a trend to distinguish Sybil attackers via mobile user's social behaviors and mobility.

Chapter 5

Exploiting Social Network To Enhance Infection Analysis With Privacy Preservation

5.1 Introduction

Infectious disease, such as flu, Ebola and some acute respiratory infections, cause people infected by pathogenic microorganisms (bacteria, viruses, fungi, parasites, etc.), and spread from human to human within a short period. According to the health report in 2013, the population of infected Canadians with these highly contagious diseases rises over $200,000$, where more than $8,000$ infected patients die as a result [168]. The outbreaks of these infectious diseases [169] usually occur when the infected patients cough and sneeze around non-infected people [170]. A recent study [171] observes that people having strong socialties and long-lasting (or frequent) contacts is likely to spread infectious diseases from the sociology and biomedical perspective. One of traditional infection prevention approaches to prevent the human-to-human spread of infectious diseases [168] is to isolate susceptible patients from the public for a certain period depending on the latent time period of the diseases. These susceptible patients might be families living in the same house, students studying in the same classroom, colleagues in the same company, etc. For example, during the outbreak season of Ebola, people traveling from the infected region or having frequent contacts with infected patients are supposed to be isolated, e.g., staying at home or in a special area of hospital, for two or three weeks to make sure that they are not infected before coming back to the normal life.

However, this traditional infection prevention approach consumes massive health expense and labor costs, results in the isolated patient's economic loss and even anxiety or panic of the whole society. To resolve this type of public health crisis, a promising e-healthcare system [64] emerges to continuously monitor users' real-time health parameters, such as temperature, heart rate and electrocardiogram (ECG), which are formatted in image, audio and text. These health multimedia data are collected by a server to analyze abnormal phenomena and provide supporting information for doctor's diagnosis. Although such an e-healthcare system is helpful to analyze user's health condition, i.e., whether a user is already infected or not, it lacks sufficient social information to infer the spread of infectious diseases, i.e., whether the user has a high probability to get infected from others.

In recent years, multimedia techniques have been used to mine social data from various applications [172]. For example, the built-in face-tagging function of Facebook application can identify user's face in pictures and infer if certain users have close social relationships; Wechat friend discovery program can find users in the physical proximity and record social interactions; speech recognition can help to detect if some people cough or sneeze. The fusion of these social multimedia data associated with the monitored health data can provide a novel paradigm to enhance infection analysis. Suppose a junior school student Bob is continuously monitored from both health and social perspectives during the outbreak of infectious disease. Once Bob's immunity strength goes very low and he frequently contacts an infected student, he may be inferred as a susceptible patient in the early stage. The health and social multimedia data are usually collected and processed by multiple independent service providers, such as health institution and social networking service provider (e.g., Facebook and Wechat), respectively. The collaboration of these service providers becomes essential to enable data sharing and processing [173], especially when the volume of continuously monitored data keeps increasing. Incorporating health cloud server collecting users' health parameters and social cloud server, which is maintained by social network service provider to collect users' social networking data including social contact and relationships between users, we envision that the infection analysis can be enhanced.

Meanwhile, users' health and social data, such as infection status and social contact, are privacy-sensitive [26], and many users are not willing to excessively reveal this private information to the untrusted or unauthorized entities [174]. If the health data and social data are sent to cloud servers in clear text, the untrusted cloud servers may track users' health condition, identity, profiles, contact and social activities, resulting in severe privacy violations, especially for the infected or susceptible patients during the outbreak of infectious diseases. To preserve data privacy, users could encrypt their data and send the ciphertexts to cloud servers [175]. However, this approach may limit the data pro-

cessing capability of cloud servers [176] and even disable the infection analysis. Therefore, it is challenging to enable the infection analysis and preserve user's privacy at the same time. In addition, social networking data contain some sensitive information of infected and susceptible patients, such as identity and contact details, which may be inferred by the social cloud server when these data are shared to other entities for further health analysis. For example, if the hospital or public health agency queries an infected patient's data on the social cloud server, the social cloud server may infer that the queried user is infected. Meanwhile, the hospital without the user's authorization should not be able to query non-infected user's social networking data. Without sufficient privacy protections, users may not want to share their social and health data to the untrusted cloud servers for infection analysis. Therefore, it is still challenging to address the aforementioned issues when exploiting social networking data to enhance infection analysis.

In this chapter, we propose a Privacy-preserving Infection Analysis approach (PIA) to infer human-to-human infection spread by integrating social networking data with health data. This approach employs a privacy-preserving data query method based on conditional oblivious transfer to enable data sharing among different entities and a privacy-preserving classification-based infection analysis method to enable the cloud servers to infer infection spread and achieve health data privacy. The main contributions of this chapter are fourfold.

• Firstly, we analyze the spread process of infectious disease with the consideration of user's social contact and health condition. We exploit several key factors of infection, including immunity strength of the susceptible user, infectivity of the infected patient, their contact duration and contact type. We also utilize naive Bayesian classification method to enhance infection analysis with the collaboration of social and health cloud servers.

• Secondly, we propose a privacy-preserving data query method (PPDQ) based on conditional oblivious transfer to allow the authorized entity (i.e., hospital) to access the infected patient's social networking data from the social cloud server, but not allow the social cloud server to access and infer any data including patient's identity. Furthermore, this method enables users to grant authorization to hospital, which cannot query any data without user's authorization.

• Thirdly, we propose a privacy-preserving classification-based infection analysis method (PCIA) to prevent user's private social and health data from disclosing to the untrusted health cloud server. The PCIA enables users to encrypt raw data based on homomorphic encryption and send ciphertexts to the cloud server. Then, the health cloud server can infer infection spread during human-to-human contact without learning any user's private information.

• Finally, privacy analysis shows that the proposed approach preserves the privacy of user's health data and social networking data, and achieves patient's identity privacy during the query. Furthermore, we conduct the extensive simulation to demonstrate that the PIA exploits the social networking data and adjusts to effectively analyze infection spread with acceptable computational overhead.

The remainder of the chapter is organized as follows. We present the system model and design goals in Section 5.2. Then, we propose the PIA with details in Section 5.3. The privacy properties are analyzed in Section 5.4, and the performance is evaluated in Section 5.5, respectively. We also review the related works in Section 5.6. Finally, we conclude the chapter in Section 5.7.

5.2 System Model and Design Goals

In this section, we propose the infection analysis system model and identify the design goals, respectively.

5.2.1 System Model

The proposed infection analysis system consists of five entities: trusted authority (TA), users (i.e., data owners), hospital, social cloud server (SC) and health cloud server (HC) as shown in Fig. 5.1. The system is divided into health domain and social domain according to different types of collected data. Users, HC and hospital have operations on health data in the health domain, while users, SC and hospital (as a query requestor) are involved in the social domain. The details of each entity in the PIA are presented as follows.

• **Trusted Authority (TA)** bootstraps the system, processes user's registration, and generates the certificates for legal user's key generation. Afterwards, TA is not involved in networking and users' interactions.

• **Users** first register to the TA and generate valid keys in the initialization phase. They measure their health parameters via wearable devices and periodically send health data to the health cloud server. When user U_i and U_j have contacts with each other, their smartphones record the contact information, such as identity, duration and social relationships, which are sent to the social cloud server.

• **Health Cloud (HC)** has powerful computational and storage capabilities to perform the complicated and time-consuming operations on health data. HC receives health data from users, and training data from medical institutions for analysis.

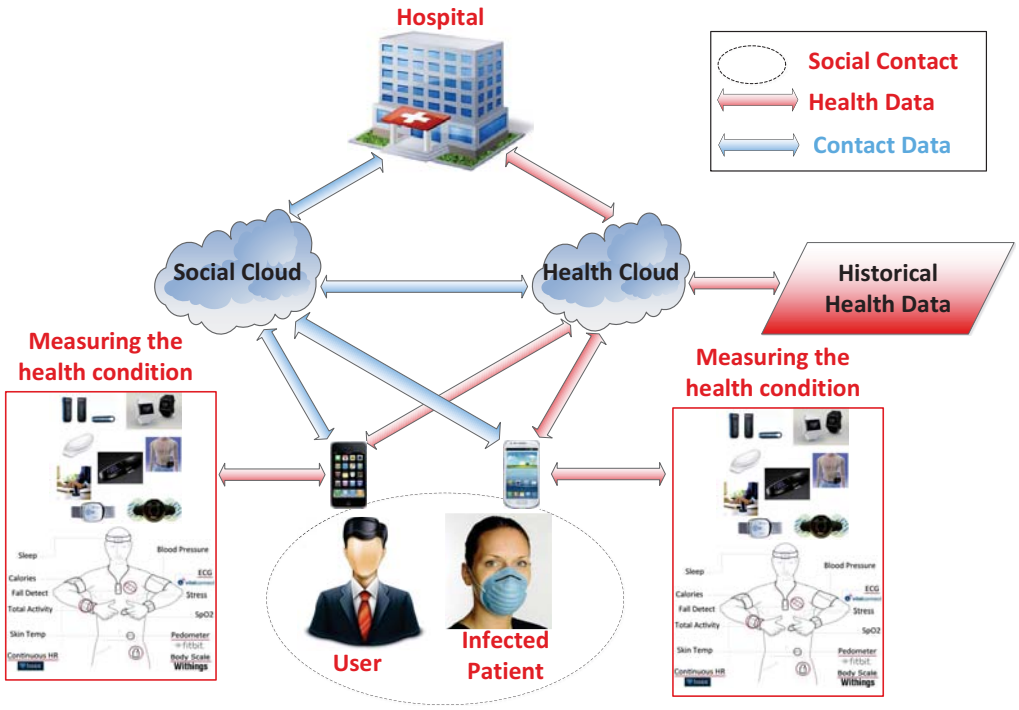


Figure 5.1: Infection analysis system

• **Social Cloud (SC)** is the cloud server dedicated for social networking data storage and processing, which is similar to HC. SC only operates in social domain.

• **Hospital (*H*)** is the entity to analyze user's infection status. If H diagnoses a user U_i as infected patient, H queries U_i's social networking data from SC. Having U_i's social networking data, H performs infection analysis with HC to determine whether U_i's encountered users are susceptible or not, and informs users the analysis results.

5.2.2 Security Model

HC and SC are honest-but-curious entities in the system, i.e., they honestly follow the protocols but are curious about users and other entities' private information. H is trusted by users and has the authorization from users to access their health data stored on HC. However, H is semi-trusted in social domain. If H diagnoses a user U_i as infected patient, U_i grants authorization to H and allows H to access U_i's social data from SC. Otherwise,

H is an honest-but-curious entity in social domain and not allowed to access any user's social data in SC except without the authorization.

5.2.3 Privacy Requirements and Design Goals

Under the honest-but-curious model, user's personal information included social data and health data should be kept confidential towards untrusted and unauthorized entities. The privacy requirements are identified as follows.

(1) *Health Data Privacy*

User's health data should be prevented from disclosing to other unauthorized entities, such as HC, SC and any unauthorized user. Particularly, the infected patient's infectivity and other health data are highly privacy sensitive and should not be disclosed to the cloud servers and other users. Moreover, the historical data (training data set) from medical institutions should be also encrypted in the ciphertext during the classification by HC.

(2) *Social Data Privacy*

User's social contact data are also part of user privacy. The encountered user's information, such as identities, contact type and duration, should be invisible to SC and other users when the data are stored in SC. Without user's authorization, H should not be able to access this user's social networking data as well.

(3) *Privacy of Susceptible User and Infected Patient*

Some users may be susceptible to be infected. Before diagnosis, their information, such as identities and health status, should be also protected against SC's inferring. In addition, susceptible user's risk analysis result) should be invisible to HC, SC and any user except authorized hospital. This risk analysis result reflects user's infection status, which is highly sensitive to him.

The proposed system should achieve privacy requirements and computational efficiency simultaneously. On one hand, the proposed system should be able to protect user's social data and health data from disclosing to (or inferring by) untrusted entities. On the other hand, it should take a reasonable computational and communication overheads, which would prolong the system's lifetime and improve user's experiences.

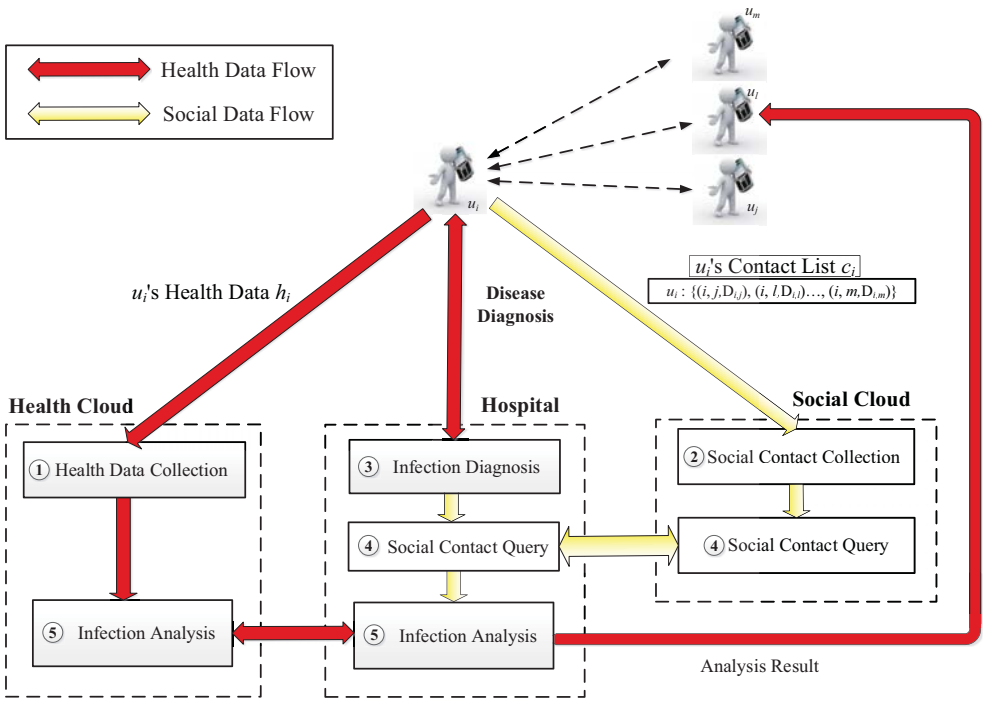


Figure 5.2: Overview of privacy-preserving infection analysis

5.3 Privacy-preserving Infection Analysis Approach

In this section, we propose the PIA, a novel privacy-preserving infection analysis approach. We first present an analytic model on spread of infectious disease and utilize naive Bayesian classification to infer the infection spread. Then, we propose a privacy-preserving data query method and a privacy-preserving classification-based infection analysis method to achieve the design goals.

5.3.1 Overview of PIA

The PIA adopts naive Bayesian classifier to detect the infected and susceptible users based on the training data set. It consists of health data collection, social data collection, infection diagnosis, social contact query and infection analysis as shown in Fig. 5.2. The infection diagnosis is performed by doctors in hospital, which is not discussed in this section. In the

other four components, user's privacy is protected from disclosing to untrusted entities .

(1) *Health Data Collection*

Users first adopt on-body sensors and wearable devices to measure their health conditions, including temperature, heart rate, sleep quality and ECG. Before sending the data to HC, users encrypt the measured health data into ciphertexts since these health data are highly privacy-sensitive to users. Finally, the health data are stored in HC.

(2) *Social Data Collection*

When users contact each other, their smartphones can record the detailed contact information, including identity, duration, contact type and social-tie. These social data are timely uploaded to SC for storage. The included information is highly private-sensitive and should be invisible to unauthorized entities, e.g., SC.

(3) *Privacy-preserving Data Query*

The hospital H diagnoses infected disease of patients and determines the infectivity IF as shown in Fig. 5.3 according to [177]. Then, H informs the infected patients with the diagnosis results. After the diagnosis, H performs a social contact query to SC with the authorization from users. H sends the query request, including infected user u_j's identity associated with the queried contact duration and some other social information, to SC in the ciphertext. SC performs operations on the query request without knowing the query result and feedbacks it to the hospital.

