Florida International University
FIU Digital Commons

FIU Electronic Theses and Dissertations

University Graduate School

10-23-2013

Trajectory Privacy Preservation in Mobile Wireless Sensor Networks

Xinyu Jin
Florida International University, xjin001@fiu.edu

DOI: 10.25148/etd.FI13120417

Follow this and additional works at: https://digitalcommons.fiu.edu/etd

Part of the Digital Communications and Networking Commons, Information Security Commons, Systems and Communications Commons, and the Systems Architecture Commons

Recommended Citation
Jin, Xinyu, "Trajectory Privacy Preservation in Mobile Wireless Sensor Networks" (2013). *FIU Electronic Theses and Dissertations*. 983.
https://digitalcommons.fiu.edu/etd/983

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

TRAJECTORY PRIVACY PRESERVATION IN MOBILE WIRELESS SENSOR NETWORKS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Xinyu Jin

2013

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Xinyu Jin, and entitled Trajectory Privacy Preservation in Mobile Wireless Sensor Networks, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Deng Pan

Jeffrey Fan

Jean H. Andrian

Yimin Zhu

S. S. Iyengar

Niki Pissinou, Major Professor

Date of Defense: October 23, 2013

The dissertation of Xinyu Jin is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2013

© Copyright 2013 by Xinyu Jin

All rights reserved.

DEDICATION

To my parents,

Maixiu Chen and Junqi Jin.

ACKNOWLEDGMENTS

First, I would like to express my great gratitude to my advisor, Dr. Niki Pissinou, whose encouragement, guidance and support from the initial to the final level enabled me to complete this work. Her encouraging advice and strict training have been guiding me to conduct independent and ethical research.

I would like to thank Dr. Kia Makki and Dr. Niki Pissinou for offering me the opportunity to pursue my doctoral degree. Without Dr. Makki's enormous guidance and great encouragement to me at the beginning, this work would have not been possible. Thank Dr. Deng Pan, who has been tirelessly guiding me on research problems and encouraging me to resolve life issues. His humble and hard-working way of life sets me a great example to follow.

Many thanks to Dr. Jeffrey Fan, my dissertation committee, who has been patiently and continuously giving me valuable advice during my doctoral study. I also appreciate Dr. Jean H. Andrian, Dr. Yimin Zhu and Dr. S. S. Iyengar for serving on my dissertation committee and giving me enlightening comments.

I would like to acknowledge the interactive research environment provided by the Telecommunication and Information Technology Institute (IT^2). Thank all the IT^2 colleagues. Special thanks go to Dr. Sitthapon Pumpichet, Dr. Qian Wang, Dr. Charles A Kamhoua, Dr. Hao Jin, Dr. Kai Chen, and Dr. Qutub Bakhtiar.

Sincere thanks to Dr. Yu Liu at Xi'an University of Posts & Telecomms, without whom my doctoral would have not been possible. Thanks to Florida International University for supporting my research. Especially thank Dr. Shekhar Bhansali, Ms. Maria Benincasa and the Dept. of Electrical and Computer Engineering.

Finally, I would like to show my deepest gratitude to my parents, Maixiu Chen and Junqi Jin, for giving me unconditional love and inspiration to follow my dream.

ABSTRACT OF THE DISSERTATION

TRAJECTORY PRIVACY PRESERVATION IN MOBILE WIRELESS SENSOR NETWORKS

by

Xinyu Jin

Florida International University, 2013

Miami, Florida

Professor Niki Pissinou, Major Professor

In recent years, there has been an enormous growth of location-aware devices, such as GPS embedded cell phones, mobile sensors and radio-frequency identification tags. The age of combining sensing, processing and communication in one device, gives rise to a vast number of applications leading to endless possibilities and a realization of mobile Wireless Sensor Network (mWSN) applications. As computing, sensing and communication become more ubiquitous, trajectory privacy becomes a critical piece of information and an important factor for commercial success. While on the move, sensor nodes continuously transmit data streams of sensed values and spatiotemporal information, known as “trajectory information”. If adversaries can intercept this information, they can monitor the trajectory path and capture the location of the source node.

This research stems from the recognition that the wide applicability of mWSNs will remain elusive unless a trajectory privacy preservation mechanism is developed. The outcome seeks to lay a firm foundation in the field of trajectory privacy preservation in mWSNs against external and internal trajectory privacy attacks. First, to prevent external attacks, we particularly investigated a context-based trajectory privacy-aware routing protocol to prevent the eavesdropping attack. Traditional shortest-path oriented routing algorithms give adversaries the possibility to locate the target node in a certain area. We designed the novel privacy-aware routing phase and utilized the trajectory dissimilarity between mobile nodes to mislead adversaries about the location where the message started its journey. Second, to detect internal attacks, we developed a software-based attestation solution to detect compromised nodes. We created the dynamic attestation node chain among neighboring nodes to examine the memory checksum of suspicious nodes. The computation time for memory traversal had been improved compared to the previous work. Finally, we revisited the trust issue in trajectory privacy preservation mechanism designs. We used Bayesian game theory to model and analyze cooperative, selfish and malicious nodes' behaviors in trajectory privacy preservation activities.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

In recent years, the field of location-aware sensing devices has been rapidly expanding. The age of combining sensing environments, processing data and communication in one device, is expected to bring an enormous increase of location-aware applications. Along with the novel applications, a new class of sensor networks, mobile Wireless Sensor Networks (mWSNs), is emerging. While on the move, sensors continuously transmit data streams of sensed values and spatiotemporal information, known as “trajectory information”. However, the mobile sensors, which compose such networks, are susceptible to trajectory privacy and security attacks by an adversary tracking the devices over time and space.

Sensors are small battery-powered wireless devices, which are equipped with limited processing power and memory. Sensor network related issues are unique and application-dependent. This includes prolonging network lifetime and security where adversarial environments pose various kinds of threats to reliable network operations. These constraints raise the issue that some of the techniques proposed in existing studies may not be applicable to mWSNs.

1.1 Motivating Applications

This research is aimed to develop an inherent TPP mechanism for mWSNs that can be used for military, civilian safety, and a vast variety of other applications with devices that have limited resources. For example, aerial sensors are deployed in public places to monitor air pollution level or detect toxic gas leaking for public safety. For natural disaster forecasting, mobile sensors are deployed in unattended severe environments to track changes in humidity, temperature, wind speed, and other parameters. Sensors could be mobile by being either self-propelled or attached to moving objects. For instance, they could be carried by remotely controlled robots or human beings to reach the optimized location for phenomena monitoring. In the infrastructure network, sensors communicate with other peer nodes and the centralized base station through wireless media. Each sensor data sample records the location, time stamp and sensed values. Samples are reported to the base station after data aggregation at local network access points.

The colligation of the mobile sensor network and wearable sensor devices is able to provide health condition monitoring and telemedicine services for seniors and chronic patients. A wearable mobile sensor device, Q sensor, has already been invented to measures emotional arousal via skin conductance, as well as temperature and activities of the sensor user [Q-s]. In addition, wearable sensors that measure blood pressure, weight, and even hemoglobin are about to emerge. These sensors are able to provide necessary data for online doctors to offer interactive telemedicine service in real time. Data streams from these sensors are transmitted to the base station by multi-hop communications to access points near houses or along road sides. Whenever a node detects dramatic environmental changes around the user, it creates the alert and reports to the base station. If it is necessary, doctors can request more frequent measurements from the wearable sensor and send reminders to the user for better preparation of the potential climate changes.

Mobile sensors installed on vehicles can be used for traffic monitoring and surveying road conditions. Motion sensors could report the speed and bumpiness of the vehicle, which reflect the traffic and road conditions, and the location and sensing time of the motion sensors, to the traffic management center. Drivers are able to know the traffic condition from both nearby vehicles and the traffic management center in advance to make a better decision on route selection. Some car rental companies have also invested vehicle-embedded mobile sensors to monitor the speeding violation of their clients and provide emergency assistance when unexpected car breakdown happens on the trip.

In these scenarios, trajectory privacy is of critical concern, especially if human beings or vehicles are carrying the sensors. Invasion by unauthorized entities into the private trajectory of a user, such as when and where he/she shops, when and which department in the hospital he/she visits, when and where he/she spends vacations and when and where he/she picks up his/her kids from school, may seriously threaten personal safety. For example, a thief might wait for someone to return to his/her home if he/she knows that person has just gone to the bank. Moreover, trajectory information can reveal personal preferences and habits, which can be used for consumer profiling. Insurance companies can utilize the trajectory information to analyze frequent routes of clients. They might charge higher premiums from the clients that often visit high accident locations. Car rental companies can help to discourage dangerous driving and offer instant assistance by tracking their vehicles. However, on the other side, clients are being "watched" all the time that they seem to open their personal lives to other entities unawares.

1.2 Research Needs and Challenges

While some researchers have addressed problems related to trajectory privacy in stationary wireless sensor networks, little work has been done to address problems related to mWSNs. In much of the earlier research, the objective is to conceal the location of static sensor nodes [KZTO05, XSS06, LR10, SYZC08, YSZ^{+}08, PL12, MLW12, PL11]. Another work [KXTZ07] attempts to conceal the time that a static node observes a target, as opposed to concealing the location of the node. However, the sensor nodes lack mobility in the above designs.

Many researchers applied location obfuscation/generalization methods, such as K-anonymity [Swe02], to generalize trajectory information to preserve users' trajectory privacy to cope with nodes' mobility [CZBP06, MCA06, XC09, GKS07, DBS10, GL08, YJHL08, GDVM09, HHJC12, ACG09]. However, this technique is vulnerable to background knowledge attacks, query sampling attacks and query tracking attacks [MGKV06, KGMP07, CM07]. Additionally, the often required anonymizer unit/agent is an extra burden for resource-restricted sensor nodes. Xu et al. proposed a distributed location generalization algorithm by allowing each node to compute its location cloaking boxes, which satisfy the predefined network safety level while on the move [XC09]. In this method, low node density results in a small cloaking box to satisfy the network safety level, which leads to a trajectory privacy leakage due to the high location resolution.

Another widely used trajectory privacy preservation method in mobile networks is that mobile nodes collectively change their pseudonyms in regions called mix zones [BS03, BS04, FRF$^+$07, FSH09]. The method assumes the existence of a trusted middleware system, positioned between the underlying location system(s) and untrusted third-party applications. To fit in decentralized/distributed networks, researchers proposed allowing each mobile node to dynamically obtain mix zones [HMYS05, LSHP06]. Both of the aforementioned methods are appealing in resolving TPP issues in mWSNs. Nevertheless, when applying the techniques of location obfuscation/generalization or pseudonyms in mix zones, the trust of the anonymizer, the middleware system or cooperating nodes is prerequisite. How to evaluate such trust-based cooperation in non-cooperative networks remains an open problem. Last but not least, cryptographic and authentication mechanisms are susceptible to critical information leaks, such as the time of an event or a set of events, which an adversary can use to infer the location of a targeted node, and therefore cannot be used as a stand-alone method to protect the privacy of mobile nodes' trajectories. Clearly, much work remains to be done.

1.3 Research Objectives

The more confidently a system hides the trajectory path of a wireless-mobile sensor device, the wider the range of mWSN applications. Our research stems from the recognition that applications of mWSNs will remain elusive unless a Trajectory Privacy Preservation (TPP) mechanism is developed. The existing notions of TPP must be modified due to the transition from stationary sensor networks to mWSNs and in-network operations. This dissertation involves the design, development, and simulation evaluation of an in-network TPP framework in mWSNs. Specifically, we investigate the following three topics:

1. Prevent external trajectory privacy attacks

We begin with analyzing a low-cost external trajectory privacy attack and designing the corresponding prevention technique. To analyze the trajectory of a mobile sensor, existing solutions represent an object's "trajectory" as an ordered list of location samples at specific instances in time [GS05]. Although, such a list can adequately represent the location changes of an object throughout its lifespan, it does not contain the data necessary to analyze the suspicious trajectory pattern or relations between peer nodes. For example, a static unauthorized node near access points could be a malicious eavesdropper. A high correlation between the trajectories of neighboring nodes could imply that one or multiple nodes are suspicious followers. In resource-constraint sensor networks, where shortest-path routing protocol is commonly implemented, attackers can track down mobile nodes by eavesdropping into network communication through static eavesdroppers and followers. Therefore, our first objective is to design a new routing protocol to prevent external attacks.

2. Detect internal security attacks

Besides external attacks, such as the eavesdropping attack we will explore in Chapter 3, mWSNs are under the internal attack/node compromise attack as well. Node compromise is a crucial security threat in sensor networks. Since sensor nodes have inherent constraints such as limited energy, processing capability and memory space, they are much more vulnerable than wired devices. Without hardware protection, sensor nodes are easily tampered with and critical-system information can be easily obtained. Under the threat of compromised nodes, nodes' trajectory privacy are hardly able to be protected since compromised nodes have authorization to access the network and key information for decryption. In this scenario, node compromise detection is a critical technique to guarantee network security and eliminate malicious nodes. Hence, trajectory privacy will be improved. Our second objective is to design a internal attack detection method.