(4) *Privacy-preserving Classification-based Infection Analysis*

HC and H compute the contacted user U_i's infection status based on U_i's immunity (measured by U_i), u_j's infectivity, contact duration, contact type, social-tie, etc. in a privacy-preserving way. Finally, H sends the analysis results to U_i as the guidelines to treat the potential disease.

5.3.2 Analysis of Infectious Disease Spread

In this section, we propose the analytic model of infectious disease spread. Many infectious diseases, such as H1N1 and measles, can be spread human-to-human via infected droplets during sneezing or coughing, as well as contaminated surfaces and hands. For instance, in a conference environment, Alice has flu and attends the conference where crowd of people are in the same area/room. Alice has many contact with other people such that the flu is likely spread to the contacted people if they do not have sufficient antibody against this type of flu.

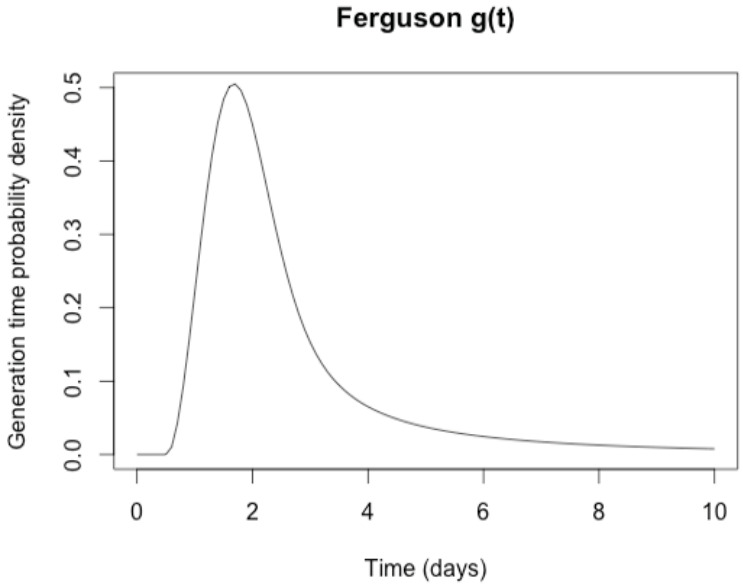


Figure 5.3: Infectious disease spread trend

(1) *Factors on Infectious Disease Spread*

The infectious disease spread process between an infected user u_a^* and a normal user u_b may be impacted by several factors, i.e., u_a^*'s status (e.g., spread strength), contact duration between u_a^* and u_b, u_b's health condition (e.g., immunity strength).

• ***Infectivity from infected user***:

We characterize infected user u_a^*'s status in terms of his infectiousness $\mathsf{IF}_{u_a^*}$ as a function of the time since u_a^* is a case [178]. Here, $u_a^* \in \mathbb{IU}$ which is a set of infected users. The infectiousness IF is impacted by the time t_s of symptom onset when u_a^* is a case, the time t_i of infection when u_a^* is a case, and a vector $\mathbf{x}_a^*$ of u_a^*'s personal health status measured by wearable devices (e.g., temperature and blood oxygen saturation) and measurements from hospital including white blood cell and red blood cell content, hemoglobin, etc. Furthermore, symptoms indicate the infectiousness to some extent. For example, users have acute respiratory diseases, such as ARI, may have at least two symptoms among fever, cough, sore throat, and runny nose. The infectiousness is proportional to the strength of these symptoms, $SP_{u_a^*}$. Note that the infectiousness is not proportional to time t. As shown in Fig. 5.3, the generation time of infectious disease is relatively short (only 2 days) [177]. The infectivity (shown as probability density) keeps increasing at the beginning and decreases after the infectious period.

• ***Contact***:

According to a recent study [171], sitting next to an infected user or being his playmate in a short contact period (e.g., contact lasting minutes with the infected user) is not expected to considerably increase the risk of infection. However, a long period contact with the infected user, such as the structuring of school into classes and grades, strongly affected spread with an increasing infection risk. The contact duration between the infected user u_a^* and normal user u_b can be denoted as $\mathsf{D}_{a,b}$.

Another important characteristics of users' contact is the type of contact $\mathsf{TC}_{a,b}$, which includes 1 = "household" (users living in the same house), 2 = "office" (users in the same office (or classroom in school)), 3 = "department" (uses in the same department (or grade in school)), 4 = "company" (users in the same company (or school)), 5 = "community" (users from the same community, club or social group), etc. The contact type also reflects the social relationship and social-tie strength between contacted users.

The contact information can be bi-directionally captured by wearable devices and smartphones in various ways. User's smartphones can start a Bluetooth discovery program to find the nearby users within a certain range, e.g., 5m or 2m. The contact duration is easy to record by smartphones. Alternatively, GPS and WiFi techniques are possible to measure the location or distance between the contacted users. But this approach is inevitable to face the problem of localization accuracy, especially in the indoor environment or when the accuracy requirement is within meters. Contact type can be captured through the contacted user's social network profiles, such as Facebook, Twitter and WeChat. As the contact information is accumulated at the user side, SC is adopted to help users to store their contact information.

• ***Normal user's health condition***:

When a normal user u_b has contact with the infected user u_a^*, u_b's health condition, especially immunity strength, and some other parameters including sleep of quality and physical strength, also has impact on disease spread. Let IS_b be the immunity strength of u_b against infectious disease. For simplicity, we consider only one type of infectious disease in the following of this chapter.

(2) *Spread Model of infectious disease*

The infectivity to user u_b during a time period T is

$$\mathsf{S}_b = \sum_{u_a^* \in \mathbb{IU}, a \neq b} S_{a^*,b}\left(\mathsf{D}_{a,b}, \frac{1}{\mathsf{TC}_{a,b}}, \mathsf{IF}_{u_a^*}, \frac{1}{\mathsf{IS}_{u_b}}\right). \tag{5.1}$$

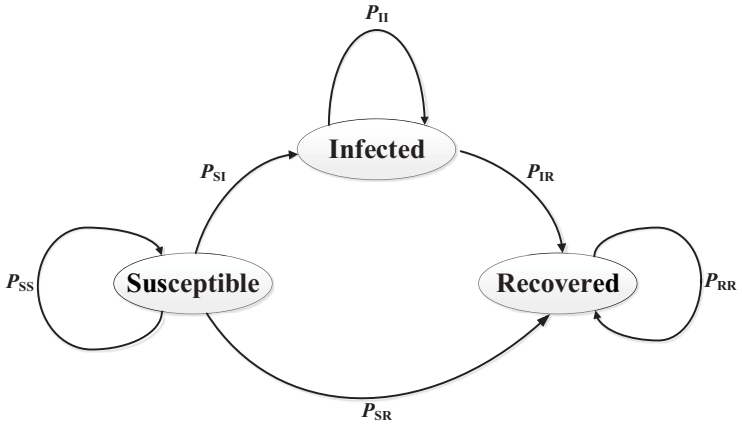


Figure 5.4: Infection states of infectious disease

Here, $S_{a^*,b}$ denotes the instantaneous infectivity from u_a^* to u_b. $S_{a^*,b}$ is proportional to $\mathsf{D}_{a,b}$, $\frac{1}{\mathsf{TC}_{a,b}}$, $\mathsf{IF}_{u_a^*}$ and $\frac{1}{\mathsf{IS}_{u_b}}$. The infectivity increases with the longer contact duration, the closer social relationship, the higher infectivity from the infected user and the lower immunity strength. The infectivity to a user depends on the contacted infected user with the maximum infectivity.

When an infectious disease breaks out in a certain human population, responses in behavior changing according to the outbreak can slow down the progression of the infectious disease to some extent [169]. If a person is aware of the disease in a certain local area or proximity, he would take preventions to considerably reduce his susceptibility. It is important to provide analysis results on the susceptible user's infection .

Generally, a type of infectious disease has three states on a user U_i, i.e., $\mathbb{S}_i = \{s_{i,1}, s_{i,2}, s_{i,3}\}$. $s_{i,j} \in \{$"Susceptible", "Infected", "Recovered"$\}$, as shown in Fig. 5.4. The infection process can be formulated as a state transition model, where the "Susceptible" state is the initial state. When an infected user recovers from the infectious disease, his immune system can generate antibodies against the pervious infected disease. Similarly to [171], in this chapter, we define "Recover" status as the end state.

(3) *Infection Analysis*

To analyze whether user u_j has a risk to get infected, u_j's health and social data can be considered together to classify if u_j is infected. We utilize naive Bayes classification [179] to analyze infection status. Suppose u_j's data $\mathbf{x} = \{x_1, \cdots, x_l\}$ is an l-dimensional vector, where $x_i \in \mathbb{R}$ and $i \in \{1, \cdots, l\}$. The format of $\mathbf{x}$ is shown in Fig. 5.5. A classification

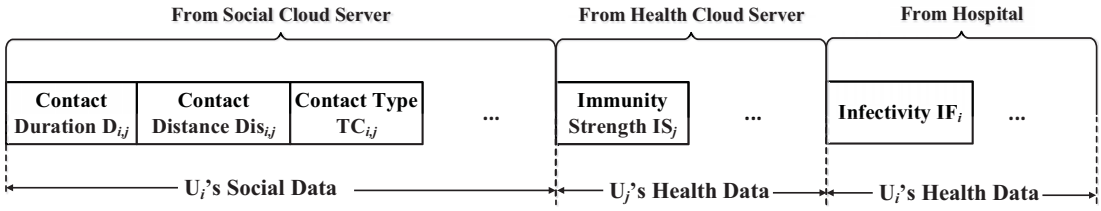


Figure 5.5: Input of Bayesian classification

algorithm $\mathsf{C}(\mathbf{x}, w) : \mathbb{R}^d \mapsto \{\mathbf{c}_1, \cdots, \mathbf{c}_k\}$ takes input as $\mathbf{x}$ and outputs $k^* = \mathsf{C}(w, \mathbf{x}) \in \{1, \cdots, k\}$. Here, k^* is the class (i.e., infection status) to which x corresponds given model w trained by ground truth data. With the abundant health data from hospital, it is feasible to obtain such a model in the PIA. In the model w, each class c_i corresponds to a probability $\{\mathsf{Prob}(C = c_i)\}_{i=1}^k$. The j-th element x_j of $\mathbf{x}$ is a and falls into a class c_i with a probability $\mathsf{Prob}(X_j = a|C = c_i)$. Here, A_j is X_j's domain and $a \in A_j$ ($j \in [1, d]$ and $i \in [1, k]$).

The naive Bayes classifier adopts a maximum a posteriori decision rule to select the class with the highest posterior probability as Equation 5.2.

$$\begin{aligned} k^* &= \arg\max_{i \in [k]} \mathsf{Prob}(C = c_i | X = x) \\ &= \arg\max_{i \in [k]} \mathsf{Prob}(C = c_i, X = x) \\ &= \arg\max_{i \in [k]} \mathsf{Prob}(C = c_i, X_1 = x_1, \cdots, X_d = x_d) \end{aligned} \tag{5.2}$$

$\mathsf{Prob}(X = x)$ is the normalizing factor and deleted given the fixed x according to Bayes theorem.

The Naive Bayes model assumes that $\mathsf{Prob}(C = c_i, X = x)$ can be factorized as

$$\mathsf{Prob}(C = c_i, X_1 = x_1, \cdots, X_d = x_d) = \mathsf{Prob}(C = c_i) \prod_{j=1}^{d} \mathsf{Prob}(X_j = x_j | C = c_i). \tag{5.3}$$

From Equation 5.3, each feature is conditionally independent given the class. The feature value's domain is finite and discrete. The optimal k^* can be selected according to Equation 5.4.

$$
\begin{aligned}
k^* &= \arg\max_{i\in[k]}\{\log \mathsf{Prob}(C=c_i|X=x)\} \\
&= \arg\max_{i\in[k]}\{\log \mathsf{Prob}(C=c_i) + \sum_{j=1}^{d} \log \mathsf{Prob}(X_j=x_j|C=c_i)\}
\end{aligned} \tag{5.4}
$$

The class c_{k^*} corresponds to the infection status of user u_j. The integration of social networking data and health data includes key factors of infection spread and enhances the traditional infection analysis.

5.3.3 Health Data Collection

To preserve user's health data privacy, users should encrypt their data before sending to the cloud servers. We revisit a Ring Learning With Error (RLWE) based homomorphic encryption scheme [180] as the preliminary to construct our building block. In the initialization phase, the TA picks the system parameters as follows: 1) a ring $R = \mathbb{Z}[x]/\langle f_\omega(X)\rangle$ where f_ω is ω-th cyclotomic polynomial; 2) an odd positive integer modulus q and a prime $p \ll q$ as the plaintext base; 3) the dimension n and $N = polylog(q, \omega)$; 4) a ring over modulus q is $R_q = R/qR$; and 5) an error distribution χ with small coefficient.

The TA runs a key generation algorithm KeyGen to generate user u's secret key sk_u and public key pk_u. $sk_u = (1, \mathbf{s}) \in R_q^{n+1}$, where $\mathbf{s}$ is randomly selected from χ^n. The TA randomly selects $\mathbf{e} = (e_1, \cdots, e_N) \in R^N$ and $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_N) \in R_q^N$. Then, the TA computes $\beta_i = \alpha_i \mathbf{s} + p \cdot e_i \bmod (f_\omega(X), q)$. $pk_u = (\beta_i, -\boldsymbol{\alpha})$.

An encryption algorithm Enc takes input as pk_u and message $M \in R_p$. It makes $\mathbf{m} = (M, 0) \in R_q^{N+1}$ and randomly selects $\mathbf{r} = (r_1, \cdots, r_N) \in R_p^N$. The ciphertext is $\mathsf{CT} = \mathsf{Enc}_{pk_u}(M) = \mathbf{m} + \sum_{i=1}^{N} r_i \cdot pk_u \bmod (f_\omega(X), q) \in R_q \times R_q$.

A decryption algorithm Dec takes input as CT and secret key sk_u, and outputs the message $M = \mathsf{Dec}_{sk_u}(\mathsf{CT})$ as $\langle \mathbf{CT}, sk_u\rangle \bmod (f_\omega(X), p)$. Here, $\langle \mathbf{CT}, sk_u\rangle = \sum_{j=1}^{n+1} \mathbf{CT}(j) \cdot sk_u(j)$ denotes the inner product. We have

$$
\langle \mathbf{CT}, sk_u\rangle = M + \sum_{i=1}^{N} r_i \langle sk_u, pk_u\rangle = M + p\sum_{i=1}^{N} r_i e_i = M + p\sum_{i=1}^{N} r_i e_i (\bmod\ f_\omega(X), q). \tag{5.5}
$$

Since $\mathbf{e}$ and $\mathbf{r}$ (i.e., e_i and r_i) are small, $\delta = \sum_{i=1}^{N} r_i e_i (\text{mod } f_\omega(X), q)$ is small such that M can be finally decrypted [180].

This homomorphic encryption scheme can support addition and multiplication operations over ciphertexts. Specifically, the addition of M_1 and M_2 is achieved via component-wise addition of the ciphertexts $\mathsf{Enc}_i(M_1)$ and $\mathsf{Enc}_i(M_2)$. Let $\langle \mathbf{CT}_1, sk_u \rangle = M_1 + p \cdot \delta_1$ and $\langle \mathbf{CT}_2, sk_u \rangle = M_2 + p \cdot \delta_2$. Then, $\langle \mathbf{CT}_1 + \mathbf{CT}_2, sk_u \rangle = (M_1 + M_2) + p \cdot (\delta_1 + \delta_2)$. $M_1 + M_2 = \mathsf{Dec}_i(\mathsf{Enc}_i(M_1) + \mathsf{Enc}_i(M_2))$ if $\delta_1 + \delta_2$ is still small.

To obtain the multiplication of M_1 and M_2, the multiplied ciphertext is $\mathsf{Enc}_i(M_1) \times \mathsf{Enc}_i(M_2)$ as shown in Equation 5.6.

$$\begin{aligned} \langle \mathbf{CT}_1, sk_u \rangle \times \langle \mathbf{CT}_2, sk_u \rangle &= (M_1 + p \cdot \delta_1) \cdot (M_2 + p \cdot \delta_2) \\ &= M_1 \cdot M_2 + p \cdot (p\delta_1\delta_2 + M_1\delta_2 + M_2\delta_1) (\text{mod } f_\omega(X), q) \end{aligned} \tag{5.6}$$

If $p\delta_1\delta_2 + M_1\delta_2 + M_2\delta_1$ is small, $M_1 \times M_2 = \mathsf{Dec}_i(\mathsf{Enc}_i(M_1) \times \mathsf{Enc}_i(M_2))$.

With the addition and multiplication over the ciphertext, homomorphic encryption schemes can allow an untrusted entity to perform these operations without knowing secret keys and the content included in the ciphertexts.

When U_i measures his health data h_i, U_i encrypts h_i as $\mathsf{Enc}_i(h_i)$. To enable the hospital (i.e., trusted entity) to access U_i's health data, U_i generates re-encryption key $\widetilde{RK}_{i \to H}$ to transform $\mathsf{Enc}_i(h_i)$ to $\mathsf{Enc}_H(h_i)$ according to [181] and [180].

5.3.4 Social Data Collection

When two users U_i and U_j move in the physical proximity of each other, the contact information, such as contacted users' identities, contact duration, contact type and social-tie, are recorded by users' smartphones. For example, a Wechat application on smartphones can start a friend discovery program to find the nearby users (running the same application) and allow them to chat with each other. We formulate social contact as follows. Let $\mathsf{CI}(i,j)$ denote the contact data between i and j. $\mathsf{CI}(i,j) = (i, j, \mathsf{D}_{i,j}, \mathsf{TC}_{i,j}, \cdots)$. Then, U_i converts $\mathsf{D}_{i,j}$ to a binary vector $\mathbb{D}_{i,j} = \{D_{i,j,1}, D_{i,j,2}, \cdots, D_{i,j,\omega}\}$ where $\omega = \lceil \log l \rceil$ and l is the maximum duration. For example, if users upload their social information to SC every hour, $l = 60$ with minute as the unit of contact time (or $l = 3600$ when using second as unit). The contact duration is a keyword during the query. It is encrypted as $\mathsf{Enc}_i(D_{i,j}) = \{\mathsf{Enc}_i(D_{i,j,1}), \mathsf{Enc}_i(D_{i,j,2}), \cdots, \mathsf{Enc}_i(D_{i,j,\omega})\}$. If U_i grants the authorization

of social information query to the hospital, U_i generates re-encryption key $RK^*_{i\to H}$ to transform $\mathsf{Enc}_i(D_{i,j})$ to $\mathsf{Enc}_H(D_{i,j})$.

To make user's uploaded social data invisible to the untrusted SC, these data should be encrypted. Let $\mathbb{G}$ be a cyclic group of order p with generator $g \in \mathbb{Z}^*_p$ [182]. U_i randomly chooses his secret key $SK_i = x_i \in \mathbb{Z}_q$. U_i computes $PK_i = g^{x_i}$. During the encryption, U_i randomly chooses $r \in \mathbb{Z}_q$, and encrypts $\mathsf{CI}(i,j)$ as $\mathsf{E}_i(\mathsf{CI}(i,j)) = (c_1, c_2) = (g^r \bmod p, \mathsf{CI}(i,j)g^{x_i r} \bmod p)$. To decrypt $\mathsf{E}_i(\mathsf{CI}(i,j))$, the decryptor computes $\mathsf{CI}(i,j) = c_2/(c_1^{x_i})^{-1}$. Finally, U_i sends $\mathsf{E}_i(\mathsf{CI}(i,j))$ and $\mathsf{Enc}_i(D_{i,j})$ to SC.

If U_i grants H the authorization to query U_i's social information in SC, U_i generates the re-encryption key to SC as a proxy to re-encrypt U_i's ciphertext for the hospital. Specifically, U_i splits his secret key x_i into two parts $x_{i,0}$ and $x_{i,1}$ such that $x_i = x_{i,0} + x_{i,1}$ [183, 184]. SC has the re-encryption key $RK_{i\to H} = x_{i,0}$. H receives the decryption key as $x_{i,1}$. To re-encrypt U_i's ciphertext $\mathsf{E}_i(\mathsf{CI}(i,j))$, SC computes $c'_2 = c_2/(g^r)^{RK_{i\to H}}$ and outputs the ciphertext as $\mathsf{E}_{i\to H}(\mathsf{CI}(i,j)) = (c_1, c'_2)$. Note that H can decrypt U_i's social information by computing $\mathsf{CI}(i,j)g^{x_{i,1}r}/(g^r)^{x_{i,1}}$.

5.3.5 Privacy-preserving Data Query

After making diagnosis of the infected patients, the hospital initiates a query to SC to find the contacted users in a certain period with the infected patients. These users may have potentials to be infected. Since SC is not trusted, the disclosing of users' contact information, e.g., when and where to meet another user, may violate their privacy such that attackers would infer user's habits and preference. In particular, the infected patient's identity is another kind of sensitive information. Imagine that SC knows the hospital querying certain user's social contact data. It is very likely that this queried user either has already been infected or is susceptible. Therefore, it is essential to prevent SC from knowing the query content from the hospital and replied results to the hospital. To protect user's social information from disclosing to SC, the uploaded social contact data should be encrypted. However, it poses a new challenging issue to enable the hospital's oblivious query [185]. To this end, we propose a privacy-preserving data query method (PPDQ) based on conditional oblivious transfer, which allows the hospital to query users' encrypted social contact data in SC without disclosing the query content and results.

The hospital picks the infected patient U_i's identity i and sends $\mathsf{Query}(i,d,s)$ to SC. The hospital receives the query result $\mathsf{Q.Result}(\mathcal{CL}_i)$. Note that $\mathcal{CL}_i = \{\mathsf{CI}(i,j_1), \mathsf{CI}(i,j_2), \cdots, \mathsf{CI}(i,j_m)\}$, where $\mathsf{CI}(i,j_x) = (i, j_x, D_{i,j_x}, ST_{i,j_x})$ $(x \in \{1,\cdots,m\})$ is i's contacted user with $D_{i,j_x} > d$ and $ST_{i,j_x} > s$. For simplicity, we present the details of the query containing

identity and contact duration. The other social metrics can be simply extended based on the PPDQ. The hospital requests a range of query user list (including n users) from SC to blind the exact queried user i.

***Step* 1:** The hospital H builds an identity query vector (n-dimension) $I = \{0, 0, \cdots, 0, 1, 0, \cdots, 0\}$, where i-th element of I is 1 and others are 0 (i.e., H queries U_i's data). Then, the hospital converts the minimum contact duration d to a binary vector $\mathbb{D} = \{D_1, D_2, \cdots, D_\omega\}$. Note that $\omega = \lceil \log l \rceil$. The hospital sends $\mathsf{Enc}_H(I)$ and $\mathsf{Enc}_H(d)$ to SC for query. Here, $\mathsf{Enc}_H(I) = \{\mathsf{Enc}_H(I_1), \mathsf{Enc}_H(I_2), \cdots, \mathsf{Enc}_H(I_n)\}$, and $\mathsf{Enc}_H(d) = \{\mathsf{Enc}_H(D_1), \mathsf{Enc}_H(D_2), \cdots, \mathsf{Enc}_H(D_\omega)\}$. In this section, we present the details of how to query 1-of-n users. The PPDQ can be also extended to query k-of-n users.

***Step* 2:** SC holds $e_{i,0} = \mathsf{E}_{i \to H}(\mathsf{CI}(i, j_x))$ and $e_{i,1} = \perp$. Then, SC performs as follows.

a) Compute $\mathsf{Enc}_H(P_y) = \mathsf{Enc}_H(d_y) - \mathsf{Enc}_H(D_{i,j,y})$ for $1 \leqslant y \leqslant \omega$, implying $P_y = d_y - D_{i,j,y}$.

b) Compute $\mathsf{Enc}_H(R_y) = (\mathsf{Enc}_H(d_y) - \mathsf{Enc}_H(D_{i,j,y}))^2$, implying $R_y = (d_y - D_{i,j,y})^2$.

c) Set $\theta_0 = 0$ and compute $\mathsf{Enc}_H(\theta_y) = 2 \cdot \mathsf{Enc}_H(\theta_{y-1}) + \mathsf{Enc}_H(R_y)$, implying $\theta_y = 2 * \theta_{y-1} + R_y$.

d) Choose a random number $r_y \in \mathbb{Z}_p$ and compute $\mathsf{Enc}_H(\beta_y) = \mathsf{Enc}_H(P_y) + \mathsf{Enc}_H(r_y) \times [\mathsf{Enc}_H(\theta_y) - \mathsf{Enc}_H(1)]$, implying $\beta_y = P_y + r_y(\theta_y - 1)$.

e) Choose a random number $\gamma \in \mathbb{Z}_p$ and compute $\mathsf{Enc}_H(\phi_y)$ as

$$\sum_{i=1}^{n} ((e_{i,1} - e_{i,0})\mathsf{Enc}_H(\beta_y) + (e_{i,1} + e_{i,0})\mathsf{Enc}_H(1)) \times (\gamma(\mathsf{Enc}_H(I_i)^2 - \mathsf{Enc}_H(I_i)) + \mathsf{Enc}_H(I_i)) + \gamma \left(\sum_{i=1}^{n} \mathsf{Enc}_H(I_i) - \mathsf{Enc}_H(1) \right),$$

implying $\phi_y = \sum_{i=1}^{n} (e_{i,1}(\beta_y + 1) + e_{i,0}(1 - \beta_y)) \times (\gamma(I_i^2 - I_i) + I_i) + \gamma \times \left(\sum_{i=1}^{n} I_i - 1 \right)$.

Then, SC has a tuple $\mathsf{Enc}_H(\phi) = \langle \mathsf{Enc}_H(\phi_1), \mathsf{Enc}_H(\phi_2), \cdots, \mathsf{Enc}_H(\phi_\omega) \rangle$. SC randomly permutes this tuple and has $\pi(\mathsf{Enc}_H(\phi))$, which is sent to the hospital as the query result.

***Step* 3:** Receiving the tuple from SC, the hospital decrypts the tuple and obtains the effective query result $2e_{i,0}$ if $d < \mathsf{D}_{i,j}$; and $2e_{i,1}$ otherwise. Finally, the hospital decrypts U_i's social information by computing $\mathsf{CI}(i, j) g^{x_2 r} / (g^r)^{x_2}$.

Finally, the hospital can obtain $\mathcal{CL}_i$=$\{(i, j_1, D_{i,j_1}, ST_{i,j_1}), (i, j_2, D_{i,j_2}, ST_{i,j_2}), \cdots, (i, j_m, D_{i,j_m}, ST_{i,j_m})\}$ where $u_{j_1}, u_{j_2}, \cdots, u_{j_m}$ have contacts (of duration $> d$) with the infected patient U_i.

Algorithm 5: Privacy-preserving Comparison Algorithm

1: **Input**: $\mathsf{Enc}_H(x)$, $\mathsf{Enc}_H(y)$
2: **Output**: $x > y$
3: HC computes $\mathsf{Enc}_H(a) = \mathsf{Enc}_H(y) + \mathsf{Enc}_H(2^l) - \mathsf{Enc}_H(x)$ and randomly selects $r \in (0, 2^{\lambda+l})$. Then, HC computes $\mathsf{Enc}_H(\theta) = \mathsf{Enc}_H(a) + \mathsf{Enc}_H(r)$ and sends $\mathsf{Enc}_H(\theta)$ to H.
4: H decrypts $\mathsf{Enc}_H(\theta)$ by using sk_H, and computes $\eta = \theta \bmod 2^l$.
5: HC computes $\omega = r \bmod 2^l$. Then, HC privately computes $\mathsf{QE}_H(u)$ with H, and obtains $u = 1$ if $\eta < \omega$ according to DGK cryptosystem [187].
6: H encrypts θ_l as $\mathsf{QE}_H(\theta_l)$, which is sent to HC.
7: HC encrypts r_l and computes $\mathsf{QE}_H(\gamma) = \mathsf{QE}_H(u) \cdot \mathsf{QE}_H(\theta_l) \cdot \mathsf{QE}_H(r_l)$. Then, HC sends $\mathsf{QE}_H(\gamma)$ to H.
8: H decrypts γ and finds $\gamma = 0$ if $x > y$; otherwise, $\gamma = 1$.

5.3.6 Privacy-preserving Classification-based Infection Analysis

We propose a privacy-preserving classification-based infection analysis method (PCIA) to analyze the infection status based on naive Bayesian classification. The input vector includes susceptible user's immune strength, contact information with infected user and infected user's infectivity as indicated in Fig. 5.5. The infectivity is diagnosed and assigned by the hospital, while the immunity strength is measured by user and stored on HC. H performs PPDQ with HC to retrieve user's health data without directly disclosing any identity and health data to HC. We present details of the key components of the PCIA, including privacy preservation techniques on comparison, argmax and classification.

i) *Privacy-preserving Comparison (PPC)*

During the comparison, HC compares two ciphertexts of integers x and y encrypted by the hospital H's public key. Let l be the bit length of x and y. Since some operations are on single bit, we adopt Quadratic Residuosity (QR) cryptosystem [186] as the additive homomorphic building block to further improve the computational efficiency. Let QR's plaintext space be $\mathbb{F}_2$ (bits) and $\mathsf{QE}(x)$ is the ciphertext of input bit x. SK_{HC} and PK_{HC} are HC's secret and public keys in QR cryptosystem.

The details can be found in Algorithm 5. HC first injects random number r in the computation of $\mathsf{Enc}_H(x)$ and $\mathsf{Enc}_H(y)$ to blind the comparison results against H. Intuitively, the PPC algorithm checks the most significant bit of $\theta = y + 2^l - x$, indicating whether $x \leqslant y$. In line 5 of Algorithm 5, HC and H privately compute $u = 1$ if $\eta < \omega$ based on DGK cryptosystem [187], which is a practical integer comparison protocol with small plaintext size and ciphertext size. It only requires 5 extra multiplication operations, which improves the algorithm efficiency.

ii) *Privacy-preserving argmax (PPAM)*

Algorithm 6: Privacy-preserving Argmax Algorithm

Input: $\mathsf{Enc}(x_1), \cdot, \mathsf{Enc}(x_n)$
Output: $\mathsf{Enc}(Max)$
HC adopts a random permutation π and computes $\mathsf{Enc}_H(x'_i) = \mathsf{Enc}_H(x_{\pi(i)})$.
Let $\mathsf{max} = 1$ and $\mathsf{Enc}_H(Max) = \mathsf{Enc}_H(x_{\pi(1)})$
for $i = 2 : n$ **do**
    H runs PPC with the result b_i in each iteration. $b_i = 1$ if $Max \leqslant a_{\pi(i)}$; otherwise, $b_i = 0$.
    HC selects two random numbers r_i and $s_i \in (0, 2^{\lambda+l})$. Then, HC computes $\mathsf{Enc}_H(m'_i) = \mathsf{Enc}_H(Max) + \mathsf{Enc}_H(r_i)$ and $\mathsf{Enc}_H(a'_i) = \mathsf{Enc}_H(a_{\pi(i)}) + \mathsf{Enc}_H(s_i)$. Then, $\mathsf{Enc}_H(m'_i)$ and $\mathsf{Enc}_H(a'_i)$ are sent to H.
    if $b_i = 1$ **then**
        H sets $\mathsf{max} = i$, and computes $\mathsf{Enc}_H(v_i) = \mathsf{Refresh}(\mathsf{Enc}_H(a'_i))$.
    else
        H computes $\mathsf{Enc}_H(v_i) = \mathsf{Refresh}(\mathsf{Enc}_H(m'_i))$
    end if
    H sends $\mathsf{Enc}_H(v_i)$ and $\mathsf{Enc}_H(b_i)$ to HC.
    HC computes $\mathsf{Enc}_H(Max) = \mathsf{Enc}_H(v_i) + (\mathsf{Enc}_H(b_i) - \mathsf{Enc}_H(1)) \cdot \mathsf{Enc}_H(r_i) - \mathsf{Enc}_H(b_i) \cdot \mathsf{Enc}_H(s_i)$.
end for
H sends $\mathsf{Enc}_H(\mathsf{max})$ to HC.
Finally, HC computes the result $\pi^{-1}(\mathsf{max})$.