3. Analyze nodes' behaviors in TPP activities

Regardless of the attacking type, the restricted resource of sensor nodes determines that relying on single sensor node to defend security and privacy attacks is an impractical approach. On the other hand, seeking cooperation among trustworthy entities could provide potential solutions by utilizing the high mobility and large quantity of mobile nodes in mWSNs. Therefore, cooperative methods are preferred in addressing security and privacy issues in our study. In this case, trust becomes an inevitable issue in the methodology design, especially for autonomous nodes and unattended mWSNs. Hence, there is an obvious need to model and analyze nodes' behaviors in such environment to evaluate the trust among cooperation-involved nodes. Our last objective is to model selfish, malicious and cooperative nodes' behaviors in TPP activities.

1.4 Research Approaches

To protect nodes' trajectory privacy against external attacks, we develop a trajectory privacy-aware routing protocol. We design the Basic Trajectory Privacy (BTPriv) and Secondary Trajectory Privacy (STPriv) preservation algorithms. Both BTPriv and STPriv employ the unique privacy-aware routing phase, where each node selects the next-hop node according to the dynamic trajectory distance in order to hide its trajectory. The privacy-aware routing phase requests each node to route its data packets through the privacy-aware path instead of the shortest path. In order to select the proper next-hop node which helps the target node to hide the location at the time of data transmission from eavesdroppers, the target node needs to collect limited trajectory information from its neighboring nodes. To avoid privacy invasion of the neighbors trajectory privacy, we propose the one-time pad virtual name for message exchanges. Using the trajectory privacy information of the neighbors, the target node computes the dynamic trajectory distance to each neighbor. The dynamic trajectory distance indicates the irrelevance between two trajectories at a specific time. Finally, the target node selects the neighbor which has the highest probability to mislead attackers as the next hop.

To detect the internal attack/node compromise attack, we design a software-based attestation method. Software solutions for the security of WSNs eliminate the need for excess hardware and are able to perform within the power constraints of the sensor network. In the attestation, we use the base station as an external verifier to challenge mobile nodes. A checksum is computed by the memory traversal function from the memory content of the node being challenged. The memory traversal function applies RC4 as the pseudorandom number generator. The selection of the nodes performing the RC4 pseudorandom number generation is unpredictable. This prevents certain nodes from being more susceptible to compromise. After the checksum attestation process, any deviations, according to an expected value, indicate the detection of compromise.

To model and analyze nodes' behaviors in TPP activities, we formulate in-network TPP activities by applying the Bayesian game, named the *TPP* game, to model selfish, malicious and cooperative behaviors of autonomous nodes in decentralized/distributed mWSNs. The selfishness is modeled by deploying trajectory privacy sensitivity customization. We analyze the TPP game from both static and dynamic points of view. The trust degree is evaluated by computing the dynamic belief among cooperation-involved neighbors. Nodes update their belief based on their observation on other nodes' actions. Finally, we suggest the equilibrium strategy for nodes to take actions in the TPP cooperation.

1.5 Research Contributions

Following the research objectives and approaches, we have developed an in-network TPP framework that incorporates protocol design, algorithm development, mathematical modelling and software simulations. To be specific, we make the following contributions:

1. Design a context-based trajectory privacy-aware routing protocol to prevent external attacks [JPC+12]

To prevent external attacks, we develop a distributed context-based trajectory privacy-aware routing protocol by utilizing the trajectory dissimilarity.

- We illustrate the possible external trajectory privacy attack in mWSNs. Since most routing algorithms are shortest-path oriented, attackers can deduce that the source node is within a certain subarea when they “hear” the corresponding packets. Even though messages are encrypted, attackers could crack the certain pattern of encryptions after a period of packets logging. Additionally, some public background knowledge can be utilized to identify users. With such information, attackers can now estimate the location of a specific mobile node.
- We develop the one-time pad virtual name with the purpose of hiding trajectories of the target node without requiring the third party. The one-time pad virtual name is designed for exchanging necessary trajectory information used in TPP cooperation. It provides the solution for data collecting among peer nodes without breaching other nodes’ trajectory privacy.
- We create the privacy-aware routing phase utilizing the dissimilarity of trajectories. The privacy-aware routing phase requests each node to route its data packets through the privacy-aware path instead of the shortest path. The next-hop node is selected based on the dynamic trajectory distance between the current-hop node and the candidate, in order to mislead eavesdroppers where the message starts its journey.

2. Design an unpredictable software-based attestation solution to detect internal attacks [JPP$^+$10]

To detect the node compromise attack, we design an unpredictable software-based attestation solution, which allows the defender to utilize the advantage of on-the-fly implementation.

- We design an efficient software-based attestation mechanism with reduced energy consumption for battery-powered nodes. Compared to the previous related work, our attestation mechanism decreases checksum computation time by almost 48% for selective attested nodes.

- We create dynamic attestation chains to achieve the unpredictability of node verification to avoid creating areas of great susceptibility in the network. The dynamic attestation chain utilizes the mobility of nodes and the highly dynamic network topology. The node-chain relationship is dynamic and independent in different attestations. The role of each node may vary in each attestation.

- We detect compromised nodes which are one hop away from the base station in mWSNs where the nodes are moving around without fixed neighbors. The proposed solution relaxes the assumption of fixed locations and neighborhoods of nodes in attestations. It provides the flexibility to be applied in mobile networks.

3. Model and analyze nodes' behaviors using game theory [JPP^{+}13]

To evaluate the trust degree among autonomous nodes, we apply Bayesian game theory to model selfish, malicious and cooperative nodes' behaviors in TPP activities.

- We apply the concept of trajectory privacy sensitivity customization to model nodes' selfishness. Generally speaking, users have different requirements for TPP levels in different trajectory contexts. Taking control of the TPP level enable users to better preserve their trajectory privacy to specifically meet their needs. Meanwhile, it releases more available peer nodes to participate into TPP cooperation.

- We formulate the strategic TPP game among two nodes considering selfish, malicious and cooperative behaviors of autonomous nodes. The game is viewed as a strategic game, where one of the players has a private type at the beginning of the game. We provide pure-strategy and mixed-strategy Bayesian Nash Equilibrium [Mye97] of the game and suggest the trust degree in TPP activities.
- We analyze the TPP game in the dynamic form and provide the belief system for nodes to update the trust degree dynamically in the multi-stage TPP game. We examine the dynamic TPP game in terms of Bayesian Condition and derive the Perfect Bayesian Equilibrium [FT91]. We simulate the TPP game in a mWSN composed of 200 mobile nodes and analyze the network trajectory privacy gain under different strategies of the nodes seeking TPP cooperation.

1.6 Dissertation Outline

This research is one of the first attempts towards synthesizing of an integrated TPP model for mWSNs that supports in-network operations and trajectory context-aware services. The remainder of this thesis is organized as follows. We survey the related work in Chapter 2. The context-based trajectory privacy-aware routing protocol is presented in Chapter 3. We focus on addressing the internal attack detection in Chapter 4. Then we model and analyze nodes' behaviors in TPP activities in Chapter 5. Finally, we discuss the limitations of the current outcomes, identify future research directions and conclude this dissertation in Chapter 6.

CHAPTER 2

RELATED WORK

Along with the fast development of mWSNs, the TPP solutions are in urgent need. However, there are only a few studies have been done to address this issue. Many existing studies focus on privacy preservation in WSNs with the premise that all the network components are static. In other words, nodes lack mobility to be deployed in mWSNs. Studies in Location Based Services (LBS) domain are not suitable for mWSNs since sensors have much lower computational and communication capabilities. Multi-hop transmission between nodes and the sinks is also a challenge for adapting those solutions. TPP for Moving Objects Databases (MOD) publishing and data mining provides some promising methods to hide trajectories. However, mWSN applications require in-network TPP operations for real time data transmissions and process. Without the informational support of off-line databases, these methods cannot work well.

In this chapter, we provide a brief review of the literature in related domains, including stationary WSNs (sWSNs), LBS applications, MODs and mWSNs. Beyond the scope of the study of this topic, we find an interesting extension of TPP in GeoSocial Networks (GeoSNs). We provide a state-of-the-art review in this area as well.

2.1 Trajectory Privacy Preservation (TPP) in sWSNs

Privacy preservation has so far been studied in stationary WSNs (sWSNs) in the existing literature under the assumption that both the sensor nodes and the sink/base station are static. According to the taxonomy of privacy preservation techniques for WSNs in [LZDT09], there are two main types of privacy concerns, data-oriented and context-oriented concerns. Context-oriented privacy, named as trajectory privacy in this article, concerns the spatiotemporal information of an event. Particularly, TPP aims to protect the location and time of an event monitored by static nodes, or the location of the sink in sWSNs. In this article, the trajectory privacy of source nodes is of interest. Source node represents the node that generates the message once it observes active events in the network.

2.1.1 Location Privacy of the Source Node

The location of a source node is crucial in the case that attackers may catch or follow the moving object that monitored by the sWSNs, such as protected animals, soldiers, etc. The existing techniques fall into three categories: routing path randomization, dummy traffic injection, and cryptographic technique. We review these techniques as follows.

Routing Path Randomization

This main idea of this technique is to prevent attackers from tracking back the message source location by adding random routing hops en route to the sink. Researchers described the Panda-Hunter Game to study location privacy of the data source node [KZTO05]. In order to hide the location of the Panda from adversaries who try to capture the panda by back-tracing the routing path of the event message, Phantom random walk routing is proposed. This routing protocol works in the way that when the source sends an event message, the message is firstly unicasted randomly or in a directed random fashion for certain hops before it is flooded or routed to the sink. To improve the randomness of the routing paths, Greedy Random Walk (GROW) is proposed. GROW requires the sensor node to randomly route the message to one of its neighbors, which has not participated in the previous random walk [XSS06].

Li et al. studied a dynamic routing protocol which explicates the restricted random selection of intermediate nodes for message forwarding [LR10]. The authors suggest that in small scale WSNs, routing through single-intermediate nodes is efficient. Such intermediate nodes are randomly selected by the source and need to be away from the source with a minimum distance restriction. However, this method is not suitable for large-scale networks since attackers may deduce that the source is located within a circle region. The excessive long random routing paths cost unnecessary network resources as well. Therefore, the authors suggest angle-based and quadrant-based multi-intermediate nodes selection, where multiple intermediate nodes are selected based on their relative angle and distance to the source according to the sink. This method performs better in terms of message delivery ratio and privacy preservation. However, since the source node needs to predetermine the selected intermediate nodes and the quadrant reference frame, the computation could become a high cost for the source.

Dummy Traffic Injection

Dummy traffic injection is another widely used technique to preserve source location privacy in sWSNs. The main purpose is to perturb the network traffic to make the real and fake events undistinguishable. Generally, this technique includes fake message injection and fake source simulation.

Kamat et al. developed a simple scheme named Short-lived Fake Source Routing. It requires each node acting as a fake source by sending fake packets with a predetermined probability [KZTO05]. This method is effective to prevent local adversaries who can observe the traffic pattern in a small area. However, it is possible for global adversaries who have the information of the entire network transmission rate and traffic patter to identify the fake packets.

To address this problem, Shao et al. proposed the statistic-based dummy message injection [SYZC08]. In this method, all the nodes not only send fake messages with intervals following a certain distribution, but also the real events. In this way, the global adversary cannot distinguish the real events from fake messages. A similar algorithm was proposed in [MLW12], where sensor nodes send packets periodically and independently, including real or dummy packets.

The aforementioned dummy traffic injection may effectively hide the location of the real source, however, transmitting excessive dummy messages in WSNs consumes unnecessary energy and dramatically shorten the lifetime of sensor nodes. Additionally, the heavy dummy traffic load may lead to high channel collision. To address this issue, researchers proposed a Proxy-based Filtering Scheme (PFS) and a Tree-based Filtering Scheme (TFS) in which some sensors are selected as proxies to filter dummy messages before they reach the sink [YSZ$^+$08].

Fake source simulation is another method of dummy traffic injection. Mehta et.al applied source simulation in [MLW12]. This method generates simulated fake objects to confuse the attacker by producing similar traffic patterns as the real object.

Cryptographic Techniques

A straightforward technique is to use cryptographic techniques to encrypt users' identities and data. Although symmetric and Public Key Cryptography (PKC) have already been applied in resource-constrained environments, there are still concerns in the implementations. For example, symmetric cryptography requires complex protocols that suffer from other constraints. “Software realizations of PKC lead to relatively long duty cycles (operating intervals) which in turn require a significant amount of energy. Computation is negligible if the PKC is performed by power efficient hardware accelerators. In such cases the corresponding transmission power becomes much more significant and dedicated hardware is required" [PLP08].