The privacy-preserving argmax algorithm (PPAM) allows HC to output the index of the largest value of $x_1, \cdots, x_n$ encrypted under H's secret key. The PPAM can achieve: 1) H can only learn the index of the largest value but learn nothing else; and 2) H cannot learn the order relations between x_i and x_j. The detailed steps of PPAM is illustrated in Algorithm 7. First, HC adopts a random permutation π to prevent H from learning the order of $\{x_1, \cdots, x_n\}$. With π, HC has $\mathsf{Enc}_H(x'_i) = \mathsf{Enc}_H(x'_{\pi(i)})$. H runs PPC with the result b_i in each iteration (totally n iterations), where $b_i = 1$ if $Max \leqslant a_{\pi(i)}$; otherwise, $b_i = 0$. In each iteration, H can randomize the encryption after determining the maximum of the compared two values. A "refresh" algorithm is introduced to randomize ciphertexts of homomorphic encryption [35]. If the "refresher" knows the secret key, it decrypts the ciphertext and re-encrypts it; otherwise, it multiplies a ciphertext of 0. This "refresh" algorithm is implemented by using re-encryption of homomorphic encryption.

iii) *Privacy-preserving Classification-based Infection Analysis (PCIA)*

In the privacy-preserving classification-based infection analysis (PCIA) method, H and HC computes user u_b's infectious status according to a training set (model) which can be obtained from the ground truth data (in medical center, institution or government). The training process follows [177]. This training set is encrypted by medical health center (T) and stored in HC for classification. T grants the hospital H the authorization of computa-

Algorithm 7: Privacy-preserving Classification-based Infection Analysis Algorithm

1: **Input**: $(\mathsf{Enc}(x_1), \cdot, \mathsf{Enc}(x_n))$ from H
2: **Output**: i^*
3: H form a vector $\mathbf{x} = (x_1, x_2, \cdots, x_d) \in \mathbb{Z}^d$ containing u_b's collected health data related to immunity strength IS_b and u_a^*'s infectivity IF_a (measured by hospital), contact duration and contact type with the infected user u_a^*, i.e., $\mathsf{D}_{a,b}$ and $\mathsf{ST}_{a,b}$ which are queried from SC.
4: HC sends $\mathsf{Enc}_T(P^*(i))$ and $\mathsf{Enc}_T(P_i^j(x))$ (for all possible x in each feature), which are sent to H.
5: H re-encrypts $\mathsf{Enc}_T(P^*(i))$ and $\mathsf{Enc}_T(P_i^j(x))$ to $\mathsf{Enc}_{HC}(P^*(i))$ and $\mathsf{Enc}_{HC}(P_i^j(x))$.
6: **for** $i = 1 : k$ **do**
7: $\quad H$ computes $\mathsf{Enc}_{HC}(\mathsf{Prob}_i) = \mathsf{Enc}_{HC}(P^*(i)) + \sum_{j=1}^{d} \mathsf{Enc}_{HC}(P_i^j(x_j))$.
8: **end for**
9: H runs the PPAM with HC. H obtains $i^* = \arg\max \mathsf{Prob}_i$.

tion between HC and H. This authorization is enabled by re-encryption of homomorphic encryption. The re-encryption key $\widetilde{RK}_{T \to HC}$ is assigned to H and allows H to transfer T's ciphertext to HC's domain. Since the input of homomorphic encryption is integer, the log of probability should be converted to integer by multiplying a constant Δ. For simplicity, let $P^*(i) = \lceil \Delta \log \mathsf{Prob}(C = c_i) \rceil$ and $P_i^j(x) = \lceil \Delta \log \mathsf{Prob}(X_j = x | C = c_i) \rceil$ where $x \in D_j$ the domain of x_j. The detailed steps are as follows.

In summary, the PIA provides a privacy-preserving computing framework not only for hospital to analyze the infection status within the contacted population but also to prevent (infected and susceptible) user's sensitive information from disclosing.

5.4 Security Analysis

In this section, we discuss the privacy features of the PIA according to the design goals in Section 5.2.

5.4.1 Health Data Privacy

We discuss the health data privacy of the PIA in the storage and processing phases. When the health data are stored in HC, user U_i's health data h_i is invisible to HC due to sematic security of homomorphic encryption [180]. In other words, any adversary holding only the public key and ciphertext of h_i cannot learn any information about h_i. Before sending to HC, h_i is encrypted with U_i's public key. Under the honest-but-curious model, HC

cannot decrypt or infer h_i without having U_i's secret key if the ring learning with error (RLWE) assumption holds. The RLWE assumption is that to distinguish the following two distributions is infeasible. The distributions are: 1) a uniform sample $(a_i, b_i) \in R_q^2$; 2) another sample $(a_i, b_i) \in R_q^2$ where we uniformly select $s \in R_q$, then uniformly sample $a_i \in R_q$ and $e_i \in \chi$ to have $b_i = a_i \cdot s + e_i$.

Before the infection analysis, HC re-encrypts $\mathsf{Enc}_i(h_i)$ with $\widetilde{RK}_{i \to H}$ which is the homomorphic re-encryption key to H's domain. Similarly, HC still cannot obtain H's decryption key to know h_i. Meanwhile, the infected patient's infectivity and identity is also encrypted in the ciphertext with H's encryption key. The infected patient's health information is invisible to HC.

In the naive Bayesian classification, each entity's view during the execution and interaction can be simulated according to his input and output. In other words, each entity cannot learn anything except its inputs and outputs, i.e., each party's view generated by a simulator is computationally indistinguishable to his view from the protocol. We show that the PPC, PPAM and PCIA protocols are secure under the honest-but-curious model.

In the PPC protocol, HC's real view is $\mathsf{view}_{HC} = (\mathsf{Enc}_H(x), \mathsf{Enc}_H(y), l, \mathsf{PK}_H, pk_H, r, \mathsf{QE}_H(u), \mathsf{QE}_H(\theta_l))$. We can also build a simulator for HC where the simulator's view is $\mathsf{Sim}_{HC} = (\mathsf{Enc}_H(x), \mathsf{Enc}_H(y), \mathsf{PK}_H, pk_H, \widetilde{r}, \mathsf{QE}_H(\widetilde{\theta}_l))$. Due to the semantic security of the adopted homomorphic encryption scheme, the ciphertexts are indistinguishable. The random number distributions are the same in the real view case and simulation case such that view_{HC} and Sim_{HC} are computationally indistinguishable. Meanwhile, H's real view is $\mathsf{view}_H = (\mathsf{SK}_H, sk_H, l, \mathsf{Enc}_H(\theta), \mathsf{QE}_{HC}(\gamma))$. The view of H's simulator is $\mathsf{Sim}_H = (\mathsf{SK}_H, sk_H, l, \mathsf{Enc}_H(\widetilde{\theta}), \mathsf{QE}_H(\widetilde{\gamma}))$. As the random number r is selected by HC and $\theta = a + r$, θ and $\widetilde{\theta}$ have the same distribution such that they are indistinguishable. Then, $(\mathsf{QE}_H(\theta), \mathsf{QE}_H(\gamma))$ and $\mathsf{QE}_H(\widetilde{\theta}), \mathsf{QE}_H(\widetilde{\gamma})$ are also computationally indistinguishable. H's real view view_H and simulation view Sim_H are also indistinguishable. Therefore, the PPC protocol is secure under the honest-but-curious model.

In the PPAM protocol, HC's real view is $\mathsf{view}_{HC} = (\{\mathsf{Enc}_H(x_i)\}_{i=\{1,\cdots,n\}}, \pi, \mathsf{PK}_H, pk_H; \{r_i, s_i\}_{i=\{1,\cdots,n\}}; \{\mathsf{Enc}_H(v_i), \mathsf{Enc}_H(b_i)\}_{i=\{1,\cdots,n\}}, \pi(\arg\max_{i\in[n]} x_i))$. HC's simulation view is $\mathsf{Sim}_{HC} = (\{\mathsf{Enc}_H(x_i)\}_{i=\{1,\cdots,n\}}, \widetilde{\pi}, \mathsf{PK}_H, pk_H; \{\widetilde{r_i}, \widetilde{s_i}\}_{i=\{1,\cdots,n\}}, \{\mathsf{Enc}_H(\widetilde{v_i}), \mathsf{Enc}_H(\widetilde{b_i})\}_{i=\{1,\cdots,n\}}; \arg\max_{i\in[n]} x_i)$. Since the distributions of r_i, s_i and $\widetilde{r_i}, \widetilde{s_i}$ are the same, they are indistinguishable. Due to the semantic security of the homomorphic encryption scheme and the PPC protocol, $\mathsf{Enc}_H(v_i), \mathsf{Enc}_H(b_i)$ and $\mathsf{Enc}_H(\widetilde{v_i}), \mathsf{Enc}_H(\widetilde{b_i})$ are also indistinguishable. In addition, π and $\widetilde{\pi}$ are selected by HC such that they are indistinguishable. Therefore, view_{HC} and Sim_{HC} are indistinguishable. On the other hand, H's real view is $\mathsf{view}_H = (\mathsf{SK}_H, sk_H; \{b_i\}_{i=\{2,\cdots,n\}}; \{\mathsf{Enc}_H(m_i'), \mathsf{Enc}_H(x_i)\}_{i=\{2,\cdots,n\}})$. The view of H's sim-

ulator is $\mathsf{Sim}_H = (\mathsf{SK}_H, sk_H; \{b_i\}_{i=\{2,\cdots,n\}}; \{\mathsf{Enc}_H(\widetilde{m_i'}), \mathsf{Enc}_H(\widetilde{x_i})\}_{i=\{2,\cdots,n\}})$ Since the permutation π is a mapping function without changing the order of $\{x_i\}_{i=\{1,\cdots,n\}}$, b_i does not change as well. As r_i, s_i are randomly selected by HC, $\mathsf{Enc}_H(m_i')$ and $\mathsf{Enc}_H(\widetilde{m_i'})$ are indistinguishable. Finally, H's real view view_H and simulation view Sim_H are indistinguishable. Therefore, the PPAM protocol is secure under the honest-but-curious model.

In the PCIA protocol, HC cannot view anything other than the inputs since the P-PC and PPAM protocols are both secure under the honest-but-curious model. H's real view is $\mathsf{view}_H = (\mathsf{SK}_H, sk_H, \{x_i\}_{i=\{1,\cdots,n\}}; \{\mathsf{Enc}_H(P_i^j)\}_{i=\{1,\cdots,n\};j=\{1,\cdots,d\}}, \mathsf{Enc}_H(P^*), i^*)$. The view of H's simulator is $\mathsf{Sim}_H = (\mathsf{SK}_H, sk_H, \{x_i\}_{i=\{1,\cdots,n\}}; \{\mathsf{Enc}_H(\widetilde{P_i^j})\}_{i=\{1,\cdots,n\};j=\{1,\cdots,d\}}, \mathsf{Enc}_H(\widetilde{P^*}), \widetilde{i^*})$. Due to the semantic security of the homomorphic encryption scheme, PPC and PPAM protocols, view_H and Sim_H are indistinguishable. Therefore, the PCIA protocol is secure under the honest-but-curious model.

5.4.2 Social Data Privacy

User's social contact information $\mathsf{CI}(i, j)$ is encrypted by U_i with his public key. Without U_i's secret key, SC cannot decrypt and have the plaintext if the decisional Diffie-Hellman problem is hard in $\mathbb{G}$. Therefore, when the social data are stored on SC, no private information of users can be disclosed to SC.

As the hospital H is an honest-but-curious entity in social domain, it follows the protocol without maliciously querying user's social data in SC. Furthermore, the diagnosis from the hospital provides the second-level decryption key for the hospital to decrypt the plaintext of U_i's social contact information. Users are able to grant social contact information access permission by issuing re-encryption key $\widetilde{RK}_{i\to H}$ to allow SC to re-encrypt $\mathsf{CI}(i, j)$ to the hospital's domain. Without the permission, the hospital still cannot decrypt to have $\mathsf{CI}(i, j)$, even though it can obtain the query results from SC. Note that re-encryption is unidirectional such that the users cannot recover the hospital's secret key to decrypt other user's social information. The infected patient's identity is also protected against HC during infection analysis. Therefore, the patient's identities and contact information, including contacted users and duration, are protected against SC. The hospital can only obtain user's social contact information after he is diagnosed as infected.

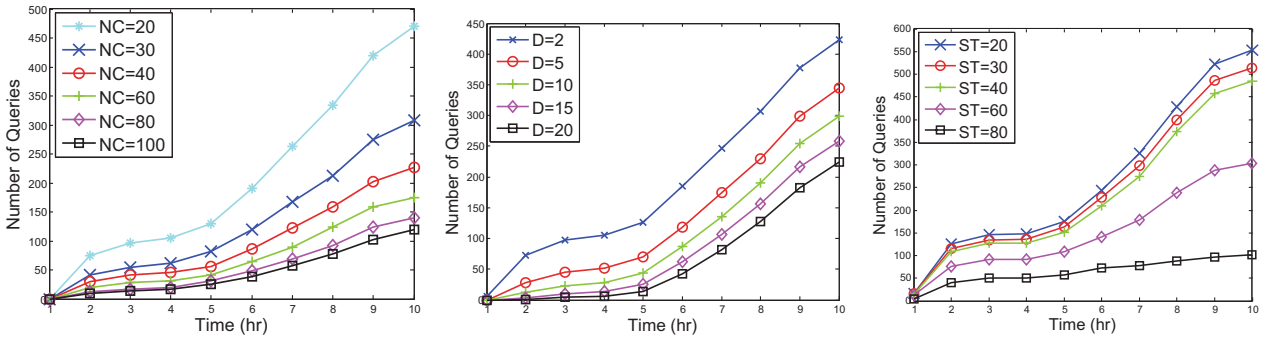
5.4.3 Susceptible and Infected User Privacy

Susceptible user's identity and analysis results can be invisible to HC, SC and any other unauthorized entities. When a patient is diagnosed, the hospital H sends social data query request to SC. During the social data query process, SC learns nothing except that n users are involved in the hospital's query request. But SC cannot know which one user (or k-of-n users) can be queried if PPDQ method is semantic secure under the honest-but-curious model. We show the semantic security of PPDQ method as follows. H's query request $\mathsf{Query}(i, d, s)$ and query result $\mathsf{Q.Result}(\mathcal{CL}_i)$ are privacy-preserving against SC because the adopted homomorphic encryption scheme and ElGamal cryptosystem are semantic secure under the honest-but-curious model. Without the secret key, $\mathsf{Query}(i, d, s)$ and $\mathsf{Q.Result}(\mathcal{CL}_i)$ are invisible to SC. SC's real view is $\mathsf{view}_{SC} = (\mathsf{Enc}_H(d), \mathsf{Enc}_H(I); r_y(1 \leqslant y \leqslant \omega), \mathsf{Enc}_H(\phi))$. The simulator's view of SC is $\mathsf{Sim}_{SC} = (\mathsf{Enc}_H(d), \mathsf{Enc}_H(I); \widetilde{r_y}(1 \leqslant y \leqslant \omega), \widetilde{\mathsf{Enc}_H(\phi)})$. Since the distributions of r_y and $\widetilde{r_y}$ are the same, they are computational indistinguishable. Due to the semantic security of the homomorphic encryption scheme and the PPC protocol, $\mathsf{Enc}_H(\phi)$ and $\widetilde{\mathsf{Enc}_H(\phi)}$ are also indistinguishable. Therefore, SC's real view view_{SC} and simulation view Sim_{SC} are indistinguishable. The identity query vector I can bound the maximum number of H's queried users. H can decrypt the valid result only if $\sum_{i=1}^{n} \mathsf{Enc}_H(I_i) - \mathsf{Enc}_H(1) = 0$. Therefore, the PPDQ is secure under the honest-but-curious model. PPDQ can be secure performed between HC and H when H retrieves users' health data from HC.