Furthermore, only relying on cryptographic methods cannot resolve trajectory privacy issues in an effective and efficient way. Although the adversary does not have the knowledge of encrypted sensor data, data packets headers are usually left unencrypted for routing purposes where the source identity is revealed. In the case when data packet headers are also encrypted, the adversary still can obtain some public information of users, such as working and home addresses. Researchers have already shown how the adversary can crack users' encrypted/anonymized identities [MGKV06, CM07] with adequate background knowledge. An effective cryptographic method to prevent such encryption cracking to some extent is to encrypt or change the header or node identities every hop en route of message transmissions [PL11]. However, such technique faces the issue of synchronizing the encryption between nodes and the BS in implementations.

2.1.2 Temporal Privacy of the Source Node

Temporal information is the other element of trajectory information, besides location information. As reviewed, location privacy in sWSNs has drawn much attention from researchers. On the contrary, temporal privacy has not been formally defined.

Kamat et al. made temporal privacy the main focus for their work [KXTZ07]. As informally defined in [KXTZ07], "If we assume that the adversary stays at the sink, collects all the packets that are generated by a source, and tries to infer the creation times of these packets from the times when they are received, then the temporal privacy can be defined as the mutual information between the received time sequences and the creation time sequences." Given the hop count and average delay on each hop, by observing the message's arrival time, attackers are able to infer the message's generation time. It is possible to locate the event to a specific region, given speed estimate. By monitoring multiple messages, the attacker may be able to predict the next spot of the event. Kamat et al. developed the Rate-Controlled Adaptive Delaying (RCAD) algorithm to protect temporal privacy. RCAD uses intermediate nodes to buffer received packets. The processing delay distribution is determined by the incoming traffic rate and available buffer space at each node. RCAD employs buffer preemption mechanism to preemptively transmit the victim packets under buffer saturation.It serves a good tradeoff between temporal privacy and the buffer utilization. This method is very effective to prevent attackers from inferring message's generation time.

Despite the advances of the work noted in the previous two paragraphs, all of the work, except [PL11], are limited in that they are restricted by the fixed locations of the network components. That is, sensor nodes must have either predefined neighborhood relations or fixed routing roles/positions in the network. In mWSNs, both sensor nodes and the sink can be mobile entities; therefore, the aforementioned methods cannot be applied. In the following paragraphs, we discuss methods that consider users' mobility.

2.2 TPP for LBS Applications

Trajectory privacy protection methods in LBS inherently consider users' mobility in that they are implemented on smart mobile devices. In some of these methods, k-anonymity is applied, that is, a predetermined value of k is set, such that when a user submits a query, either the query is aggregated with queries from $k-1$ users or the query includes k Point Of Interests (POIs). For example, in reference [Swe02], researchers proposed a framework that requires a trusted third party or extra unit, known as an anonymizer/cloaking agent. Upon receiving the query from a mobile user, the anonymizer produces an "imprecise" service request and forwards it to the server. The imprecise result is produced by removing the user's ID and aggregating the query with $k-1$ queries from other users in a certain cloaking region. After the server responds to the queries, the anonymizer translates/filters the response and sends the "precise" results back to the user. Building on this framework, the following methods were implemented: In reference [MCA06], researchers employed a grid-based complete pyramid data structure that hierarchically decomposes the space into H levels to create cloaking regions. In reference [GKS07, DBS10], researchers generated cloaking regions by implementing a Hilbert space filling curve. In reference [YJHL08], researchers generated cloaking regions by computing the exact k Nearest Neighbors (kNN)in an incremental fashion. In reference [GL08], researchers proposed a personalized k-anonymity model that allows each mobile user to define and modify the anonymity level in both temporal and spatial dimensions. Last but not least, in reference [GKK$^+$08], researchers proposed a method that does not use an anonymizer, but requires an off-line phase that maps POIs' locations on the service regions into indexes. In this method, users encrypt the query with redundant information, and filter the redundancy in the response. A similar idea proposed in reference [KYS05] is to simply generate dummy queries to confuse attackers.

Regardless the efficiency, effectiveness and the practicalness of the above solutions, k-anonymity is vulnerable to background obtained attack, query sampling attack and query tracking attack. To prevent background knowledge attacks, researchers proposed the l-diversity principle, which requires that the sensitive attribute needs to contain at least l well-represented values [MGKV06]. To prevent query sampling attacks, researchers proposed the Reciprocity concept, which means a cloaking region not only contains at least k users, but the region is also shared by at least k of these users [KGMP07, CM07]. Finally, to prevent query tracking attacks, researches proposed to apply a memorization property for cloaked queries [CM07].

2.3 TPP in the Off-line Manner

In the LBS domain, trajectory privacy is considered as location sample privacy in most existing literature. A sequence of anonymized location samples' can be easily utilized by attackers to infer the trajectory. Very often, a third party, named anonymizer, is necessary for message translation, which is a heavy burden for sensor networks. To anonymize the trajectory, rather than only location samples, known as trajectory anonymization [Ghi09], other researchers studied this issue in off-line applications, such as MOD publishing and mining. One of the early work proposed to select a trusted location broker and set up privacy policy (e.g. the sensitivity and update frequency [GX04]). When the user's location is updated, the broker checks the update and compares it with the policy to check for violation. If any is found, it simply suppresses the update. Another researcher proposed to use the ID proxy to strip off the user's ID and forward the anonymous updates [HGXA07]. The ID proxy also cloaks k updates to achieve k-anonymity. The same author developed a path confusion approach in [HG05]. The main idea is to cut the trajectories into short segments and select some segments to apply the path confusion algorithm to mislead attackers. A more common approach for hiding trajectories is trajectory generalization. Briefly speaking, the trajectory needs to be transformed or distorted to some extent and grouped with other $k-1$ trajectories into a cluster [NAS08].

Hilbert curve [YBLW09] and Greedy Algorithm [MAA$^+$10, ABN08] were employed to find the $k-1$ nearest neighbors to form a cluster. Although trajectory privacy in database domain considers both users' mobility and spatiotemporal relations among samples, the premise is the availability of the whole trajectory information. Hence, existing solutions can only be applied to off-line data. Concentration in the development of in-network, online context-aware solutions has been neglected.

2.4 TPP in MWSNs

There are only a few works that studied trajectory privacy issues, specifically focusing on mWSNs. One proposed technique to preserve mobile nodes' trajectory privacy is to reduce the location resolution to meet the privacy requirement [XC09]. The authors consider an ad hoc network composed of sensor nodes deployed in a hostile environment. They developed a location cloaking technique which allows nodes to provide location information for services, yet prevents malicious locating. To be more specific, each node will recursively compute a cloaking box by broadcasting its current locating region partition P and counting the number of neighbors within P. P is divided into equal halves until the number of nodes within P meets the desired safety level, where P is set to be the cloaking box. This cloaking box is used as location information for reporting to service providers. To compute the cloaking box in the presence of node mobility, LEAVE, JOIN and MERGE messages need to be created to update the cloaking boxes for all the nodes in the corresponding partitions upon nodes' movements.

Another TPP method in mobile networks is that mobile nodes collectively change their pseudonyms in pre-defined regions called mix zones [BS03, BS04, FRF$^+$07, FSH09]. The mechanism works in the way that if multiple mobile nodes change their pseudonyms in the mix zone, the adversary cannot link new pseudonyms to the previous ones. Hence, the attacker cannot track a particular node. This method assumes the existence of a trusted middleware system, positioned between the underlying location system(s) and untrusted third-party applications. To fit in decentralized/distributed networks, researchers proposed to allow each mobile node to dynamically obtain mix zones [HMYS05, LSHP06]. The use of mix zones is appealing in resolving TPP issues in mWSNs. Nevertheless, the trust of the middleware system or cooperating nodes is prerequisite. How to evaluate such trust-based cooperation in non-cooperative networks remains an open problem.

2.5 TPP in GeoSNs

With the incredibly fast spreading of online social networks and the evolution of real-time geocoding and geotagging technology, GeoSN applications are becoming increasingly popular. As formally defined in Gambs et al.'s work, a GeoSN is "a web-based or mobile-based service that allow users to (1) construct a profile containing some of their geolocated data (along with additional information), (2) connect with other users of the system to share their geolocated data and (3) interact with the content provided by other users" [GHP11]. Popular GeoSNs, such as Foursquare, Twitter, Facebook Places, provide convenient interfaces for users to share their trajectory data with families, friends, and even the public. However, the trajectory privacy leakage during such sharing experience may lead to severe safety threats [Rob].

In this section, we focus on trajectory privacy concerns in the field of GeoSNs and highlight related literature. We first briefly introduce the findings in analysing privacy risk of the representative GeoSN applications. Then, we review the TPP mechanisms proposed in the existing literature. Finally, we summarize TPP challenges and open issues in GeoSNs in Chapter 6.

2.5.1 Trajectory Privacy Risk Analysis in GeoSNs

Most of GeoSN providers did not pay much attention on trajectory privacy issues in the first place. Gambs et al. provide a comparative privacy risk analysis of several existing GeoSNs, including Foursquare, Qype, La Ruche and Twitter [GHP11]. The authors compare these GeoSNs in terms of privacy issues in the criteria of registration information, real identities versus pseudonyms, information available to others and privacy settings. They conclude that most current GeoSNs do not directly integrated many privacy features.

Trajectory privacy risk in GeoSNs could be caused by three different resources, including 1) social network profiles, 2) location check-ins and queries, and 3) event posts and multimedia tagging.

Pontes et al. analyze the datasets collected from Foursquare, Google+ and Twitter. Without location inferring mechanisms, they found that the vast majority of Foursquare users provided valid home city locations in the corresponding venue attributes [PMV$^+$12]. Similar findings are presented in [PVA$^+$12], where a large number of users are found to exposure full addresses in their residential venue profiles. By analyzing the frequency and statistics of check-ins and geolocated data posts, these two groups of researchers also provide the location inferring mechanisms to infer possible residential addresses of GeoSN users. The results indicate that 78% of the analyzed users' home cities can be easily and correctly inferred within 50 kilometers of distance. Jin et al. focused on modeling and comparing the access control mechanisms for users' check-ins in the popular GeoSNs [JLJ12].

They conclude that there is no fine-grained access control mechanism for users' check-in information components. Co-location tagging also creates vulnerabilities to trajectory privacy. In [CCL13], Cheng et al. proposed a probabilistic framework overcoming the absence of granular location information in the posts to estimate a microblog (such as Twitter) user's location purely relying on her publicly available posts. Besides utilizing the geolocated data posts, the authors also developed a classifier to identify words in status updates with a local geographic scope to discover the association of certain words or phrases with certain locations. As a result, this framework provides k estimated cities for each user with a descending order of possibility. On average, 51% of randomly selected microblog users are located within 100 miles of their actual locations. These aforementioned findings indicate that trajectory privacy leakage and risk associated with GeoSN applications cannot be ignored and they are the critical issues which hinder the development of GeoSNs.

2.5.2 TPP Techniques in GeoSNs

Trajectory Obfuscation

The TPP issue in GeoSNs is more complex compared with in other LBS applications (we will analyze the challenges in Chapter 6). There are only a few works that have been proposed to address this issue. One of the major approaches is to apply the trajectory obfuscation technique, such as spatiotemporal cloaking and space transformation.

Cuellar et al. first formally defined a number of notions for evaluating a location obfuscation function [COR12]. The authors formalize the concept of indistinguishability of obfuscation functions, which requires that the user's actual location should be indistinguishable from a set a possible positions, e.g. an obfuscation region which includes the real location and noise. Indistinguishability needs to be satisfied under different scenarios: 1) when the attacker queries a user's position at regular intervals, the attacker should not be able to increase the precision of his knowledge on the real location up to the predefined threshold chosen by the user; 2) an adversary should not be able to determine the destination from the user's original position and location updates in route; 3) an adversary cannot deduce that the user revisits a certain location. Additionally, with such indistinguishable properties, the obfuscation region needs to be constant for all points contained within it. For more strict trajectory privacy protection, the obfuscation function needs to prevent attackers from determining users' current and past routes as well. Although this work defined an explicit concept of indistinguishability, it did not consider the impact of social relationships and interactions among users on the privacy leakage, which is highly possible to be utilized by the adversary to infer users' locations.

On the contrary, Freni et al. attempted to deploy a centralized trusted entity to process a user's original resource, which may involve multiple users before publishing it to the GeoSN to insure it complies involved users' privacy preferences [FRVM$^+$10]. The authors introduce the notion of Minimal Uncertainty Region (MUR) as a spatiotemporal region for which an adversary cannot infer any internal point as the users' actual location. The location cloaking system architecture is presented in Figure 2.1. The pre-processing module is designed to retrieve the privacy preferences of all the involved users and the corresponding MUR. The cloaking module applies spatial generalization and/or temporal generalization (depending on the service attribute whether it is time sensitive) algorithms to generalize the pre-processed *srg* if the disclosure of *srg* introduces privacy violations. Then, the result is processed by the publisher module, where the users' absence privacy preferences will be complied by postponing the publication of the resources, if necessary. This step

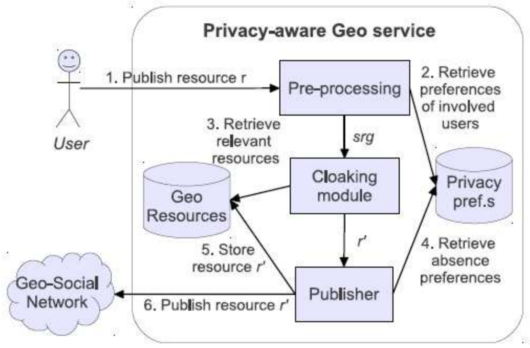


Figure 2.1: System architecture in [FRVM+10]

is to enforce users' trajectory privacy under the scenario that an attacker may infer a user's absence by utilizing the current MUR and maximum velocity. This system has great potential to be adapted on top of the current GeoSN architectures. However, the centralized trusted clocking entity could be an issue to be designated.