Only infected patients grant social data access to H after they are diagnosed in the hospital. Then, the infected patient's decryption key for his re-encrypted ciphertexts is sent to H such that H can decrypt patient's social data after the query. If H arbitrarily builds identity query vector I, H cannot find any valid information due to the semantic security of ElGamal cryptosystem.

5.5 Performance Evaluation

In this section, we evaluate the performance of the PIA with respect to simulation and computational overhead.



(a) Impact of contact number (b) Impact of contact duration (c) Impact of social-tie

Figure 5.6: Impact of social characteristics

5.5.1 Simulations

We conduct extensive simulation based on Infocom06 data set [102], which contains 78 mobile users in a conference. Each user takes a portable device with Bluetooth proximity discovery program to find the nearby users. The social-tie (used to reflect contact type) is also obtained according to user's interactions in the data set. We use this scenario to simulate the infectious disease spread under an indoor environment. In this simulation, we randomly select 8 infected patients with a random assigned infectivity value ranging from 50 to 100. We also set user's immunity strength similarly in the range of $[50, 100]$.

In the simulation, we aim to show the trend of the social characteristic impact other than quantifying the formula between immunity strength and infectivity. The hospital or users can define thresholds to trigger queries, where we consider the thresholds of contact number NC, contact duration D and social-tie ST as shown in Fig. 5.6. From Fig. 5.6(a), we can see that the number of queries decreases with the increasing threshold contact number. When NC is small, e.g., 20 or 30, more queries are triggered since the PIA provides a conservative strategy to include more queries. As shown in Fig. 5.6(b), the decreased duration threshold results in the increasing number of queries since the longer contact between the infected patient and normal users could increase the infection risk of the normal users. In Fig. 5.6(c), the PIA operates with a conservative strategy as ST is small. But the number of queries does not vary too much when ST is from 20 to 40. The reason is that a higher social-tie in a certain range (e.g., in a low level from 20 to 40) may not indicate frequent contacts which are the key factor to accumulate the infection spread [169]. When ST keeps increasing, it shows significant impact on the number of queries. This result also validates the point from [169] that the social relationship is an important

factor to influence the spread process of infectious disease and the close relationships (e.g., students in the same class, or families) may cause severe infection spread. By adjusting ST, the PIA can efficiently notify the people with high social-ties to take actions to prevent the infection spread from human-to-human contact. Therefore, the above results validate the trend in Equation 5.1 and show that the PIA is effective in responding to the spread of infectious disease.

5.5.2 Computational Performance

We use the acute inflammations data set [188] including 120 instances with attributes (i.e., patient's temperature, lumbar pain, urine pushing, micturition pains, urethra status) and corresponding decisions (i.e., inflammation of urinary bladder, and nephritis of renal pelvis origin). We first test the accuracy of the PIA. The total 59 instances with inflammation of urinary bladder and 50 instances with nephritis of renal pelvis origin are all detected. But the PIA detects 47 non-inflammation instances and 59 non-nephritis ones. The accuracy towards individual decision is 88.33% and 90.83%, respectively.

To demonstrate the advantages of using social data for enhancing infection analysis, we generate a data set including the contact information from the real world human trace and synthetic health data. In this data set, each instance contains: contact duration, social-tie, immunity strength, infectivity and infection status. According to [177], we use the 1/4 data set (corresponding to the first day of the conference) to label the training set including 100 instances. We generate 200 input instances with the randomly selected health data (i.e., immunity strength, infectivity and infection status according to [177]) for a baseline classification scheme that only has health data to analyze the infection status. Note that we only label "Susceptible" and "Recovered" in the infection status since we focus on the analysis of infection spread. Meanwhile, we generate 200 instances including contact information from the other 3/4 data set of the real world human trace and the same health data used in the baseline classification scheme. In the 200-instance data set, the number of "Susceptible" and "Recovered" is 120 and 80, respectively. As shown in Table 5.1, the PIA detects 103 "Susceptible" instances and 73 "Recovered" ones, while the baseline scheme detects 71 and 62. Therefore, the integrated social data benefit the infection spread analysis.

With respect to the computational cost of PCIA, we conduct the experiment under a homomorphic encryption library HELib [189] on an Intel Core i5 2.7GHz machine with 4GB RAM to test the computational running time of the proposed methods based on Infocom06 trace. It achieves 80 bits of security with the parameter settings. To mimic

Table 5.1: Infection Analysis Comparison

	PIA	Baseline Scheme
"Susceptible"	103/120 (85.83%)	71/120 (59.16%)
"Recovered"	73/80 (91.25%)	62/80 (77.5%)
Overall	176/200 (88%)	133/200 (66.5%)

the real network environment, we set the communication overhead as 30ms during each interaction of different entities (similar to [35]). In the PPC, H takes 42.94ms, while HC takes 65.674ms. In the PPAM, H takes 6.350s, while HC takes 12.741s. To perform the PCIA, H takes 7.016s, while HC takes 24.282s. Therefore, we can see that HC takes over the majority of the computational overhead since HC has powerful computational capability. The overhead for H is not high.

We also test the running time of the PPDQ with HELib and Crypto++ [190]. We set $l = 1024$, $\omega = 10$ and $n = 78$. H takes 329.234ms to generate the query to SC and retrieve the results, while SC takes 6.487s to return the query results. The majority of computational overhead is at the SC side.

5.6 Related Works

Health data analysis has attracted a lot of attentions from both academic and industrial fields as the big volume of health multimedia data are collected for analysis [191]. Some sophisticated machine learning schemes, such as support vector machine, naive Bayesian classification and decision tree based classification [192], are widely applied in practical applications [35]. These schemes usually require the labeled training data set to establish the learning/classification model, which is used to classify the new data. In addition, abnormal event detection is of great importance especially in health data analysis, and requires prior expert knowledge with well-defined models [193, 194]. Due to rarity, unexpectedness and relevance features of abnormal events, Zhang et al. [112] develop a semi-supervised adapted Hidden Markov Model with Bayesian adaptation to adjust abnormal events. It first labels an abnormal event model in an unsupervised pattern from a large volume of ground truth data. An iterative structure is utilized to adapt any emerging abnormal event at each iteration. This framework can address the difficulty in labeling abnormal events and the scarcity of training data [54].

To leverage privacy preservation and data usability [195] for health data analysis, extensive research efforts have been put in recent years. A variant of "doubly homomorphic"

encryption scheme [196] for secure multi-party computation is introduced to perform flexible operations over the encrypted data. With the advanced and efficient homomorphic encryption techniques [180], Graepel et al. [197] propose a machine learning scheme with privacy preservation to outsource the heavy computation tasks to the powerful cloud servers. At the same time, data confidential and user privacy are achieved with the advantages of the adopted leveled homomorphic encryption scheme. This privacy-preserving machine learning scheme mainly solves the privacy issues during the data training phase. To perform both training and learning over encrypted data, Bost et al. [35] develop a set of secure machine learning classification schemes based on leveled fully homomorphic encryption. In [35], a client performs learning operations with an untrusted server over ciphertexts. In [38], Barni et al. develop a neural network based classification scheme with privacy preservation with linear branching programs to address privacy issues in ECG classification. Samanthula et al. [198] propose a k-nearest neighbor classification algorithm based on Paillier cryptosystem [199] which enables operations over ciphertexts for e-healthcare systems. In [192], a privacy-preserving clinical decision support system is proposed based on naive Bayesian classification. It first aggregates user's health data for training, and then enables untrusted cloud servers to perform secure classification algorithm over encrypted data. Users are also allowed to retrieve top-k diagnosis results with their interests and requests. Yuan et al. [36] propose a privacy-preserving back-propagation neural network learning algorithm based on "doubly homomorphic" encryption. It allows every user to send encrypted data to the cloud server, which performs most of the computation tasks in learning algorithm without compromising the privacy of user's raw data. Another type of lightweight machine learning is decision tree based classification, which is studied in [200] and developed with privacy protection mechanisms. Recently, Zhou et al. [201] propose a secure health text mining scheme, where a privacy-preserving data aggregation method [202] is served as the building block to enable data training in cloud assisted e-healthcare system. Considering the health data access problem, Zhou et al. [203] propose a user-controlled multi-level cooperative authentication scheme to protect user's attribute information from disclosing during the data exchange in healthcare system.

However, most of existing works focus on a single cloud platform involving in e-healthcare systems. Due to the unique characteristics of infectious disease, it is necessary to integrate various sources of user's information, such as health and social data for infection analysis. Meanwhile, the large volume of long-lasting health and social data from users pose a big challenge for data management and collaboration in the traditional e-healthcare framework. Therefore, multiple independent cloud servers with different functionalities are involved in our approach to enhance the infection analysis with sufficient knowledge of patients and susceptible users from both health and social perspectives. In addition,

data privacy, usability (i.e., secure operations over encrypted data) and efficiency should be jointly considered when designing a novel infection analysis system.

5.7 Summary

In this chapter, we have proposed a human-to-human infection analysis approach by utilizing social networking data and health data to enhance infection analysis with privacy preservation. First, we have analyzed the infectious disease spread process and adopted naive Bayesian classification to detect user's infection status. Furthermore, we have exploited social cloud server to collect users' social networking data, and relied on health cloud server to process/classify users' health data. We have proposed a privacy-preserving data query method to enable hospital to query infected patient's social contacts without allowing the social cloud server to infer the patient's identity and contact details. We have also proposed a privacy-preserving classification-based infection analysis method to perform infection analysis over the encrypted social and health data on the health cloud server. Privacy discussion shows that the infected patient's identity and contact details, user's social data and health data are protected from being inferred by untrusted cloud servers and unauthorized entities. Performance evaluation demonstrates that the PIA can enhance infection analysis efficiency and consume acceptable overhead. The PIA provides a social network application for infection analysis with privacy preservation from a novel perspective.

Chapter 6

Conclusions and Future Work

In this thesis, we have investigated security and privacy for MSNs. Based on the aforementioned analysis and discussion, we highlight the main contributions of this thesis and discuss several open research directions for future work.

6.1 Conclusions

In this thesis, we have developed a set of security and privacy protection schemes for MSNs. We summarize the following highlights of this thesis.

- To resist spam in MSNs, we have proposed a personalized fine-grained spam filtering scheme, which allows users to personalize fine-grained keyword-based filters and prevents their private information from disclosing when filtering. By investigating MSN data forwarding process, we have developed a filter distribution scheme based on user's social interests to efficiently distribute filters and block spam. Then, we have proposed privacy-preserving coarse-grained and fine-grained filtering schemes, which enable filter creators to personalize their filters based on their social interests. In addition, we have developed a Merkle Hash tree based filter structure to authenticate the filter validity and update the filters according to user's demands. In the proposed PIF scheme, the filter creator's private information included in his filters can be protected from disclosing. Meanwhile, the PIF cannot only reduce the data forwarding delay, communication and storage overhead but also achieve a high filtering accuracy and efficiency.

- To detect misbehaviors in MSNs, we have investigated mobile users' pseudonym changing and social contact behaviors and proposed a social based Sybil detection scheme according to their abnormal social behaviors in mobile environments. We have exploited the contact statistics of the used pseudonyms and detected Sybil attackers by comparing these contact statistics of pseudonyms from normal users and those from Sybil attackers, when the Sybil attackers frequently change their pseudonyms to cheat other users. In addition, we have proposed a semi-supervised learning scheme with hidden Markov model to detect the collusion among mobile users. Due to the limited storage and computation capabilities of mobile users, we have adopted cloud servers to store and process the massive social contact data from users, alleviating the burden of mobile users. The proposed SMSD scheme also addresses the collusion attacks and resists cloud data modification when employing the untrusted cloud server for mobile Sybil detection..

- We have proposed a novel infection analysis approach, named PIA, to infer human-to-human infection spread by integrating social networking data with health data. We have analyzed the spread process of infectious disease with the consideration of user's social contact and health condition. Several key factors related to infection spread, including immunity strength of the susceptible user, infectivity of the infected patient, their contact duration and contact type, are investigated. We also utilize naive Bayesian classification method to enhance infection analysis with the integration of social and health cloud data from different cloud servers. To address the privacy issues during the collaboration of social and health cloud server, we have proposed a privacy-preserving data query method (PPDQ) based on conditional oblivious transfer to allow the authorized entity (i.e., hospital) to access the infected patient's social networking data from the social cloud server, but not allow the social cloud server to access and infer any data including patient's identity. In addition, we propose a privacy-preserving classification-based infection analysis method (PCIA) to prevent user's private social and health data from disclosing to the untrusted health cloud server. The PCIA enables users to encrypt raw data based on homomorphic encryption and send ciphertexts to the cloud server. Then, the health cloud server can perform the analysis without learning any user's private information.

6.2 Future Research Directions

This thesis introduces the MSN architecture and applications, identifies security and privacy challenges in MSNs, and proposes several promising solutions to achieve security and

privacy goals. Although some preliminary results on security and privacy in MSNs are provided, there are still several open research directions including but not limited to the followings.

6.2.1 Secure and Lightweight Social Data Sharing

MSNs leverage smartphones and wearable devices to offer diverse sensing functionalities, causing continuous computation and communication overhead. Some private information may be inferred from the shared social network data such that it is necessary to check whether the sharing data contain private information, such as private area in a photo, bar code and security information in multimedia to be shared. Transfer learning [204, 205] is a useful approach to investigate new feature or solve a different problem based on the training knowledge [206]. In addition, a privacy risk evaluation [94] is helpful to estimate data owner's potential privacy leakage before sharing data. However, how to define and measure the privacy leakage over social network data becomes critical and challenging.

Meanwhile, to guarantee secure data transmission and sharing, encryption techniques are applied on top of them [207]. Due to the power constraints and portability of smartphones and wearable devices, traditional cryptographic schemes dramatically increase the computation and communication overheads. To this end, it is necessary to develop lightweight cryptographic schemes for the encryption of these social related sensing data measured by smartphones and wearable devices. Several cryptographic schemes, such as NTRU [208, 209], may provide some benefits for lightweight data encryption in terms of overhead. The encryption keys of NTRU are easily created with a reasonable key length. Both of the encryption and decryption of NTRU consume low memory but perform fast. NTRU has the smallest average power consumption, but the largest message size. However, it is still an open problem to develop practical NTRU schemes for MSN data, requiring further research efforts. In addition, compressive sensing [210] is an effective approach to integrating the lightweight data sensing and security, i.e., encryption and signature. Conventional sampling schemes follow Shannon's celebrated theorem: the sampling rate must be at least twice the maximum frequency present in the signal (i.e., Nyquist rate) [211]. Against the common wisdom in data sampling and acquisition, compressive sensing can recover certain signals (or images) based on fewer measurements and samples than traditional methods. It relies on two principles: 1) sparsity (which pertains to the signals of interest) and 2) incoherence (which pertains to the sensing modality). Having the sensing matrix, the raw data, which can be sparsely expressed in some domain (e.g., time, frequency, or wavelet), are compressed with different rates. During the construction of sensing matrix, it is difficult to find such a matrix with low coefficient between any two columns.