Puttaswamy et al. applied a user-specific, distance-preserving space transformation algorithm, named *LocX*, under the assumption that the GeoSN servers or other intermediaries are untrusted. This algorithm is intended to deal with point queries and nearest-neighbor queries [PWS+12]. The main idea is that users share secret keys with their friends and use these keys to encrypt location data and store them in the proxy to process location queries. The encryption mapping is split into two pairs: a mapping from the transformed location to an encrypted index, named L2I, and a mapping from the index to the encrypted location data, named I2D. These two pairs are stored in two modules/servers in the proxy. When a user submits a query referencing to a certain POI, she submits the query in the form of transformed location data encrypted by symmetric keys to the proxy. The proxy returns all the pairs of L2I that are in the user's query. Then, the user decrypts the index and queries the I2D pair. The proxy then returns the encrypted location data pair with the corresponding index. Although the authors claimed that *LocX* can run on mobile devices with low computation and communication cost, how the computation power consumption is evaluated is not presented that , which is an important concern of mobile users. Moreover, the symmetric key distribution and rekeying management could be difficult to develop in practical GeoSNs.

There is another k-anonymity-based location cloaking algorithm proposed by Masoumzadeh et al., focusing on anonymizing GeoSN Datasets [MJ11]. In this work, the authors tackle the issue that the location information revealed by social connections may assist an attacker in re-identifying a user. In order to guarantee the privacy of users' locations and identifies, the GeoSN datasets should satisfy L_k^2-anonymity, meaning that there are at least $k-1$ other users assigned the same location information for each user, and each user's adjacent users in the social network are also assigned the same location information as the $k-1$ other users' adjacent users. The algorithm starts with each user as a separate cluster, centering at her location. In each iteration, the distance between every pair of clusters is computed in order to select the minimum pair to form a new cluster. The iteration stops until the cloaking region satisfies L_k^2-anonymity. The possible difficulty to implement this algorithm is that, in some cases, the uniqueness of a user may result in impossibility to satisfy L_k^2-anonymity, such as a user is travelling out of town. Additionally, this work focuses on the data sets anonymization. There are still issues to adopt this algorithm into in-network computing, such as how to retrieve the location of other users to compute the distance. Nevertheless, the concept of L_k^2-anonymity has great potential to resolve TPP in GeoSNs, where the social relationship is playing an important role.

Context Analysis

In GeoSN applications, context analysis is particularly an important tool for security and privacy protection due to the rich semantic contents during social interactions. Riboni et al. proposed an initial investigation to prevent users' POI preferences from being learned by other users, which leads to privacy leakage [RB12]. The main idea is to apply PINQ [McS09] query engine to enforce ϵ-differential privacy by adding random noise to extract statistics about personal preferences for POIs. To protect trajectory privacy, users submit their queries with a spatial granule in which they are currently located instead of the accurate location. Unfortunately, as an initial investigation, there are no details of how to define the granularity level of the granule to guarantee both trajectory privacy and service accuracy.

Jagtap et al. proposed a context-aware access control mechanism [JJFZ11]. They adopt Web Ontology Language to capture users' characteristics, including location and surroundings, the presence of other people and devices, feeds from social networking systems they use and the inferred activities in which they are engaged. The reasoning engine is used to handle queries and performs reasoning for access control decisions. These two components support the privacy control module to enforce access control of the query to protected information. The privacy control module is embedded at both client and server sides to check the privilege to access protected resources for peer-to-peer queries and peer-to-server queries. There are several concerns to apply this mechanism: 1) the server might not be trustworthy; 2) the energy consumption of the client is not neglectful; and 3) since the privacy control module collects all the user's social profiles and preferences, it may need extra protection for security reasons.

Identity and Location Encryption

Cryptography technique is a fundamental and direct solution for security and privacy protection. In GeoSNs, symmetric key encryption and hashing methods are commonly used in TPP in proximity services. Since proximity service is a particular application which mainly involves location coordinates manipulation and distance computation, the existing methods lack generality to be applied in other GeoSN applications. Hence, we limit the number of related literature here. More explicit materials are surveyed in [MFB$^+$10, AEM$^+$07].

Mascetti et al. proposed two protocols, named *C-Hide&Seek* and *C-Hide&Hash*, to protect a user's location privacy from untrusted Service Providers (SPs) and other users when she submits a proximity service request. In the *C-Hide&Seek* protocol, the SP replies with a message containing the latest encrypted location updates of each friend of the requester. Since the encryption uses symmetric key, the requester can decrypt the message using the keys that shared with her friends. To prevent attackers from inferring that the target is crossing the boundary between two granules by monitoring the time stamp of the location updates, the location is only updated after a certain interval and identified by an interval index. The major difference of *C-Hide&Hash* with respect to *C-Hide&Seek* is that the requester provides a set of granules to check if any of her friend's location falls in this set. In this case, other users' locating granules are protected from the requester. This work has considered untrusted SPs and curious users, and provided the complete privacy preservation protocols in proximity service. The guaranteed location privacy has been theoretically approved. However, how to properly define the update interval still remains a question. Additionally, these protocol might not be applicable in time sensitive services due to the improper update interval. Moreover, as most other methods using symmetric keys, the key distribution issue needs to be addressed before practical implementations. Li et al. also applied symmetric key hashing to transform location information during proximity service querying under similar system assumptions [LHX12]. Their work is focusing on increasing the proximity detection accuracy by using optimal grid overlay and multi-level grids.

In [PMM], Provost et al. suggest to used one-to-one or many-to-one hashing to hash users' identities and location information. This work focuses on utilizing GeoSN behavior similarity for advertisement targeting like-minded individuals. Carbunar et al. designed the mechanisms that construct users' location centric profiles in a private and correct manner [CRRB12]. In this work, the authors bring attention to the issue of dishonesty in GeoSN applications, such as Yelp, Foursquare, where users may create incorrect check-ins to gain benefits, as well as the privacy violation issue when venues collect users' location centric profiles. The main idea is to install a device at each venue to initiate a challenge and authorize a timestamped token encrypted by its secret key to check-in users. This step is used to prove a user's physical presence at the venue. Then, the venue collects the user's profile by increasing the counter by 1 on a certain dimension and the corresponding range where user's profile value falls in. Furthermore, the statistics of the collected profiles is published. Although these two studies lack details of TPP mechanisms, they are inspiring studies that consider the TPP issue in the inherent design of GeoSNs.

In Chapter 6, we will conclude the characteristics of trajectory privacy and discuss the challenges in GeoSNs.

2.6 Summary

In this chapter, we have reviewed the existing main techniques to protect trajectory privacy in the domains of WSNs, LBS applications, MODs, and GeoSNs. The previous work is of high inspiration and have greatly enlightened our work. In the following chapters, we will present our results so far and provide promising future extensions of this topic.

CHAPTER 3

TRAJECTORY PRIVACY-AWARE ROUTING PROTOCOL

In this chapter, we intend to resolve the TPP issue under the common external attack, i.e. eavesdropping attack. The aim of this work is to develop a mechanism which hides the trajectory of the data source node, denoted as the target node, on the fly with considering nodes' mobility in WSNs. In this chapter, "on the fly" is defined as a node hiding its trajectory while undergoing data transmissions. In other words, it is a distributed method that protects trajectory privacy prior to data reaching the centralized database/base station.

3.1 System Model

The considered network is the mobile sensor network with stationary backbone infrastructures for realistic applications.

- Base Station (BS): The BS is the sink receiving data from APs. It is the center for data analysis and management.
- Access Points (APs): Multiple APs are interconnected through a backbone network and connected to the BS. The network area is partitioned into N distinct subareas $SUB = \{Sub_1, Sub_2, ..., Sub_N\}$. The AP is located at the center of a subarea Sub_i and serves near nodes within the entire subarea, named correspondingly AP_i. Nodes report sensed data through one or multiple hops to reach one of the APs. Then APs forwards the data to the BS. We assume the physical location distance is equivalent to routing path distance during data transmissions between nodes and APs.
- Mobile nodes: Mobile nodes are homogenous sensors with diverse trajectory patterns and moving velocities. Nodes send data to the nearby APs; include

both sensing data and trajectory information for system localization purposes. The mobility is provides by carriers, named users, such as human bodies, cellular phones, and endangered species. We consider $U = \{u_1, u_2, ..., u_k\}$ a set of K nodes moving within SUB during a time period $T = \{t_1, t_2, ..., t_T\}$. The accurate trajectory of node u is T_{ru}, defined in the next section. The real trajectory set of all K nodes is $R = \{T_{r1}, T_{r2}, ..., T_{rk}\}$. The approximate trajectory set of node u is $r_u = \{Sub_{ut_1},\ Sub_{ut_2}, ..., Sub_{ut_T}\}$ during T.

3.2 Adversarial Model

We developed this project with considering the passive attack. Inspired by the framework in [STLBH11], we characterize the adversary by his knowledge and attack.

- Knowledge: The adversary has adequate computation capability, energy and memory for data storage. The adversary does not have the knowledge of encrypted sensor data. However, data packets headers are usually left unencrypted for routing purposes where the source identity is revealed. Moreover, the adversary could easily obtain some public information of users, such as working and home addresses. Therefore, the adversary could even crack encrypted identities. The adversary is also assumed to know some system parameters, include the network partition SUB and the user set U. We define the whole knowledge of the adversary AK.
- Attack: The adversary behaves in honest-but-curious model [Gol04]. He deploys multiple stationary nodes to eavesdrop wireless communications in the network. The preferred locations of these eavesdroppers are near APs, where sensor data streams are aggregated. Since most routing algorithms

> are shortest-path oriented, the adversary can deduce that the source node is near a certain AP or within a certain subarea when he "hears" the corresponding packets. The adversary monitors traffics over the entire network. In other words, this is a global adversary. The adversary does not launch active attacks to interfere with proper functions of the network. Moreover, node compromise attack is excluded. The objective of the adversary is to find out the whole trajectory of nodes' tracking attack; or to localize a node at a given time instant-localization attack. This node is the target node. We consider data source nodes as target nodes in this project. The trajectory set of all K users observed by the adversary is $R' = \{T'_{r1}, T'_{r2}, ..., T'_{rk}\}$. The approximate trajectory set of node u in time period T observed by the adversary is $r'_u = \{Sub'_{ut'_1}, Sub'_{ut'_2}, ..., Sub'_{ut'_T}\}$. All observations of the adversary is denoted as O.

The eavesdropping attack is a simple and effective trajectory privacy attack. Privacy attackers behave in honest-but-curious model [Gol04]. They deploy multiple stationary sensor nodes called eavesdroppers around each AP. Since most routing algorithms are shortest-path oriented, attackers can deduce that the source node is near a certain AP or within a certain subarea when they "hear" the corresponding packets. Even though messages are encrypted in some highly-secured sensor networks, packets headers are usually left unencrypted for routing purposes where the source identification is revealed. Moreover, even if headers are encrypted, attackers could crack the certain pattern of header encryption of each node after a period of packets logging. With the revealed header information, attackers can now estimate the location of a specific mobile node by eavesdropping near APs. After a time span, the eavesdropped records near all APs could provide attackers the fundamentals to analyze the trajectory pattern of the user.

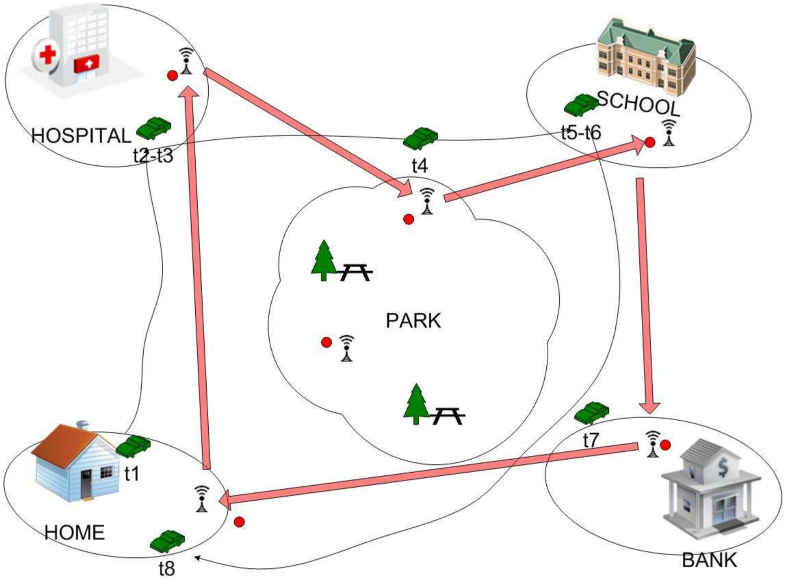


Figure 3.1: Trajectory privacy attack - eavesdropping attack illustration

Figure 3.1 is an illustration of trajectory privacy attack by eavesdropping. The user carries a wearable sensor moving in the resident area. The attacker have deployed several eavesdroppers, which are shown as red dots, at APs. The black curve represents the real trajectory of the user, while the red straight line represents the obtained trajectory by the attacker. The attacker may even perceive that the user visited the hospital and the school, and passed by the park and the bank without a stop, by comparing the amount of packets heard at different APs. Without accessing the centralized database and message content, they still can obtain the private trajectory information, as shown in Table 3.1.

Table 3.1: Trajectory obtained by eavesdroppers

T	r_u	r'_u
$t1$	HOME	HOME
$t2 - t3$	HOSPITAL	HOSPITAL
$t4$	TRAVEL	PARK
$t5 - t6$	SCHOOL	SCHOOL
$t7$	TRAVEL	BANK
$t8$	HOME	HOME

3.3 Objectives and Evaluations

With the knowledge AK and observation O, the adversary is trying to reconstruct R and r correctly. If the adversary is able to localize users to specific subareas in time period T, he will also successfully track users. Therefore, our objective is to prevent the adversary from reconstructing r for each user. The evaluations metrics are developed based on adversarial evaluation metrics proposed in [STLBH11]. We define our evaluation metrics in the following.

Let $x'_{ut}, u \in U$ and $t \in T$, be the estimate of the subarea that node u was located in at time t. For all target nodes during T, all estimates based on the observation O comply with the probability distribution $P(X = x'_{ut}|O), x'_{ut} \in SUB$. The real subarea that u was located in is denoted as $x_{ut} \in SUB$. The incorrectness E, which is the adversary's expected estimation error, can be quantified using the expected distance between x_{ut} and x'_{ut} with probability distribution $P(X|O)$. For example, this distance can be the distance between AP_{ut} and AP'_{ut}. Assuming the coordinates of these APs in 2D space are (a_{ut}, b_{ut}) and (a'_{ut}, b'_{ut}), respectively, the incorrectness can be computed as follows:

$$E = \sum_{x'_{ut}} P(X = x'_{ut}|O)\sqrt{(a_{ut} - a'_{ut})^2 + (b_{ut} - b'_{ut})^2}. \tag{3.1}$$

The incorrectness E of the adversary is the metric that evaluates the trajectory privacy of the system. The higher E is, the higher the trajectory privacy is.

Recall that the adversary in our model is a global adversary. Therefore, it is possible that he could estimate possible trajectories by analyzing the traffic distribution in the entire network. For an instance, if packets from target nodes are "heard" by certain eavesdroppers with obvious higher probability, the adversary will be able to learn that the real trajectories have special connections with the these subareas. The estimated trajectories could be constrained by the special connections. We use entropy of the distribution $P(X = x'_{ut}|O), x'_{ut} \in SUB$ to evaluate the performance of our algorithms against the global adversary. The entropy is computed by:

$$H = -\sum_{x'_{ut}} P(X = x'_{ut}|O) \log P(X = x'_{ut}|O). \tag{3.2}$$

The higher the entropy is, the more uniform the distribution is. It gives the lower probability to the adversary to find out the special connections.

3.4 Proposed Method

In this section, we first present a definition of trajectory, which is often used in MOD literature [GS05]. For simplicity, we assume that nodes are moving in 2D space. We have the following:

Definition 3.4.1 *A trajectory T_r of a moving node is a polyline in the three dimensional space, where two dimensions refer to space and the third dimension to time. It is represented as a sequence of points $< (x_1, y_1, t_1), (x_i, y_i, t_i), (x_n, y_n, t_n) >$ with $t_1 < t_i < t_n$.*

Each point (x_i, y_i, t_i) in the sequence represents the 2D location (x_i, y_i) of the node, at time t_i. Observe that the trajectory above is defined in MOD, where all information is collected into the offline database for trajectory analysis. Moreover, the difference in initial points between two trajectories is often ignored during analysis, such as trajectory pattern mining, trajectory similarity computing, etc. In order to hide the node's trajectory on the fly in mWSNs, we need to analyze the dynamic trajectory distance between two nodes on the network.

The dynamic trajectory distance represents the irrelevance of two trajectories in terms of 2D location and velocity at a specific time. At time t, given the 2D location (x_{it},y_{it}) and velocity vector, $\overrightarrow{i_t}$, of node i , and the 2D location (x_{jt}, y_{jt}) and velocity vector, $\overrightarrow{j_t}$ of node j, the location distance between i and j

$$D_l(i,j,t) = \frac{\sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2}}{r_{max}}, \tag{3.3}$$

where r_{max} is the maximum transmission distance of a node in the network. The velocity direction distance is derived from the cosine similarity of two vectors, which is

$$D_v(i,j,t) = 1 - Sim(\overrightarrow{i_t}, \overrightarrow{j_t}) = 1 - \cos\lambda = 1 - \frac{\overrightarrow{i_t} \cdot \overrightarrow{j_t}}{||\overrightarrow{i_t}|| \cdot ||\overrightarrow{j_t}||}, \tag{3.4}$$

where λ is the relative angle between $\overrightarrow{i_t}$ and $\overrightarrow{j_t}$. The unified speed of node j

$$S(j,t) = \frac{||\overrightarrow{j_t}||}{S_{max}}, \tag{3.5}$$

where S_{max} is the possible maximum speed of sensor nodes. Finally, we have the following:

Definition 3.4.2 *The Dynamic Trajectory Distance (DTD) represents the irrelevance of two trajectories in terms of 2D location and velocity at a specific time. Given $D_l(i,j,t)$ and $D_v(i,j,t)$ between the target node i and its neighbor node j, and $S(j,t)$ of the neighbor j, the dynamic trajectory distance*

$$D_T = w_1 D_l(i,j,t) + w_2 D_v(i,j,t) + w_3 S(j,t), \tag{3.6}$$

where $W(w_1, w_2, w_3)$ is a predefined weighted vector and $w_1 + w_2 + w_3 = 1$.

3.4.1 Basic Trajectory Privacy Preservation

From Figure 3.1, we observe the following: 1. without breaking the network security, privacy attackers could estimate the target node's trajectory by simply eavesdropping messages en route at low cost; 2. attackers could obtain trajectory information during message transmissions in-network without accessing the centralized database/base station; 3. conventional shortest-path oriented routing protocols (for simplicity, we will use conventional routing protocols) create the possibility for attackers to deduce trajectory information.

Based on the above observation, we propose to use Trajectory Privacy-aware routing (TPriv) to hide the target node's trajectory on the fly. With TPriv implementation, the target node takes the advantages of nodes mobility and routes messages to non-local APs in order to mislead attackers. They are not able to deduce the real trajectory through passive attack on the network.

TPriv has two different phases for message routing: conventional routing phase and privacy-aware routing phase. For the message requesting high priority in transmission delay, such as emergency sent by patients to ask instant help from doctors, the conventional routing phase is only needed for the data transmission. For regular data, which is collected by the base station periodically from nodes, the privacy-aware routing phase is requested, followed by the conventional routing phase.

In the conventional routing phase, packets are routed according to the shortest-path routing algorithm. In sensor networks, this algorithm can be pro-active, reactive, geographical, power-aware, hybrid routing algorithm, etc. Due to the high dynamic network topology, on-demand oriented routing algorithms are recommended. However, details of the algorithms are out of the scope of this work. We contribute to propose the privacy-aware routing phase to address TPP issues specifically. The objective of this phase is intuitive and effective: Through forwarding the message to non-local APs, it prevents the trajectory privacy eavesdropping attack at network APs. To achieve this objective, the target node needs to select the next hop under the principle that among all the neighboring nodes the candidate has the highest probability to forward the packet to a non-local AP.

To meet this principle, two difficulties need to be overcome as an on-line distributed design: The target node does not have the trajectory information of its neighboring nodes as the background knowledge to select the next hop; assuming the first problem is overcome, the trajectory privacy of neighboring nodes can be breached by the target node.

TPriv overcomes both these difficulties and meets the design principle. The target node first collects limited information from its neighbors before data transmission in query-and-reply fashion. Then each neighbor is evaluated in terms of its dynamic trajectory distance to the target node. Finally, sensed data is routed to a non-local AP through the selected next hop. Note that, in order to prevent the target node to breach other nodes' trajectory privacy, one-time pad Virtual Identification (VID) is employed during trajectory query and reply. In the following, we will explain how the privacy-aware routing phase works in the basic TPriv in detail.

Dynamic Trajectory Information Collection

During this stage, two types of VID will be used: Query VID (QVID) for query messages and Reply VID (RVID) for reply messages. Both are one-time pads that generated by the node itself. Each VID is valid for a certain period of time. The life time for QVID $LT_Q = \Delta T$, while $LT_R = 2\Delta T$ for RVID, where ΔT is the message transmission interval. LT_Q is designed to prevent the target node from sending continuous queries during one message transmission interval which may lead to a waste in bandwidth or even network congestion. LT_R is designed to prevent the target

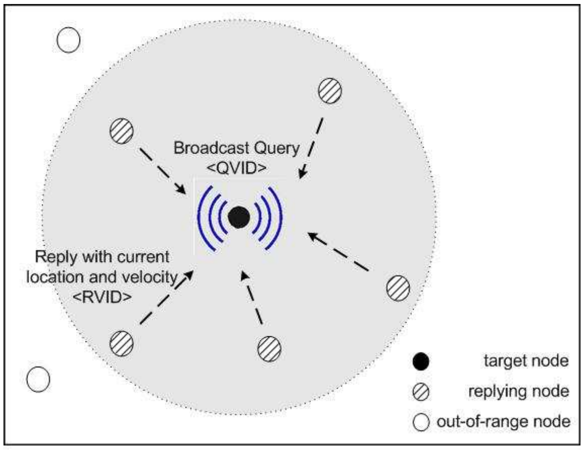


Figure 3.2: Query-and-reply under VID

node from selecting one neighboring node as the next hop continuously. The VID is not used for data transmissions. There is no need to set up the third party [CZBP06] for virtual name translation/mapping between nodes and the base station, which gives extra burden to the network.

Before the target node transmits data, it broadcasts a dummy query message under its QVID. Each one-hop neighbor replies this dummy query under its RVID. The reply contains the dynamic trajectory information of each neighbor, including current location and velocity. During time ΔT, any duplicate query message with the same QVID will be ignored by neighbors. After the target node receives the reply, it temporarily stores the dynamic trajectory information with the corresponding RVID except the entry which has the same RVID as the selected next hop during the last data transmission. This is for preventing the curious neighboring node from acting as a beacon node to track the target node [YBLW09]. The query-and-reply operation is shown in Figure 3.2. The shadow area indicates the radio coverage area of the target node.

DTD Computation and the Next-hop Node Selection

At this stage, the target node selects a next-hop node for data transmissions. Next-hop nodes could be more than one, which may provide better performance in terms of privacy. However, there is a tradeoff between privacy and power consumption. So far, we have considered one next-hop node only. The target node computes the D_T to each neighbor according to Equation 3.6, respectively. Since D_T represents the irrelevance of two trajectories in terms of the 2D location and velocity at a specific time, the neighbor, which has the greatest D_T to the target, is selected as the next hop at the time of data transmissions. Assuming D_T computation time is negligible, the candidate remains staying at the same location and with the same velocity.

After the selected next-hop node receives the message, it resets the hop count entry in the packet header to be 0 and starts the conventional routing phase. With the implementation of TPriv, the target node routes each regular message to a non-local AP with a certain probability. It misleads the passive privacy attacker at each time of data transmission. However, the probability of reaching non-local APs with one-hop privacy-aware routing phase is undesirable in large-scale networks. Moreover, the frequency of each non-local AP to be on the route may disclose the trajectory of the target node to global attackers after a sequence of time. Therefore, we propose the improved TPriv, named as Secondary TPriv, to resolve these problems. The above method is named as Basic TPriv (BTPriv).

3.4.2 Secondary TPriv(STPriv)

STPriv improves the basic TPriv in two aspects to resolve the aforementioned problems of BTPriv: extend one-hop to M-hop privacy-aware routing phase and randomize the probability of routing in privacy-aware routing fashion.