We intend to investigate this problem and develop a novel compressive sensing scheme, which performs encryption and signature at the same time, such that the efficiency would be dramatically increased.

6.2.2 Misbehavior Detection

In social applications of MSN, as the smarter attackers trend to mimic normal users to hide themselves against detections [25, 212], the traditional security solutions focusing on resisting the attacking behaviors may not be always effective. The misbehavior detection relies on the learning procedures where learning and training are alternatively applied. Furthermore, human intelligence is highly desirable during the misbehavior modeling and detection to adjust the tunable security and privacy solutions.

Crowdsourcing would be a promising approach to facilitate the existing misbehavior detections [213]. In MSNs, mobile user's detection capability is not as powerful as that at the server side, or even weaker than online users. Outsourcing the detection tasks to the crowd (or a group of mobile users) provides comprehensive knowledge and powerful collaborative detection capabilities. The crowdsourcing users may detect the suspicious Sybil attackers in the early stage via cryptographic schemes, such as authentication of identities associated with user's contacts, and event signatures. The collected detection results or information from crowdsourcing users can assist user's behavior learning, social graph establishment and community detection, which finally benefits the global detection or decision making. Therefore, the crowdsourcing based Sybil detection is envisioned to be a promising tendency for future research directions.

6.2.3 Secure Social Data Processing

In MSNs, it is urgent to allow the cloud server to perform complicated operations over the encrypted social data [192, 197]. Machine learning and data mining algorithms [35], such as neural network and deep learning, can be applied to analyze the social network characteristics and benefit user's social interactions in return. The existing homomorphic encryption [214] and privacy protection techniques [180, 187] can only efficiently support some basic operations, such as addition and multiplication. The cryptographic computation overhead of privacy-preserving machine learning is still too high to be directly applied in the large scale MSNs.

Meanwhile, the anonymity techniques [215, 216] can be integrated with the cryptography schemes to balance the privacy and the social data availability. The unlinkability

is another important feature of designing privacy-preserving machine learning. However, there exist trade-offs among data availability, security and complexity of data processing, especially from the perspective of QoP [64]. In addition, side channel attackers may analyze or infer the type of social data or processing results [217, 218]. The communication patterns of MSN users or even traffic flows are possible to be monitored and analyzed by outside eavesdroppers [219]. For example, in healthcare social network, a global attacker may distribute some malwares at routers or user's devices to monitor the users' data flows. As a result, user's private information, such as relationships with other privacy-sensitive identities (e.g., doctors) and roles of different identities, may be inferred by this global attacker. In addition, some powerful attackers can analyze the operation characteristics on the processed data, e.g., operations over the data and processing time in the cloud server, and infer the data type or whether the data are critical or not. Thus, it is necessary to hide and develop security schemes against side channel attacks when processing social data.

References

[1] Wersm. [Online]. Available: http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/

[2] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, "Applications, architectures, and protocol design issues for mobile social networks: A survey," *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2130–2158, 2011.

[3] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained Private Matching for Proximity-based Mobile Social Networking," in *Proc. of IEEE INFOCOM*, 2012, pp. 1969–1977.

[4] K. Lin, C. Wang, C. Chou, and L. Golubchik, "SocioNet: A Social-Based Multimedia Access System For Unstructured P2P Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 7, pp. 1027–1041, 2010.

[5] G. Cardone, A. Corradi, L. Foschini, and R. Montanari, "Socio-Technical Awareness to Support Recommendation and Efficient Delivery of IMS-enabled Mobile Services," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 82–90, 2012.

[6] I. Roussaki, N. Kalatzis, N. Liampotis, P. Kosmides, M. Anagnostou, K. Doolin, E. Jennings, Y. Bouloudis, and S. Xynogalas, "Context-awareness in wireless and mobile computing revisited to embrace social networking," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 74–81, 2012.

[7] Z. Su, Q. Xu, K. Zhang, K. Yang, and X. Shen, "Dynamic Bandwidth Allocation in Mobile Social Networks with Multiple Homing Access," in *Proc. of WCSP*, 2015, pp. 1–6.

[8] Comscore. [Online]. Available: https://www.comscore.com/Insights/Blog/Millennials-Boast-Huge-Social-Networking-Growth-and-Engagement-on-Smartphones

[9] Google Latitude. [Online]. Available: http://www.google.com/latitude

[10] Loopt. [Online]. Available: http://www.loopt.com

[11] Z. Zhu and G. Cao, "APPLAUS: A Privacy-Preserving Location Proof Updating System for Location-based Services," in *Proc. of IEEE INFOCOM*, 2011, pp. 1889–1897.

[12] R. Zhang, Y. Zhang, and C. Zhang, "Secure Top-k Query Processing via Untrusted Location-based Service Providers," in *Proc. of IEEE INFOCOM*, 2012, pp. 1170–1178.

[13] Dada. [Online]. Available: http://www.playme.it/

[14] N3twork. [Online]. Available: http://www.n3twork.com/

[15] GoogleGlass. [Online]. Available: http://www.google.com/glass/start/

[16] HugShirt. [Online]. Available: http://cutecircuit.com/collections/the-hug-shirt/

[17] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "SACRM: Social Aware Crowdsourcing with Reputation Management in Mobile Sensing," *Computer Communications*, vol. 65, pp. 55–65, 2015.

[18] B. Furht, *Handbook of Social Network Technologies and Applications*. Springer, 2010.

[19] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, "Geo-Community-Based Broadcasting for Data Dissemination in Mobile Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 734–743, 2013.

[20] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[21] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, pp. 606–621, 2009.

[22] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting Channel-aware Reputation System Against Selective Forwarding Attacks in WSNs," in *Proc. of IEEE Globecom*, 2014, pp. 330–335.

[23] A. Abbas and S. U. Khan, "A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431–1441, 2014.

[24] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A Conditional Privacy-Preserving Authentication With Access Linkability for Roaming Service," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 46–57, 2014.

[25] X. Liang, X. Lin, K. Zhang, and X. Shen, "Security and Privacy in Mobile Social Network: Challenges and Solutions," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 33–41, 2014.

[26] K. Zhang, X. Liang, R. Lu, and X. Shen, "Exploiting Multimedia Services in Mobile Social Network from Security and Privacy Perspectives," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 58–65, 2014.

[27] H. Wang, D. Peng, W. Wang, H. Sharif, H. Chen, and A. Khoynezhad, "Resource-aware secure ECG healthcare monitoring through body sensor networks," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 12–19, 2010.

[28] C. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile multimedia applications," in *Proc. of IEEE ICME*, 2003, pp. 137–140.

[29] E. Shi, T. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. of NDSS*, 2011, pp. 1–17.

[30] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[31] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems," in *Proc. IEEE INFOCOM*, 2010, pp. 758–766.

[32] T. Chan, E. Shi, and D. Song, "Privacy-Preserving Stream Aggregation with Fault Tolerance," *Financial Cryptography and Data Security*, pp. 200–214, 2012.

[33] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in *Proc. of EUROCRYPT*, 2003, pp. 416–432.

[34] K. Zhang, R. Lu, X. Liang, J. Qiao, and X. Shen, "PARK: A Privacy-preserving Aggregation Scheme with Adaptive Key Management for Smart Grid," in *Proc. of IEEE ICCC*, 2013, pp. 236–241.

[35] R. Bost, R. Popa, S. Tu, and S. Goldwasser, "Machine Learning Classification over Encrypted Data," in *Proc. of NDSS*, 2015, pp. 1–14.

[36] J. Yuan and S. Yu, "Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 212–221, 2014.

[37] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," *IEEE Internet Computing*, no. 1, pp. 69–73, 2012.

[38] M. Barni, P. Failla, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider, "Privacy-Preserving ECG Classification With Branching Programs and Neural Networks," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 452–468, June 2011.

[39] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in *Proc. of IEEE INFOCOM*, 2010, pp. 534–542.

[40] J. Shahen, J. Niu, and M. Tripunitara, "Mohawk+ T: Efficient Analysis of Administrative Temporal Role-Based Access Control (ATRBAC) Policies," in *Proc. of SACMAT*, 2015, pp. 15–26.

[41] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," *IEEE Network*, vol. 24, no. 4, pp. 19–24, 2010.

[42] Y. Liu, K. Liu, and M. Li, "Passive Diagnosis for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1132–1144, 2010.

[43] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of Resource Constrained Devices in the Internet of Things," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 144–149, 2012.

[44] K. Ren, W. Lou, K. Zeng, and P. Moran, "On Broadcast Authentication in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 4136–4144, 2007.

[45] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[46] H. Celdran, G. Clemente, G. Perez, and M. Perez, "SeCoMan: A Semantic-Aware Policy Framework for Developing Privacy-Preserving and Context-Aware Smart Applications," *IEEE Systems Journal*, to appear.

[47] J. Huang, Y. Meng, X. Gong, Y. Liu, and Q. Duan, "A Novel Deployment Scheme for Green Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 196–205, 2014.

[48] X. Li, R. Lu, X. Liang, X. Shen, J. Chen, and X. Lin, "Smart Community: An Internet of Things Application," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 68–75, 2011.

[49] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An Information Framework of Creating a Smart City through Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[50] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. Biswas, and K. Moessner, "Enabling Smart Cities Through A Cognitive Management Framework for the Internet of Things," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 102–111, 2013.

[51] Q. Lian, Z. Zhang, M. Yang, Y. Zhao, Y. Dai, and X. Li, "An Empirical Study of Collusion Behavior in the Maze P2P File-Sharing System," in *Proc. of IEEE ICDCS*, 2007, pp. 56–66.

[52] M. Yang, Z. Zhang, X. Li, and Y. Dai, "An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System," in *Proc. of IPTPS*, 2005, pp. 182–192.

[53] Businessinsider. [Online]. Available: http://www.businessinsider.com/

[54] K. Zhang, X. Liang, R. Lu, K. Yang, and X. Shen, "Exploiting Mobile Social Behaviors for Sybil Detection," in *Proc. of IEEE INFOCOM*, 2015, pp. 271–279.

[55] D. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-Resilient Online Content Voting," in *Proc. of NSDI*, 2009, pp. 15–28.

[56] Y. Reddy, "A Game Theory Approach To Detect Malicious Nodes In Wireless Sensor Networks," in *Proc. of SENSORCOMM*, 2009, pp. 462–468.

[57] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Zhao, "You are How You Click: Clickstream Analysis For Sybil Detection," in *Proc. of USENIX*, 2013, pp. 241–255.

[58] X. Liang, K. Zhang, R. Lu, X. Lin, and X. Shen, "EPS: An Efficient and Privacy-Preserving Service Searching Scheme for Smart Community," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3702–3710, 2013.

[59] Q. Xu, Z. Su, K. Zhang, P. Ren, and X. Shen, "Epidemic Information Dissemination in Mobile Social Networks with Opportunistic Links," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 3, pp. 399–409, 2015.

[60] M. Katsarakis, G. Fortetsanakis, P. Charonyktakis, A. Kostopoulos, and M. Papadopouli, "On User-Centric Tools for QoE-based Recommendation and Real-Time Analysis of Large-Scale Markets, year=2014, volume=52, number=9, pages=37-43,," *IEEE Communications Magazine*.

[61] M. He, K. Zhang, and X. Shen, "PMQC: A Privacy-preserving Multi-quality Charging Scheme in V2G Network," in *Proc. of IEEE Globecom*, 2014, pp. 675–680.

[62] A. Luo, C. Lin, K. Wang, L. Lei, and C. Liu, "Quality of Protection Analysis and Performance Modeling in IP Multimedia Subsystem," *Computer Communications*, vol. 32, no. 11, pp. 1336–1345, 2009.

[63] Y. Sun and A. Kumar, "Quality-of-Protection (QoP): A Quantitative Methodology to Grade Security Services," in *Proc. of IEEE ICDCS*, 2008, pp. 394–399.

[64] K. Zhang, K. Yang, X. Liang, Z. Su, X. Shen, and H. Luo, "Security and Privacy for Mobile Healthcare Networks — from Quality-of-Protection Perspective," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 104–112, 2015.

[65] K. Wei, M. Dong, K. Ota, and K. Xu, "CAMF: Context-Aware Message Forwarding in Mobile Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2178–2187, 2015.

[66] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proc. of ACM CCS*, 2012, pp. 662–673.

[67] Nexgate. [Online]. Available: http://nexgate.com/solutions/social-media-spam-abusive-and-offensive-content/

[68] F. Soldo, A. Le, and A. Markopoulou, "Blacklisting Recommendation System: Using Spatio-Temporal Patterns to Predict Future Attacks," *IEEE Journal of Selected Area on Communications*, vol. 29, no. 7, pp. 1423–1437, 2011.

[69] R. Lu, X. Lin, T. H. Luan, X. Liang, X. Li, L. Chen, and X. Shen, "PReFilter: An Efficient Privacy-preserving Relay Filtering Scheme for Delay Tolerant Networks," in *Proc. of IEEE INFOCOM*, 2012, pp. 1395–1403.

[70] K. Zhang, X. Liang, R. Lu, and X. Shen, "SAFE: A Social Based Updatable Filtering Protocol with Privacy-preserving in Mobile Social Networks," in *Proc. of IEEE ICC*, 2013, pp. 6045–6049.

[71] B. Agrawal, N. Kumar, and M. Molle, "Controlling Spam Emails at the Routers," in *Proc. of IEEE ICC*, 2005, pp. 1588–1592.

[72] Z. Li and H. Shen, "SOAP: A Social Network Aided Personalized and Effective Spam Filter to Clean Your e-mail Box," in *Proc. of IEEE INFOCOM*, 2011, pp. 1835–1843.

[73] M. Sirivianos, K. Kim, and X. Yang, "SocialFilter: Introducing Social Trust to Collaborative Spam Mitigation," in *Proc. of IEEE INFOCOM*, 2011, pp. 2300–2308.

[74] 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 *Proc. of Mobihoc*, 2014, pp. 43–52.

[75] K. Zhang, X. Liang, R. Lu, and X. Shen, "PIF: A Personalized Fine-grained Spam Filtering Scheme with Privacy Preservation in Mobile Social Networks," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 41–52, 2015.

[76] J. Kim, K. Chung, and K. Choi, "Spam Filtering With Dynamically Updated URL Statistics," *IEEE Security & Privacy*, vol. 5, no. 4, pp. 33–39, 2007.

[77] R. Henry and I. Goldberg, "Formalizing Anonymous Blacklisting Systems," in *Proc. of IEEE Symposium on Security and Privacy*, 2011, pp. 81–95.

[78] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges," *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, 2007.

[79] A. Ramachandran and N. Feamster, "Understanding the Network-Level Behavior of Spammers," in *Proc. of ACM SIGCOMM*, 2006, pp. 291–302.

[80] J. Kleinberg, "The Small-World Phenomenon: An Algorithm Perspective," in *Proc. of STOC*, 2000, pp. 163–170.

[81] A. Lahmadi, L. Delosières, and O. Festor, "Hinky: Defending against Text-Based Message Spam on Smartphones," in *Proc. of IEEE ICC*, 2011, pp. 1–5.

[82] S. Hameed, X. Fu, P. Hui, and N. Sastry, "LENS: Leveraging Social Networking And Trust To Prevent Spam Transmission," in *Proc. of ICNP*, 2011, pp. 13–18.

[83] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and Evaluation of a Real-Time URL Spam Filtering Service," in *Proc. of IEEE Symposium on Security and Privacy*, 2011, pp. 447–462.

[84] H. Shen and Z. Li, "Leveraging Social Networks for Effective Spam Filtering," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2743–2759, 2014.