M-hop Privacy-aware Routing Phase

In large-scale networks, the probability that packets are routed to different subareas is decreased. Basic TPriv only provides a very limited range for finding non-local APs. Therefore, we invent the m-hop privacy-aware routing phase, which means the following m next-hops will be selected in a privacy-aware fashion. M is a predefined parameter, depending on the network deployment. Here we take m equal to 2 as an example. The following next-hop selection process is the same. Upon receiving packets from the target node, the intermediate node takes similar procedures as the target node in selecting the second next-hop node, except for message filtering and the DTD computation.

To avoid a routing loop, in the case that any intermediate node routes the message back or closer to the previous-hop node, the intermediate node broadcasts the query message with the same QVID as the previous-hop node. Any neighbor who receives the duplicate query message consecutively ignores this query. Therefore, the next-hop selection is constrained to the neighbors, which are at least two hops away from the target node at the start of this data transmission.

For the DTD computation,in order to forward packets in further distance, the second-hop candidate with greater D_l, higher speed and smaller D_v, is preferred. This is because the current intermediate node already has very different velocity direction from the target node after the basic privacy-aware routing phase. The DTD is computed as follows:

$$D'_T = w_1 D_l(i,j,t) + w_2(1 - D_v(i,j,t)) + w_3 S(j,t) \tag{3.7}$$

Note that, the greater M does not indicate better performance. M determines the tradeoff between privacy level and transmission delay.

Table 3.2: Pseudo Code of the next hop selection algorithm in STPriv

```
Input:  M, P,W[3], m, the dynamic trajectory information of the neighbors
Output: RVID of the next-hop node
$a$ = random number between 0 and 1
IF ($a <= P$ and $m == 0$) OR ($m == M$):
   Use the conventional routing algorithm for data transmission
   $m = 0$
ELSE:
   $RVID = 0$
   $\_max = 0$
   FOR each valid neighbor:
      IF ($m == 0$):
         Compute the DTD according to Eq 3.3-3.6
      ELSE:
         Compute the DTD according to Eq 3.3-3.5 & 3.7
      IF($\_max <= \_DT$):
         $\_max = \_DT$
         RVID = Current Neighbors ID
Return RVID
```

Randomized probability of TPriv

BTPriv requests that all the regular messages are routed in the privacy-aware phase first. Therefore, the target node which always moves between several specific subareas will hardly transmit data through the corresponding local APs. For the global attacker who is able to obtain statistics regarding APs en route, it will be obvious that the target node is moving within certain subareas. To address this problem, STPriv allows the target node to choose the conventional routing phase directly with a certain probability P. It depends on the number of subareas in the entire network field.

Our proposed STPriv algorithm is presented in Table 3.2. The velocity of each node is its average velocity, defined as the change in position divided by elapsed time. Each node calculates its velocity and stores it locally. The location of each node is also available from the self-attached GPS unit. The target node obtains its neighbors' location and velocity through query-and-reply, shown in Figure 3.2. Table 3.2 is the next-hop selection procedure with M-hop STPriv implementation. The neighbor which has the maximum D_T to the target node is selected. Network administrators or sensor users predefine the system parameters, including the total number of hops in the privacy-aware routing phase, M, the probability to choose conventional routing algorithms directly, P, and the weighted vector, W. The maximum transmission range and the velocity of mobile nodes depend on the network application and are also given by network administrators. Current number of hops for privacy routing phase m is read from the packet header.

3.5 Simulation Results and Analysis

In this section, we present the simulation data to demonstrate the effectiveness of our design. The simulations are implemented in both small-scale and city-wise networks to evaluate the generality of the proposed routing protocols.

3.5.1 Simulation Setting

The proposed algorithms were implemented using Python and Matlab on Windows XP on a 2.13 GHz Intel Core 2 CPU equipped with 2GB of main memory. Experiments are set up in different scales of networks to simulate different applications. We consider a square area divided into $40X40m^2$ subareas to represent institution network fields. The network scale varies among containing 4, 9 and 16 subareas. Nodes are moving in the manner of random waypoint. Nodes are categorized into three types according to their trajectory patterns: restricted nodes (moving within one subarea), repeating node (repeating certain trajectory in some subareas), and traversing nodes (randomly traversing the entire network). Nodes are deployed by given random locations in the network grid and the speed between $0.2 - 22$ mph. We also use a $200X200m^2$ area within DHDN/3-degree Gauss-Kruger zone 2 (EPSG code: 31466) to represent part of city-wise network areas. It contains 9 subareas of interests. The trajectory data of nodes are generated by using the Random Street model of BonnMotion [Bon]. The weighted vector, $W = [0.5, 0.2, 0.3]$.The maximum transmission range of each node is set to be 20 m. It is referenced to the average transmission range for reliable connection in our real experiments on MEMSIC sensors equipped with MTS420 boards. We conducted 20 independent rounds of simulations, for each set of parameters. In one round, each data source node transmits 100 packets. Note that, since we focus on the application and the network layers, some nodes in the network are set to be dummy nodes, which only forward packets, to avoid channel collision.

3.5.2 Results and Analysis

Recall the evaluation metrics we defined in section 3.3, we firstly evaluate the performance of our design by computing attackers' incorrectness value E according to 3.1. The method to define the distance between x_{ut} and x'_{ut} could vary in different applications and spaces. In order to have straight forward evaluation results, here we define the distance between x_{ut} and x'_{ut} is 0 if and only if $x_{ut} = x'_{ut}$. Otherwise, the distance is 1. Then, we have

$$E = 1 - P(x_{ut}|O). \tag{3.8}$$

Figure 3.3 is the box-and-whisker plot by implementing BTPriv. The data were collected by considering all the repeating and traversing as target nodes. The mean of E increases along with the average node density and the network scale. BTPriv has less time and power consumption compared to STPriv in terms of the number of hops for the privacy-aware routing phase. After the node density reaches to the certain limit, E slightly increases. With the improvements from M-hop STPriv, the value of E dramatically increases. We simulated 2 to 5 hops of STPriv and compared the results with the performance of BTPriv. The result (mean ± standard deviation) is shown in Figure 3.4. We use line plot to represent discrete data in order to get better illustration. The node density in this figure is 1 node/100 m^2, as well as in Figure 3.7 and Figure 3.6. It is observed that after M is greater than 3, the performance of STPriv keeps stable. Note that, we have defined p close to 1/2N for choosing conventional routing algorithm. Even though, the stable point still reaches 81% in 16-subarea network. With 5-hop STPriv implementation, E is as high as 92.6%. In other words, the adversary fails in locating target nodes for 92.6% of the time. Figure 3.5 and 3.6 show the performance of STPriv in the city-wise network. The average node speed used in Figure 3.5 is 20 mph (as well as in Figure 3.7). Similar to small-scale networks, STPriv performs better when M increases. STPriv also has stable performance when node density and average node speed vary.

Next, we evaluate the performance of preventing global adversaries by computing the entropy H according to 3.2. The ideal distribution $P(X|O)$ is uniform distribution. For example, in a 4-sub network, $P(X = x'_{ut}|O) = 0.25, \forall x'_{ut} \in SUB$ and the normalized entropy $H = 1$. Figure 3.7 shows the simulation results in small-scale networks. For repeating nodes and traversing nodes (Rp & T), BTPriv offers comparable performance as the ideal distribution. H is as high as 0.999. In Figure 3.7b, STPriv also gives high H value with small M in city-wise networks. This is due to the randomness of nodes movements. However, more hops in the privacy-aware routing phase are necessary for hiding trajectory of restricted nodes (R). The

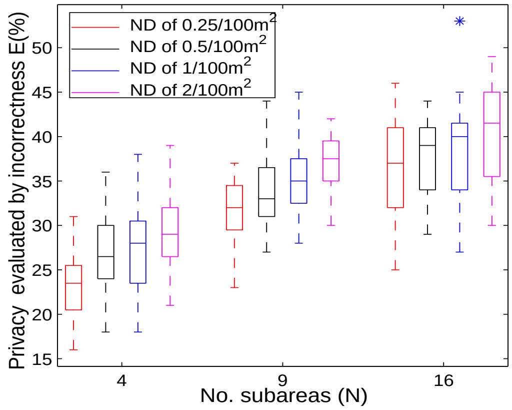


Figure 3.3: Performance of BTPriv in different scales of networks

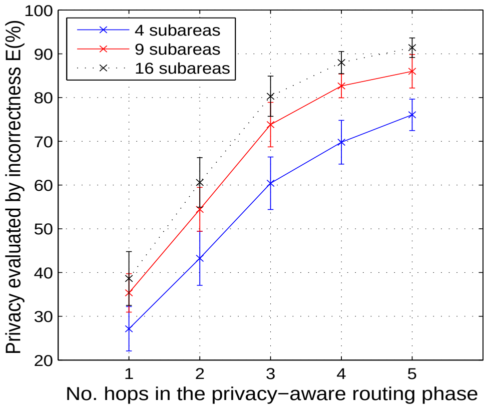


Figure 3.4: Performance improvement of STPriv

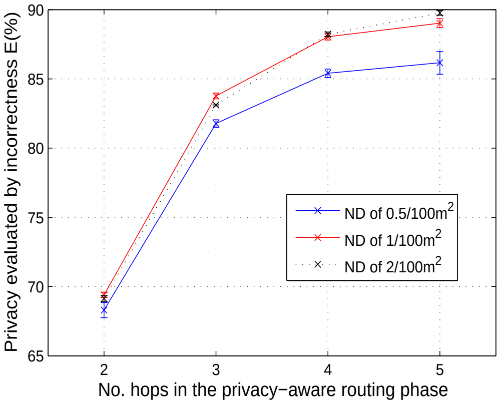


Figure 3.5: Node density impact on STPriv performance in city-wise networks

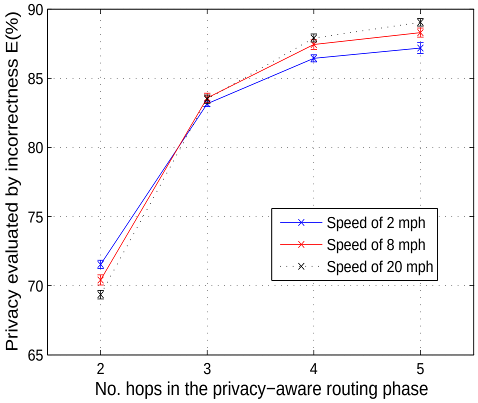
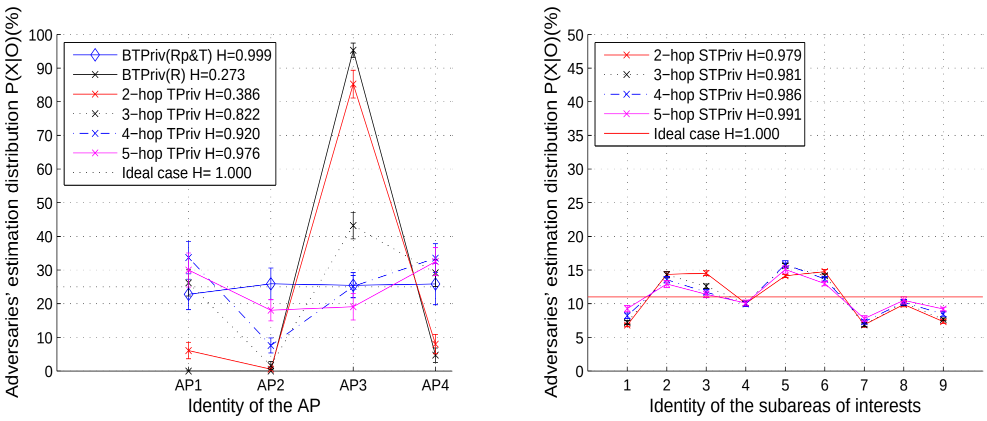


Figure 3.6: Node speed impact on STPriv performance in city-wise networks



(a) small-scale networks

(b) city-wise networks

Figure 3.7: Performance of preventing global adversaries

corresponding normalized entropy is computed and also shown in Figure 3.7. With greater M, STPriv is also effective to prevent global adversaries in extreme cases.

3.6 Summary

In this chapter, we investigated the eavesdropping attack in mWSNs. We utilized the trajectory dissimilarity and developed the context-based trajectory privacy-aware routing protocol to resolve the privacy issue of conventional shortest-path routing algorithms. We showed the effectiveness of the proposed simulated BTPriv and STPriv algorithms by simulations in both city-wise and small-scale mWSNs. The limitations and future directions will be discussed in Chapter 6.

CHAPTER 4

UNPREDICTABLE SOFTWARE-BASED ATTESTATION SOLUTION

In Chapter 3, we investigate the trajectory privacy-aware routing protocol to counter the external eavesdropping attack. The proposed TPriv algorithm can mislead attackers about target nodes' locations and bypass suspicious followers. However, in the presence of node compromise attack, how to protect trajectory privacy from compromised nodes, which have the authority to access the network secret or private information, becomes critical in TPP.

Sensor nodes have inherent constraints such as limited energy, processing capability and memory space. Additionally, the mobility of sensor nodes offer more applications in adversarial environments. Hence, mobile sensors are more vulnerable to node compromise attack than wired or high-powered mobile computing devices. Hartung et al. demonstrated the process to compromise sensor hardware [HBH05]. Without any prevention solutions or hardware protection, sensor nodes are easily tampered with and system-critical information can be easily obtained.

Solutions to address node compromise focus on detection and increasing network resilience. The latter is under the condition that the detection of compromised nodes has occurred. Most of the previous work is based on misbehavior detection or on attestation mechanisms [KSE08]. A voting mechanism is designed to detect misbehaving nodes based on a neighboring node monitoring mechanism, in which all nodes within a relative proximity coordinate to determine if a node is compromised [LZL$^+$06]. For physical attacks, Node redeployment detection is proposed based on the change of node neighborhood and the change of measured distances between nodes [SXZC07]. Using mobile agents to detect node compromise in path-based DoS attacks is proposed in [LB07]. Other hardware solutions, including the utilization of powerful high-end sensors in heterogeneous sensor networks to achieve better security and performance, were proposed in [Du08]. In this research, the detection mechanism is required to be in-network operation. In addition, the detection cannot apply the above existing techniques because mobile nodes do not have fixed positions or neighborhoods in the network.

Software solutions eliminate the need for excess hardware and are able to perform within the power constraints of sensor networks. Software-based ATTestation for embedded devices (SWATT) [SPvDK04] is an example of software solutions. SWATT is an external attestation scheme for WSNs using pseudorandom memory traversal. In SWATT, an external verifier challenges other nodes. A checksum was computed from the memory contents of the device being challenged. Deviations from the memory contents referring to an expected value result in the detection of tampering. Every node in the network will have to perform a memory traversal using RC4 as the pseudorandom number generator (PRG). Researchers improved SWATT by decreasing iterations of memory traversal and deployed attestation schemes in distributed WSNs [YWZC07]. However, these schemes are designed for static sensor networks, need a large quantity of message transmissions between sensor nodes, and consume a considerable amount of power.

We propose Unpredictable Software-based Attestation Solution (USAS), providing both an efficient and effective algorithm based on software attestation techniques, in order to improve the node compromise detection with the consideration of the large scale and mobility of mobile wireless sensor networks (mWSNs). USAS eliminates redundancy by reducing the number of nodes performing the RC4 pseudorandom number generation. The selection of the nodes performing the RC4 pseudorandom number generation will be unpredictable. This prevents certain nodes from being more susceptible to be compromised.

The rest of the chapter is organized as follows: we describe the system model and assumptions in Section 4.1. Then we explain the procedures of the proposed attestation algorithm in Section 4.2. The USAS algorithm simulation and analysis is presented in Section 4.3, followed by discussions in Section 4.4.

4.1 System Model and Assumptions

USAS can be applied in both infrastructure networks and ad hoc networks. This work focuses on the applications in infrastructure networks for simplicity. The mWSN considered is of low cost and is battery powered. Base stations are set up in proximity regions and are able to connect with mobile sensor nodes under their own coverage or to communicate with another base station. A base station has the location information of each node that is currently in its coverage region. Base stations also act as network managers, which have access to a centralized database for all the sensor nodes in the network and are protected by tamper-resistant hardware. In other words, full awareness of the capability of each node, including the processing speed and memory capacity is required. This can be achieved by simple registration before node deployment. It is assumed that the base station is a trustworthy entity in the mWSN, leaving the base station as the verifier for other nodes.

The network can deploy a multi-level key management system such as LEAP to prevent basic attacks from outside attackers [ZSJ03]. However, USAS does not rely entirely on encryption for the security of the communication. Normally, the nodes close to the base station are more vulnerable since attackers try to focus on these nodes to compromise more information sent from peripheral nodes. It is very important to detect any compromised nodes among those that are one hop away from the base station. Additionally, with the consideration of preventing malicious data injection, this work considers that the message transmission from an attested node to the base station is one hop in two-generation node chains. The relationship between the I-node and F-node will be discussed in latter section.

It is assumed that the node does not have virtual memory, as is the case with microcontrollers. The MEGA163L was used in the place of current architecture to allow for comparisons to be made with previous studies. The specifications for this microcontroller are as follows: 8 bit processor architecture; 16 KB program memory and 16 bit Addressing; maximum of 8 million instructions per second (MIPS) [Atm]. USAS allows pseudorandom memory traversals to occur in the order of $n \ln n$ in accordance with the Coupon Collector's problem [MU05], where n is the number of memory blocks available. In this case, it is 8K blocks ($16X8K$).

4.2 USAS Procedures

USAS applies a dynamic node chain which is composed of one Initiator node (I-node) and multiple Follower nodes (F-nodes) to conduct the attestation. The base station acts as the external verifier and initializes the attestation. It sends out required random challenge messages to the I-node. Based on the received message, the I-node will run RC4 to generate the random access address for computing memory checksum through random memory traversal. The result will be used as the new random challenge messages for the next generation, F-nodes, to traverse memory and compute checksum. These checksum results will be verified by the base station referring to the checksum of the original registered memory pattern. As long as the memory is rewritten by attackers to compromise the node, the base station will detect the differences, therefore, detecting compromised nodes. The detailed USAS procedures are presented as following subsections in order.

4.2.1 Memory Space Noise Filling

The memory space of sensor nodes can be classified as program memory and data memory. After programming a microcontroller, the remaining program space will be filled with zeros. To prevent compromised nodes from copying the original memory pattern into remaining memory space to deceive the attestation, the scheme proposed in [YWZC07] will be used; the remaining program memory will be filled with pseudorandom number before each node is deployed. Each node has a unique noise pattern which is derived from a noise generation seed called Su. Different from [YWZC07], Su will be deleted before deployment. Each node stores the one way hash function value $H(Su)$ instead of the Su value. A centralized database will store Su that is able to be accessed by base stations when it is necessary. Noise patterns for the future authentication and memory traversal of each node will also be accessible for base stations.

4.2.2 Dynamic Node Attestation Chain

As shown in Figure 4.1, the base station triggers an attestation chain either randomly or as a result of detecting nodes misbehavior in its coverage region. The base station will randomly select a node to be an I-node and send this node a challenge, which includes a random number as a seed for pseudorandom number generation. Along with this seed, authentication messages for the I-node and other F-nodes are included. Challenge messages to each F-node are generated by the I-node, including authentication messages and checksum values computed by the I-node. The design of dynamic node attestation chains offers the randomness for attestation initialization.

To address node compromise issue in mWSNs, the mobility of nodes and dynamic network topology need to be considered. Solutions for static WSNs relying on

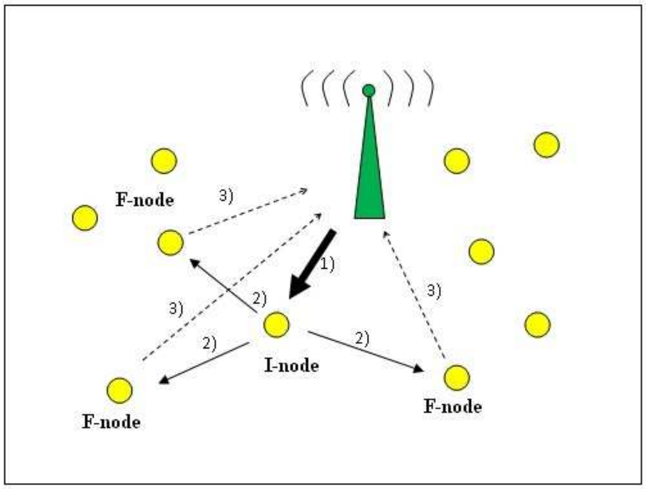


Figure 4.1: Message transmissions during the attestation

fixed neighborhood relationships or clustering are not suitable here. USAS applies attestation solution in mWSNs by temporarily setting up dynamic node attestation chains based on the real time network topology. In our system model, each node in the network is freely moving around without a fixed relative location to other nodes. There is a constraint that USAS only attests nodes one hop from the base station, for the reason we have discussed in Section 4.1. USAS needs at least three message transmissions and double memory checksum computations to attest a two-generation node chain. The attested nodes are required to stay one hop away from the base station to avoid attestation message missing or malicious data injection by intermediate nodes. However, the time consumption for checksum computation is very low. We will present a rough calculation later. Therefore, the impact on node velocity is trivial. There is a tradeoff between the performance of USAS and power consumption of the network. With higher power for message transmission, the node radio range is larger which leads to more attestable nodes in USAS.

4.2.3 Memory Checksum Computing

For all the attested nodes, message authentication through one way hash function computing is requested before running the memory traversal function. That is, each node computes $H(Su)$ upon receiving the challenge message, which includes unique noise generation seed Su and compares the result with the hash value stored locally. If these two values are identical, the memory traversal function is triggered.

USAS applies pseudorandom memory traversal. A function accesses program memory pseudo randomly and loads memory data to compute memory checksum. In order to traverse memory pseudo randomly, the pseudorandom access address is needed. USAS designs different methods to generate this address for I-nodes and F-nodes. Briefly speaking, I-nodes rely on stream cipher computation while F-nodes do not. Instead, F-nodes utilize the challenge messages from I-nodes directly. I-nodes generate memory access address by computing a stream cipher (e.g. RC4 is used as the pseudorandom number generator in SWATT [SPvDK04]) based on the random number seed included in the challenge message given by the base station. The processor accesses this address to load an 8-bit memory data block. Therefore, 8 bits of a current checksum are updated in each round by using the previously computed checksum and performing an XOR operation with the current loaded memory. As a result, a 64-bit checksum is produced.

F-nodes combine a checksum sent from an I-node with the loaded memory and the updated checksum from the previous iteration to generate a 16-bit memory access address. The F-node does not have to perform a stream cipher for each round of memory traversal. The F-node procedure for memory traversal is shown in the form of pseudo code in Table 4.1. After the message challenge from the I-node is authenticated, the F-node uses the checksum value of the I-node as the seed to traverse its program memory and compute the memory checksum. To be specific, the

Table 4.1: Pseudo Code F-node Memory Verification

```
Input:   m: number of iterations of the verification procedure
Output:  Checksum of memory
         Let C be the checksum vector, csfi is the vector of checksum from
         I-node, and j be the current index into the checksum vector
         for i = 1:m
         Build address for memory read
             $A_j \leftarrow ((csfi_j \bigoplus C_{(j+1)mod8}) \ll 8) + C_{(j-1)mod8}$
         Update checksum byte
             $C_j \leftarrow C_j + (Mem[A_j] \bigoplus C_{(j-2)mod8} + (csfi_j \bigoplus C_{(j+1)mod8})_{i-1})$
             $C_j \leftarrow$ rotate left one bit $(C_j)$
             $j \leftarrow (j+1)mod8$ //Update checksum index
         return C
```

64-bit checksum is separated into 8 vectors. In the first memory traversal iteration, each vector is used as the most significant byte of the memory access address. The least significant byte is a predetermined initial vector. The processor accesses this 16-bit address to load memory data and update the 8-bit checksum. After the first iteration, the most significant byte of the memory address is randomized by XORing with the updated 8-bit checksum from another vector computation. The least significant byte is updated by current loaded memory. This procedure produces a new 16-bit memory access address for the next iteration. Then the processor loads the memory data and updates the 8-bit memory checksum. Finally, the F-node produces a 64-bit checksum and sends it back to the base station.

4.2.4 Checksum Result Verification

The base station uses the same memory traversal seed to compute memory checksums based on the original program memory pattern of each node, which is stored in the centralized database before node deployment, in order to verify the content. After comparing checksum values sent from the F-nodes with the checksum computed locally, the base station can detect compromised nodes. Although the I-node does not send its memory checksum back to the base station, the base station can still detect if the I-node is compromised. Since the checksum of the I-node is used to generate pseudorandom memory addresses for the F-nodes, a genuine I-node message will allow F-nodes to compute correct checksums. As long as one F-node sends a correct response, the I-node and this F-node are considered to be trustworthy because the lower bound on the checksum collision probability is only 2^{-64}[SPvDK04].

4.3 Simulation Results and Analysis

In this section, we test the effectiveness of USAS by comparing the checksums computed by a compromised node and by the base station. Then we analyze the efficiency of USAS by calculating computation time consumption and node compromise detection rate.