[85] K. Li, Z. Zhong, and L. Ramaswamy, "Privacy-Aware Collaborative Spam Filtering," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 725–739, 2009.

[86] L. Fan, Z. Lu, W. Wu, B. Thuraisingham, H. Ma, and Y. Bi, "Least Cost Rumor Blocking in Social Networks," in *Proc. of IEEE ICDCS*, 2013, pp. 540–549.

[87] D. Shah and T. Zaman, "Rumors in a Network: Who's the Culprit?" *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[88] Z. Wang, W. Dong, W. Zhang, and C. Tan, "Rooting our Rumor Sources in Online Social Networks: The Value of Diversity From Multiple Observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 663–677, 2015.

[89] G. Stringhini, M. Egele, A. Zarras, T. Holz, C. Kruegel, and G. Vigna, "B@bel: Leveraging Email Delivery for Spam Mitigation," in *Proc. of USENIX Security*, 2012, pp. 16–32.

[90] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication Routing in DTNs: A Resource Allocation Approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 596–609, 2010.

[91] W. Gao, Q. Li, B. Zhao, and G. Cao, "Social-Aware Multicast in Disruption-Tolerant Networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1553–1566, 2012.

[92] K. Zhang, X. Liang, M. Barua, R. Lu, and X. Shen, "PHDA: A Priority based Health Data Aggregation With Privacy Preservation for Cloud Assisted WBANs," *Information Sciences*, vol. 284, pp. 1–12, 2014.

[93] K. Zhang, X. Liang, R. Lu, X. Shen, and H. Zhao, "VSLP: Voronoi-socialspot-aided Packet Forwarding Protocol With Receiver Location Privacy in MSNs," in *Proc. of IEEE GLOBECOM*, 2012, pp. 348–353.

[94] K. Zhang, X. Liang, R. Lu, and X. Shen, "Exploiting Private Profile Matching for Efficient Packet Forwarding in Mobile Social Networks," *Opportunistic Mobile Social Networks*, pp. 283–312, 2014.

[95] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in *Proc. of CRYPTO*, vol. 2139, 2001, pp. 213–229.

[96] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, "Fully Anonymous Profile Matching in Mobile Social Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 641–655, 2013.

[97] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data," in *Proc. of TCC*, 2007, pp. 535–554.

[98] J. Park, "Efficient Hidden Vector Encryption for Conjunctive Queries on Encrypted Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 10, pp. 1483–1497, 2011.

[99] M. Wen, R. Lu, K. Zhang, J. Lei, X. Liang, and X. Shen, "PaRQ: A Privacy-Preserving Range Query Scheme Over Encrypted Metering Data for Smart Grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 178–191, 2013.

[100] R. Merkle, "Protocols for Public Key Cryptosystems," in *Proc. of IEEE Symposium on Security and Privacy*, 1980, pp. 122–134.

[101] Szydlo, "Merkle Tree Traversal in Log Space and Time," in *Proc. of EUROCRYPT*, vol. 3027, 2004, pp. 541–554.

[102] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/infocom (v. 2006-01-31)," Jan. 2006.

[103] J. Zhou, Z. Cao, X. Dong, X. Lin, and A. Vasilakos, "Securing m-healthcare Social Networks: Challenges, Countermeasures and Future Directions," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 12–21, 2013.

[104] D. Quercia and S. Hailes, "Sybil Attacks Against Mobile Users: Friends and Foes to the Rescue," in *Proc. of IEEE INFOCOM*, 2010, pp. 336–340.

[105] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks Via Social Networks," *IEEE ACM Transactions on Networking*, vol. 16, no. 3, pp. 576–589, 2008.

[106] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense Against Sybil Attacks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 885–898, 2010.

[107] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Zhao, and Y. Dai, "Uncovering Social Network Sybils in the Wild," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 1–7, 2014.

[108] X. Liang, X. Lin, and X. Shen, "Enabling Trustworthy Service Evaluation in Service-Oriented Mobile Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 310–320, 2014.

[109] X. Lin, "LSR: Mitigating Zero-Day Sybil Vulnerability in Privacy-Preserving Vehicular Peer-to-Peer Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 237–246, 2013.

[110] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil Attacks and Their Defenses in the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 372–383, 2014.

[111] F. Zhang, R. Safavi-Naini, and W. Susilo, "An Efficient Signature Scheme from Bilinear Pairings and Its Applications," in *Proc. of Public Key Cryptography*, 2004, pp. 277–290.

[112] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-Supervised Adapted HMMs for Unusual Event Detection," in *Proc. of CVPR*, 2005, pp. 611–618.

[113] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[114] C. Bron and J. Kerbosch, "Finding All Cliques of An Undirected Graph (Algorithm 457)," *Communications of the ACM*, vol. 16, no. 9, pp. 575–576, 1973.

[115] A. Cheng and E. Friedman, "Sybilproof Reputation Mechanisms," in *Proc. of SIGCOMM*, 2005, pp. 128–132.

[116] K. Walsh and E. Sirer, "Experience With An Object Reputation System For Peer-to-Peer Filesharing," in *Proc. of NSDI*, 2006, pp. 11–14.

[117] R. Andersen, F. Chung, and K. Lang, "Local Graph Partitioning Using PageRank Vectors," in *Proc. of FOCS*, 2006, pp. 475–486.

[118] T. Haveliwala, "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[119] R. Morselli, B. Bhattacharjee, M. Marsh, and A. Srinivasan, "Efficient Lookup on Unstructured Topologies," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 62–72, 2007.

[120] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday Paradox, Coupon Collectors, Caching Algorithms and Self-organizing Search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.

[121] F. Spitzer, *Principles of random walk*. Springer, 1964.

[122] H. Yu, "Sybil Defenses Via Social Networks: A Tutorial And Survey," *SIGACT News*, vol. 42, no. 3, pp. 80–101, 2011.

[123] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "SoK: The Evolution of Sybil Defense via Social Networks," in *IEEE Symposium on Security and Privacy*, 2013, pp. 382–396.

[124] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All Your Contacts Are Belong To Us: Automated Identity Theft Attacks On Social Networks," in *Proc. of WWW*, 2009, pp. 551–560.

[125] L. Von Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using Hard AI Problems For Security," pp. 294–311, 2003.

[126] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs Over Time: Densification Laws, Shrinking Diameters And Possible Explanations," in *Proc. of KDDWS*, 2005, pp. 177–187.

[127] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the Detection of Fake Accounts in Large Scale Social Online Services," in *Proc. of NSDI*, 2012, pp. 1–14.

[128] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes Using Social Networks," in *Proc. of NDSS*, 2009, pp. 1–15.

[129] P. Denning, "Fault Tolerant Operating Systems," *ACM Computing Surveys*, vol. 8, no. 4, pp. 359–389, 1976.

[130] P. Fong, "Preventing Sybil Attacks by Privilege Attenuation: A Design Principle for Social Network Systems," in *Proc. of IEEE Symposium on Security and Privacy*, 2011, pp. 263–278.

[131] N. Tran, J. Li, L. Subramanian, and S. Chow, "Optimal Sybil-Resilient Node Admission Control," in *Proc. of IEEE INFOCOM*, 2011, pp. 3218–3226.

[132] Q. Cao and X. Yang, "SybilFence: Improving Social-Graph-Based Sybil Defenses with User Negative Feedback," *CoRR*, 2013.

[133] P. Dandekar, A. Goel, R. Govindan, and I. Post, "Liquidity In Credit Networks: A Little Trust Goes A Long Way," in *Proc. of ACM EC*, 2011, pp. 147–156.

[134] B. Viswanath, M. Mondal, K. Gummadi, A. Mislove, and A. Post, "Canal: Scaling Social Network-based Sybil Tolerance Schemes," in *Proc. of EuroSys*, 2012, pp. 309–322.

[135] A. Mohaisen, N. Hopper, and Y. Kim, "Keep Your Friends Close: Incorporating Trust Into Social Network-based Sybil Defenses," in *Proc. of IEEE INFOCOM*, 2011, pp. 1943–1951.

[136] R. Delaviz, N. Andrade, J. Pouwelse, and D. Epema, "SybilRes: A Sybil-resilient Flow-Based Decentralized Reputation Mechanism," in *Proc. of IEEE ICDCS*, 2012, pp. 203–213.

[137] B. Viswanath, A. Post, K. Gummadi, and A. Mislove, "An Analysis of Social Network-based Sybil Defenses," in *Proc. of SIGCOMM*, 2010, pp. 363–374.

[138] W. Wei, F. Xu, C. Tan, and Q. Li, "SybilDefender: Defend Against Sybil Attacks in Large Social Networks," in *Proc. of IEEE INFOCOM*, 2012, pp. 1951–1959.

[139] L. Shi, S. Yu, W. Lou, and T. Hou, "SybilShield: An Agent-aided Social Network-based Sybil Defense Among Multiple Communities," in *Proc. of IEEE INFOCOM*, 2013, pp. 1034–1042.

[140] Z. Cai and C. Jermaine, "The Latent Community Model for Detecting Sybils in Social Networks," in *Proc. of NDSS*, 2012, pp. 1–13.

[141] J. Xue, Z. Yang, X. Yang, X. Wang, L. Chen, and Y. Dai, "VoteTrust: Leveraging Friend Invitation Graph to Defend Against Social Network Sybils," in *Proc. of IEEE INFOCOM*, 2013, pp. 2400–2408.

[142] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, and B. Zhao, "Understanding Latent Interactions in Online Social Networks," in *Proc. of IMC*, 2010, pp. 369–382.

[143] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. of IEEE CVPR*, 2005, pp. 886–893.

[144] C. Hsu and C. Lin, "A Comparison of Methods for Multi-class Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[145] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. Metzger, H. Zheng, and B. Zhao, "Social Turing Tests: Crowdsourcing Sybil Detection," in *Proc. of NDSS*, 2012, pp. 1–16.

[146] H. Yu, C. Shi, M. Kaminsky, P. Gibbons, and F. Xiao, "DSybil: Optimal Sybil-Resistance for Recommendation Systems," in *IEEE Symposium on Security and Privacy*, 2009, pp. 283–298.

[147] W. Chang, J. Wu, C. Tan, and F. Li, "Sybil Defenses in Mobile Social Networks," in *Proc. of IEEE GLOBECOM*, 2013, pp. 641–646.

[148] D. Boneh and H. Shacham, "Group Signatures With Verifier-Local Revocation," in *Proc. of CCS*, 2004, pp. 168–177.

[149] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis and Defenses," in *Proc. of IPSN*, 2004, pp. 259–268.

[150] Q. Zhang, P. Wang, D. Reeves, and P. Ning, "Defending against sybil attacks in sensor networks," in *Proc. of IEEE ICDCS*, 2005, pp. 185–191.

[151] F. Li, P. Mittal, M. Caesar, and N. Borisov, "SybilControl: Practical Sybil Defense with Computational Puzzles," *CoRR*, vol. abs/1201.2657, 2012.

[152] T. Zhou, R. Choudhury, P. Ning, and K. Chakrabarty, "Privacy-Preserving Detection of Sybil Attacks in Vehicular Ad Hoc Networks," in *Proc. of MobiQuitous*, 2007, pp. 1–8.

[153] ——, "P^{2}DAP - Sybil Attacks Detection in Vehicular Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 582–594, 2011.

[154] B. Triki, S. Rekhis, M. Chammem, and N. Boudriga, "A Privacy Preserving Solution for the Protection against Sybil Attacks in Vehicular Ad Hoc Networks," in *Proc. of WMNC*, 2013, pp. 1–8.

[155] J. Ni, K. Zhang, X. Lin, H. Yang, and X. Shen, "AMA: Anonymous Mutual Authentication with Traceability in Carpooling Systems," in *Proc. of IEEE ICC*, 2016, pp. 1–6.

[156] K. Ren, H. Su, and Q. Wang, "Secret Key Generation Exploiting Channel Characteristics in Wireless Communications," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 6–12, 2011.

[157] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and Channel-aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, to appear.

[158] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Channel-based Detection of Sybil Attacks in Wireless Networks," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 492–503, 2009.

[159] X. Ding, H. Zhao, J. Zhu, K. Zhang, and D. Li, "A novel localization algorithm based on rssi for wireless sensor networks," in *Proc. of WiCOM*, 2011, pp. 1–4.

[160] M. Demirbas and Y. Song, "An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks," in *Proc. of WOWMOM*, 2006, pp. 564–570.

[161] S. Lv, X. Wang, X. Zhao, and X. Zhou, "Detecting the Sybil Attack Cooperatively in Wireless Sensor Networks," in *Proc. of CIS*, 2008, pp. 442–446.

[162] G. Guette and B. Ducourthial, "On the Sybil Attack Detection in VANET," in *Proc. of MASS*, 2007, pp. 1–6.

[163] B. Yu, C. Xu, and B. Xiao, "Detecting Sybil Attacks In VANETs," *Journal of Parallel Distributed Compututing*, vol. 73, no. 6, pp. 746–756, 2013.

[164] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat, "Lightweight Sybil Attack Detection in MANETs," *IEEE Systems Journal*, vol. 7, no. 2, pp. 236–248, 2013.

[165] C. Piro, C. Shields, and B. Levine, "Detecting the Sybil Attack in Mobile Ad hoc Networks," in *Proc. of SecureComm*, 2006, pp. 1–11.

[166] M. Mutaz, L. Malott, and S. Chellappan, "Leveraging Platoon Dispersion for Sybil Detection in Vehicular Networks," in *Proc. of PST*, 2013, pp. 340–347.

[167] S. Park, B. Aslam, D. Turgut, and C. Zou, "Defense Against Sybil Attack In The Initial Deployment Stage Of Vehicular Ad Hoc Network Based On Roadside Unit Support," *Security and Communication Networks*, vol. 6, no. 4, pp. 523–538, 2013.

[168] "The chief public health officer's report on the state of public health in canada (infectious disease — the never-ending threat)," 2013. [Online]. Available: http://www.phac-aspc.gc.ca/cphorsphc-respcacsp/2013/infections-eng.php

[169] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale Mobility Networks and The Spatial Spreading of Infectious Diseases," *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 484–21 489, 2009.

[170] Z. Sun, F. Wang, and J. Hu, "LINKAGE: An Approach for Comprehensive Risk Prediction for Care Management," in *Proc. of SIGKDD*, 2015, pp. 1145–1154.

[171] S. Cauchemez, A. Bhattarai, T. Marchbanks, R. Fagan, S. Ostroff, N. Ferguson, D. Swerdlow, S. Sodha, M. Moll, and F. Angulo, "Role of Social Networks in Shaping Disease Transmission During a Community Outbreak of 2009 H1N1 Pandemic Influenza," *Proceedings of the National Academy of Sciences*, vol. 108, no. 7, pp. 2825–2830, 2011.

[172] D. Chen, J. Yang, R. Malkin, and H. Wactlar, "Detecting Social Interactions of The Elderly in A Nursing Home Environment," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 1, pp. 6:1–25, 2007.

[173] B. J. Kolowitz, G. R. Lauro, J. Venturella, V. Georgiev, M. Barone, C. Deible, and R. Shrestha, "Clinical Social NetworkingA New Revolution in Provider Communication and Delivery of Clinical Information across Providers of Care?" *Journal of Digital Imaging*, vol. 27, no. 2, pp. 192–199, 2014.

[174] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[175] M. Barreno, B. Nelson, R. Sears, A. Joseph, and J. Tygar, "Can Machine Learning Be Secure?" in *Proc. of ASIACCS*, 2006, pp. 16–25.

[176] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Proc. of EUROCRYPT*, 2011, pp. 129–148.

[177] N. Ferguson, D. Cummings, S. Cauchemez, C. Fraser, S. Riley, A. Meeyai, S. Iamsirithaworn, and D. Burke, "Strategies for containing an emerging influenza pandemic in southeast asia," *Nature*, vol. 437, no. 7056, pp. 209–214, 2005.

[178] C. Fraser, S. Riley, R. Anderson, and N. Ferguson, "Factors that Make an Infectious Disease Outbreak Controllable," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 16, pp. 6146–6151, 2004.

[179] A. McCallum and K. Nigam, "A Comparison Of Event Models For Naive Bayes Text Classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, 1998, pp. 41–48.

[180] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," in *Proc. of ITCS*, 2012, pp. 309–325.

[181] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proc. of STOC*, 2009, pp. 169–178.

[182] T. ElGamal, "A Public Key Cryptosystem And A Signature Scheme Based On Discrete Logarithms," in *Advances in cryptology*, 1985, pp. 10–18.

[183] A.-A. Ivan and Y. Dodis, "Proxy Cryptography Revisited," in *Proc. of NDSS*, 2003, pp. 1–20.

[184] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes With Applications To Secure Distributed Storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[185] I. Blake and V. Kolesnikov, "Strong Conditional Oblivious Transfer and Computing on Intervals," in *Proc. of ASIACRYPT*, 2004, pp. 515–529.

[186] S. Goldwasser and S. Micali, "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information," in *Proc. of STOC*, 1982, pp. 365–377.

[187] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic Encryption And Secure Comparison," *International Journal of Applied Cryptography*, vol. 1, pp. 22–31, 2008.

[188] "Acute Inflammations Data Set," 2009. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations

[189] "HELib," 2013. [Online]. Available: http://shaih.github.io/HElib/index.html

[190] "Crypto++," 2015. [Online]. Available: https://www.cryptopp.com/

[191] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big Data-driven Optimization for Mobile Networks Toward 5G," *IEEE Network*, vol. 30, no. 1, pp. 44–51, 2016.

[192] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-Preserving Patient-Centric Clinical Decision Support System on Naive Bayesian Classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 2, pp. 655–668, 2016.

[193] C. Stauffer and W. Grimson, "Learning Patterns Of Activity Using Real-Time Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[194] H. Zhong, J. Shi, and M. Visontai, "Detecting Unusual Activity in Video," in *Proc. of CVPR*, 2004, pp. 819–826.

[195] R. Rivest, "Cryptography and Machine Learning," *Lecture Notes in Computer Science*, vol. 739, pp. 427–439, 1993.

[196] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Proc. of TCC*, vol. 2, 2005.

[197] T. Graepel, K. Lauter, and M. Naehrig, "ML Confidential: Machine Learning on Encrypted Data," in *Proc. of ICISC*, vol. 7839, 2012, pp. 1–21.

[198] B. Samanthula, Y. Elmehdwi, and W. Jiang, "k-Nearest Neighbor Classification over Semantically Secure Encrypted Relational Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1261–1273, May 2015.

[199] P. Paillier and D. Pointcheval, "Efficient Public-Key Cryptosystems Provably Secure Against Active Adversaries," in *Proc. of ASIACRYPT*, 1999, pp. 1–13.

[200] P. Fong and J. Weber-Jahnke, "Privacy Preserving Decision Tree Learning Using Unrealized Data Sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 353–364, 2012.

[201] J. Zhou, Z. Cao, X. Dong, and X. Lin, "PPDM: Privacy-preserving Protocol for Dynamic Medical Text Mining and Image Feature Extraction from Secure Data Aggregation in Cloud-assisted e-Healthcare Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1332–1344, 2015.

[202] J. Ni, K. Zhang, X. Lin, and X. Shen, "EDAT: Efficient Data Aggregation without TTP for Privacy-Assured Smart Metering," in *Proc. of IEEE ICC*, 2016, pp. 1–6.

[203] J. Zhou, X. Lin, X. Dong, and Z. Cao, "PSMPA: Patient Self-Controllable and Multi-Level Privacy-Preserving Cooperative Authentication in Distributedm-Healthcare Cloud Computing System," *IEEE Transactions on Parallel and Distribited Systems*, vol. 26, no. 6, pp. 1693–1703, 2015.

[204] S. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[205] Z. Ding, M. Shao, and Y. Fu, "Missing Modality Transfer Learning via Latent Low-Rank Constraint," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4322–4334, 2015.

[206] Z. Deng, K. Choi, Y. Jiang, and S. Wang, "Generalized Hidden-Mapping Ridge Regression, Knowledge-Leveraged Inductive Transfer Learning for Neural Networks, Fuzzy Systems and Kernel Methods," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2585–2599, 2014.

[207] D. He, C. Chen, S. Chan, J. Bu, and P. Zhang, "Secure and Lightweight Network Admission and Transmission Protocol for Body Sensor Networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 664–674, 2013.

[208] J. Hoffstein, J. Pipher, and J. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem," in *Proc. of ANTS*, 1998, pp. 267–288.

[209] P. Nguyen and D. Pointcheval, "Analysis and Improvements of NTRU Encryption Paddings," in *Proc. of CRYPTO*, 2002, pp. 210–225.

[210] Y. Bao, J. Beck, and H. Li, "Compressive Sampling For Accelerometer Signals In Structural Health Monitoring," *SAGE Publications*, pp. 235–246, 2011.

[211] E. Candès and M. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[212] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and Privacy in Mobile Crowdsourcing Networks: Challenges and Opportunities," *IEEE Communications*, vol. 53, no. 8, pp. 75–81, 2015.

[213] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting Mobile Crowdsourcing for Pervasive Cloud Services: Challenges and Solutions," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 98–105, 2015.

[214] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can Homomorphic Encryption Be Practical?" in *Proc. of CCSW*, 2011, pp. 113–124.

[215] K. El-Emam, F. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley, "A Globally Optimal k-Anonymity Method for the De-Identification of Health Data," *American Medical Informatics Association*, vol. 5, no. 16, pp. 670–682, 2009.

[216] L. Sweeney, "Achieving k-Anonymity Privacy Protection Using Generalization and Suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[217] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-Channel Leaks in Web Applications: A Reality Today, a Challenge Tomorrow," in *Proc. of IEEE Symposium on Security and Privacy*, 2010, pp. 191–206.

[218] Forbes. [Online]. Available: http://www.forbes.com/sites/robertvamosi/2015/07/20/side-channel-analysis-can-protect-iot-scada/

[219] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "SAGE: A Strong Privacy-preserving Scheme Against Global Eavesdropping for ehealth Systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 365–378, 2009.

List of Publications

Book/Book Chapter

[B1]. **K. Zhang**, and X. Shen, "Security and Privacy for Mobile Healthcare Networks," **Springer Verlag**, November 2015. (ISBN-10: 3319247158, ISBN-13: 978-3319247151)

[B2]. **K. Zhang**, X. Liang, R. Lu, and X. Shen, "Exploiting Private Profile Matching for Efficient Packet Forwarding in Mobile Social Networks," **Handbook on Opportunistic Mobile Social Networks**, CRC Press, Taylor & Francis Group, USA, August 2014.

Journal Papers

[J1]. **K. Zhang**, X. Liang, J. Ni, K. Yang, and X. Shen, "Exploiting Social Network Data to Enhance Human-to-Human Infection Analysis Without Privacy Leakage", **IEEE Transactions on Dependable and Secure Computing**, under review.

[J2]. **K. Zhang**, J. Ni, K. Yang, X. Liang, J. Ren, and X. Shen, "Security and Privacy for Smart City Applications: Challenges and Solutions", **IEEE Communications Magazine**, under review.

[J3]. **K. Zhang**, X. Liang, R. Lu, and X. Shen, "PIF: A Personalized Fine-grained Spam Filtering Scheme with Privacy Preservation in Mobile Social Networks", **IEEE Transactions on Computational Social Systems**, vol.2, no.3, pp.41-52, September 2015.

[J4]. **K. Zhang**, K. Yang, X. Liang, Z. Su, X. Shen, and H. Luo, "Security and Privacy for Mobile Healthcare Networks from Quality-of-Protection Perspective", **IEEE Wireless Communications**, vol.22, no.4, pp.104-112, August 2015.

[J5]. **K. Zhang**, X. Liang, M. Barua, R. Lu, and X. Shen, "PHDA: A Priority Based Health Data Aggregation with Privacy Preservation for Cloud Assisted WBANs", **Information Sciences, Elsevier**, vol.284, pp.130-141, November 2014.

[J6]. **K. Zhang**, X. Liang, R. Lu, and X. Shen, "Sybil Attacks and Their Defenses in the Internet of Things", **IEEE Internet of Things Journal**, vol.1, no.5, pp.372-383, October 2014.

[J7]. **K. Zhang**, X. Liang, R. Lu, and X. Shen, "Exploiting Multimedia Services in Mobile Social Network from Security and Privacy Perspectives", **IEEE Communications Magazine**, vol.52, no.3, pp.58-65, March 2014.

[J8]. J. Ren, Y. Zhang, **K. Zhang**, and X. Shen, "Adaptive and Channel-aware Detection of Selective Forwarding Attacks in Wireless Sensor Networks", **IEEE Transactions on Wireless Communications**, to appear.

[J9]. J. Ren, Y. Zhang, **K. Zhang**, A. Liu, J. Chen, and X. Shen, "Lifetime and Energy Hole Evolution Analysis in Data-Gathering Wireless Sensor Networks", **IEEE Transactions on Industry Informatics**, to appear.

[J10]. Q. Xu, Z. Su, **K. Zhang**, P. Ren, and X. Shen, "Epidemic Information Dissemination in Mobile Social Networks with Opportunistic Links", **IEEE Transactions on Emerging Topics on Computing**, vol.3, no.3, pp.399-409, September 2015.

[J11]. K. Yang, **K. Zhang**, J. Ren and X. Shen, "Security and Privacy in Mobile Crowdsourcing Networks: Challenges and Opportunities", **IEEE Communications Magazine**, vol.53, no.8, pp.75-81, 2015.

[J12]. J. Ren, Y. Zhang, **K. Zhang**, and X. Shen, "Exploiting Mobile Crowdsourcing for Pervasive Cloud Services: Challenges and Solutions", **IEEE Communications Magazine**, vol.53, no.53, pp.98-105, 2015.

[J13]. J. Ren, Y. Zhang, **K. Zhang**, and X. Shen, "SACRM: Social Aware Crowdsourcing with Reputation Management in Mobile Sensing", **Computer Communications, Elsevier**, vol.65, pp.55-65, 2015.

[J14]. N. Zhang, N. Cheng, A. Gamage, **K. Zhang**, J. W. Mark, X. Shen, "Cloud Assisted HetNets Toward 5G Wireless Networks", **IEEE Communications Magazine**, vol.53, no.6, pp.59-65, 2015.

[J15]. C. Lai, H. Li, X. Liang, R. Lu, **K. Zhang**, and X. Shen, "CPAL: A Conditional Privacy-Preserving Authentication with Access Linkability for Roaming Service", **IEEE Internet of Things Journal**, vol. 1, no. 1, pp.46-57, 2014.

[J16]. X. Liang, **K. Zhang**, X. Shen, and X. Lin, "Security and Privacy in Mobile Social Networks: Challenges and Solutions", **IEEE Wireless Communications**, vol.21, no.1, pp.33-41, 2014.

[J17]. X. Liang, X. Li, **K. Zhang**, R. Lu, X. Lin, and X. Shen, "Fully Anonymous Profile Matching in Mobile Social Networks," **IEEE Journal on Selected Areas in Communications**, vol.31, no.9, pp.641-655, 2013.

[J18]. X. Liang, **K. Zhang**, R. Lu, X. Lin, and X. Shen, "EPS: An Efficient and Privacy-Preserving Service Searching Scheme for Smart Community," **IEEE Sensors Journal**, vol.13, no.10, pp.3702-3710, 2013.

[J19]. M. Wen, R. Lu, **K. Zhang**, J. Lei, X. Liang, and X. Shen, "PaRQ: A Privacy-Preserving Range Query Scheme Over Encrypted Metering Data for Smart Grid", **IEEE Transactions on Emerging Topics in Computing**, vol.1, no.1, pp.178-191, 2013.

[J20]. M. Wen, **K. Zhang**, J. Lei, X. Liang, and X. Shen, "CIT: A Credit-based Incentive Tariff Scheme with Fraud-traceability for Smart Grid", **Security and Communication Networks**, pp.1-10, November 2013.

Conference Papers

[C1]. **K. Zhang**, X. Liang, R. Lu, K. Yang, and X. Shen. "Exploiting Mobile Social Behaviors for Sybil Detection", in Proc. of IEEE **INFOCOM** — IEEE International Conference on Computer Communications (Acceptance Rate: 19%), Hong Kong, China, pp.271-279, April, 2015.

[C2]. **K. Zhang**, X. Liang, R. Lu, and X. Shen, "SAFE: A Social Based Updatable Filtering Protocol with Privacy-preserving in Mobile Social Networks", in Proc. of IEEE **ICC** — IEEE International Conference on Communications, Budapest, Hungary, pp.6045-6049, June 2013.

[C3]. **K. Zhang**, R. Lu, X. Liang, J. Qiao, and X. Shen, "PARK: A Privacy-preserving Aggregation Scheme with Adaptive Key Management for Smart Grid", in Proc. IEEE

ICCC — IEEE/CIC International Conference on Communications in China, Xi'An, China, pp.236-241, August 2013.

[C4]. **K. Zhang**, X. Liang, R. Lu, X. Shen, and H. Zhao, "VSLP: Voronoi-Socialspot-Aided Packet Forwarding Protocol with Receiver Location Privacy in MSNs", in Proc. IEEE **Globecom** — Global Communication Conference, Anaheim, California, USA, pp.348-353, December, 2012.

[C5]. J. Ni, **K. Zhang**, X. Lin, H. Yang, and X. Shen, "AMA: Anonymous Mutual Authentication with Traceability in Carpooling Systems", Proc. of IEEE **ICC** — IEEE International Conference on Communications, Kuala Lumpur, Malaysia, May 2016.

[C6]. J. Ni, **K. Zhang**, X. Lin, and X. Shen, "EDAT: Efficient Data Aggregation without TTP for Privacy-Assured Smart Metering", Proc. of IEEE **ICC** — IEEE International Conference on Communications, Kuala Lumpur, Malaysia, May 2016.

[C7]. Z. Su, Q. Xu, **K. Zhang**, K. Yang, and X. Shen, "Dynamic Bandwidth Allocation in Mobile Social Networks with Multiple Homing Access", in Proc. of IEEE **WCSP** — International Conference on Wireless Communications and Signal Processing, Nanjing, China, October, 2015.

[C8]. M. He, **K. Zhang**, X. Shen, "PMQC: A Privacy-Preserving Multi-Quality Charging Scheme in V2G network", in Proc. of IEEE **Globecom** — Global Communication Conference, Austin, Texas, USA, pp.675-680, December, 2014.

[C9]. J. Ren, Y. Zhang, **K. Zhang**, X. Shen, "Exploiting Channel-Aware Reputation System Against Selective Forwarding Attacks in WSNs", in Proc. of IEEE **Globecom** — Global Communication Conference, Austin, Texas, USA, pp.330-335, December, 2014.

[C10]. Y. Liu, Z. Shi, **K. Zhang**, Y. Zheng, R. Lu, X Shen, "A Novel Low-power Mixed-mode Implementation of Weight Update in Particle PHD Filters", in Proc. of IEEE **WCNC** — IEEE Wireless Communications and Networking Conference, Shanghai, China, pp.4647-4652, April 2013.

Vita

Kuan Zhang received the B.Sc. degree in Communications Engineering and M.Sc. degree in Computer Science from Northeastern University, Shenyang, China, in 2009 and 2011, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His current research interests include security and privacy for mobile social networks, e-healthcare, Internet of Things, and cloud computing. Mr. Zhang served as a Technical Program Committee Member for IEEE Globecom'15, ICNC'16, VTC-Spring'16 and VTC-Fall'16.