4.3.1 Memory Traversal Function Testing

In order to test the effectiveness of USAS, simulations were performed for the memory traversal function procedure of each individual F-node. The input seed checksum from the I-node was given as the fixed input to the F-node. A 16 KB file was created as mock program memory of the F-node for the simulation. For simulating the compromised node, the program memory layout of the F-node varied from 1-bit difference to 16 KB difference. The memory traversal function ran 100,000 iterations in each round. For verification, the same checksum computing process ran in the original mock memory. Figure 4.2 shows the number of bit differences from the

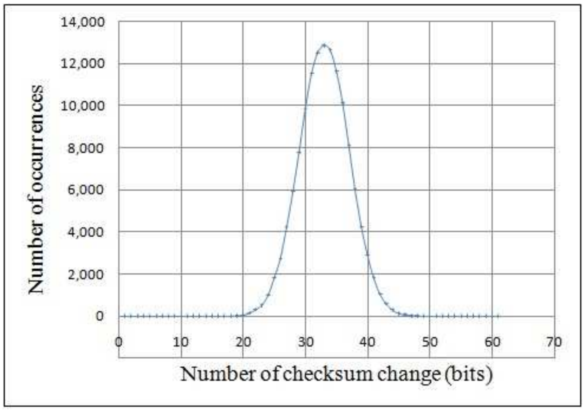


Figure 4.2: Frequency description of the checksum changes

genuine checksum computed by the base station versus occurrences. The difference is around the range of 20 to 40 bits even with a few bits change of the memory file.

4.3.2 Computing Improvement Analysis

With the functions available by the MEGA163L microcontroller, we obtain the number of operations required for proper implementation, as well as the number of clock cycles. In order to determine clock cycles, an analysis of each function used will have to be considered. From the microprocessor's technical document [Atm], F-nodes take 12 clock cycles in each round to update a checksum value for a single memory block in USAS. As a comparison, in SWATT, attesting each node takes 23 clock cycles [SPvDK04] in each round. To put this in perspective, for 100,000 iterations of memory traversal, our implementation would take 0.15s assuming 8MIPS while SWATT would take 0.2875s. USAS decreases memory traversal computation time of selective nodes (F-nodes) by about 48%. Figure 4.3 shows comparison result between SWATT and USAS. When the node density is higher, more F-nodes can be attested in one dynamic node attestation chain; therefore, more computation time can be reduced.

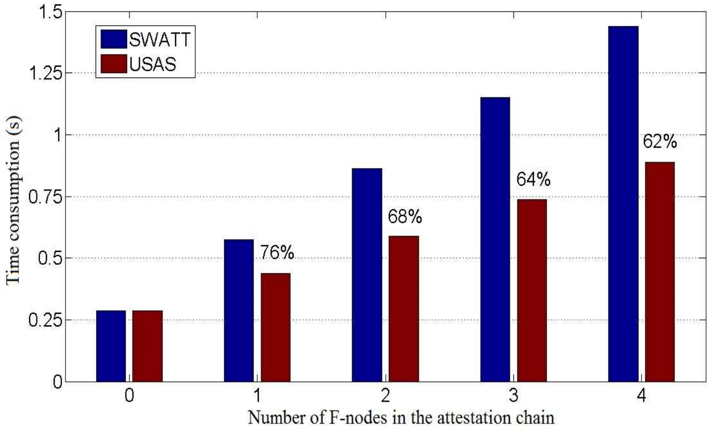


Figure 4.3: Computation time comparison based on 100,000 iterations

4.3.3 Node Compromise Detection Rate Analysis

Based on our system model, we derive the detection rate, which is the probability for base stations successfully verify an F-node. The detection rate for two-generation chain attestation is considered. Assume that there are n nodes that are one hop away from multiple base stations in the entire mWSN. The probability for each node to be compromised is the same.

In compromised nodes, the number of modified memory is m_c in bytes out of the total program memory size m. The probability of base stations detecting program changes from the checksum result P_{ch} is:

$$P_{ch} = (1 - (\frac{m - m_c}{m})^{m \ln m})(1 - 2^{-64}). \tag{4.1}$$

From the view of the network, we need to select a genuine node as the I-node to detect node compromise successfully. The probability that there are i genuine nodes among the attestable n nodes is:

$$P_i = \binom{n}{i}(1 - P_c)^i P_c^{n-i}, \tag{4.2}$$

where P_c is the probability for each node to be compromised. To select one of these genuine nodes as the I-node, the probability P_s is:

$$P_s = \frac{i}{n}\binom{n}{i}(1-P_c)^i P_c^{n-i}. \tag{4.3}$$

We can derive the detection rate P_d from 4.1 to 4.3:

$$\begin{aligned} P_d &= \sum_{i=k}^{n} \frac{i}{n}\binom{n}{i}(1-P_c)^i P_c^{n-i} P_{ch} \\ &= \sum_{i=k}^{n} \frac{i}{n}\binom{n}{i}(1-P_c)^i P_c^{n-i}(1-(\frac{m-m_c}{m})^{m\ln m})(1-2^{-64}), \end{aligned} \tag{4.4}$$

where k is the lower bound of the number of genuine nodes among attested nodes n. In other words, at most $(n-k)$ nodes are compromised. Here we calculate the detection rate when $k = n/2$. Figure 4.4 shows the detection rate when at most half of the attestable nodes are compromised with different number of attestable nodes, $n = 10, 50, 100$, respectively. $m_c = 100$. P_c varies from 0 to 1. When $k = 0$, it indicates the ideal detection rate calculated without considering the current status of the network. $P_d \approx 1 - P_c$. From this figure we observe the following: USAS has good performance in terms of detection rate before $P_c = 1-k/n$ and the performance does not degrade along with the increase of network scale. It is suitable for large scale networks. Although we consider P_c and k as independent elements to each other, in the real network, P_c is the inherent security level of each node, while k depends on both P_c and network maintenance. The lower probability for each node to be compromised (P_c) offers higher security level (k) for the network. Therefore, USAS has high performance in the realistic network status.

4.4 Discussions and Summary

In this chapter, we described the design of USAS to detect compromised nodes in mWSNs. This internal attack detection algorithm is of great importance to elimi-

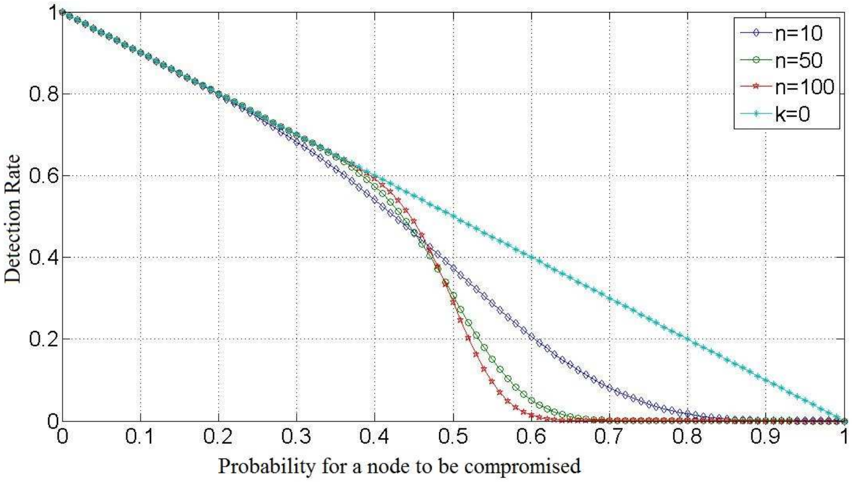


Figure 4.4: Detection rate with $k = n/2$

nate malicious nodes, therefore, to improve trajectory privacy level of the network.

Based on the assembly code analysis, we simplified the checksum computing for the F-node compared with SWATT [SPvDK04]. Therefore, the number of clock cycles for checksum function computation is reduced about 48%. Also, USAS is not constrained by the assumption that additional 'if' statements will detectably slow down the checksum computation. Furthermore, USAS does not rely on majority voting mechanisms which are commonly considered in WSNs. One of the drawbacks of majority voting is the considerable consumption due to the voting message transmissions and message distribution. In USAS, for verifying each node, only a few message transmissions are required.

Additionally, the node chain relationship is dynamic and independent in different attestations. Not all nodes will be I-nodes or F-nodes at once, yet they can both store F-node and I-node algorithms since they are very similar. Also, the role of each node may vary in each attestation. USAS helps to decentralize the network. By having a decentralized model, an attacker cannot predict which node will be the I-node. In this way, focusing on attacking a small number of nodes is not helpful to compromise the overall network. It reduces the probability of the attestation message interception and malicious data injection.

Last but not the least, USAS utilizes the powerful base station and combines with the decentralized network structure to apply node compromise detection in mWSNs. It does not rely on fixed neighborhood relationships. On the contrary, the mobility of the nodes benefits node compromise detection in two aspects: 1) the unpredictable I-node designation and the diffusion of the attestation. 2) Each node has the possibility of being one hop away from base stations.

CHAPTER 5

MODELING COOPERATIVE, MALICIOUS AND SELFISH BEHAVIORS

While some researchers have addressed issues related to TPP in sensor networks, the autonomy of sensor nodes in decentralized/distributed networks has been generally overlooked. More and more sensor network applications consist of temporary, on-the-fly connections among typically autonomous sensing devices where each device decides whether and to what extent it wishes to participate in the network. In pursuit of their own interests, participating devices could therefore misbehave-either by being selfish or by being malicious. These scenarios necessitate decentralized control of the network in which sensor nodes make autonomous decisions regarding their network usage and preserving their trajectory privacy, based on their individual needs.

We have previously proposed a privacy-aware routing protocol to hide source's trajectories in the presence of eavesdropping attacks in chapter 3. It can effectively mislead the adversary by masking the exact location where a particular packet started its journey. This algorithm works relying on the cooperation among sensor nodes. Furthermore, the internal attack is of great importance to be considered into TPP in mWSNs due to the risk of node compromise. We have developed the software-based attestation solution to detect internal attacks in chapter 4. The attestation relies on the formulation of dynamic attestation tree among peer nodes. However, in decentralized mWSNs where nodes are autonomous entities, the cooperation among nodes is not guaranteed. In fact, any protocols or techniques involving cooperation among autonomous nodes need to have the mechanism to consider trust among nodes. In the context of TPP, it is even more challenging to balance between resource sharing for cooperation and privacy protection.

5.1 Related Work Using Game Theory

Game theory is an important tool to model behaviors and strategies of interacting entities. It has been applied to study the TPP related issues recently. Shokri et al. used Stackelberg Bayesian game to formalize the mutual optimization of user-adversary objectives [STT^{+}12]. They aimed to enable system administrators to find the optimal mechanism for TPP. Humbert et al. studied the problem of designing mix zones in the presence of local eavesdroppers [HMFH10]. The work proposed in [CO11, CA12] mainly focused on the incentive designs to provide the required trajectory privacy level for individual users as well as the desired granularity for service providers/mobile commerce companies. The above work attempted to address the optimal strategies for two competing entities, user-adversary or user-service provider. There is some other work that has been proposed to study the interactions between peer nodes. Researchers used game theory to model the cooperation behaviors while taking the selfishness of autonomous nodes into consideration in the non-cooperative network environment [FMHP09, DMNRDSHSH11, KPM11]. In such networks, each node aims to maximize its own payoff while determining whether it cooperates TPP activities to gain (or assist other nodes to gain) trajectory privacy. To study the malice of compromised nodes, researchers have presented comprehensive analysis in intrusion detection games in mobile networks [WCK09, YCM06]. The previous work is of great inspiration in discussing the degree of trust in cooperation among peer nodes. However, solely considering selfishness or malice is insufficient for vulnerable networks composed of autonomous nodes.

In this chapter, we formulate in-network TPP activities by applying the Bayesian game, named the *TPP* game, to model the cooperation and trust behaviors of autonomous nodes with taking account both selfishness and malice of sensor nodes in decentralized/distributed mWSNs. The selfishness is modeled by deploying trajectory privacy sensitivity customization. We analyze the TPP game from static and dynamic point of views and provide the suggested equilibrium strategy for nodes to trust neighbors to preserve their trajectory privacy.

The rest of this chapter is organized as follows. An introduction to our TPP game model is provided in section 5.2, followed by the analysis details in section 5.3, 5.5 and 5.6. Finally, we summarize this work in section 5.7.

5.2 Trajectory Privacy Preservation Game Preliminaries

We aim to investigate the game theoretic approach to model and analyze the cooperative, selfish and malicious behaviors of autonomous nodes in TPP activities. Therefore, nodes seeking TPP cooperations can evaluate the degree of trust and tolerate node compromise attacks in mWSNs.

5.2.1 System and Attack Models

The system we consider is composed of mobile sensor nodes and one or more base stations/sinks. Nodes are aware of their locations and transmit sensing data (and spatiotemporal data by needs) through one-hop or multi-hop communications with the sink and other peer nodes. Access points/gateways, where data are collected and primarily processed (e. g. basic data cleaning and reduction) before they are forwarded to the sink, are not restricted in the system. We assume access points (if they exist) and sinks have adequate transmission and computing capabilities or necessary hardware units to counter security and privacy attacks. Additionally, cryptographic techniques are deployed to secure data transmissions.

The adversary is capable of launching an eavesdropping attack. Although proper cryptographic techniques can prevent such attacks from breaching nodes' data privacy, they are vulnerable when the adversary has certain background knowledge of the target. Additionally, the adversary can compromise nodes in the network and launch internal attacks. For instance, compromised nodes are used to track the target or reveal its trajectories to eavesdroppers or other compromised nodes. This attack model is set up on top of the attack model in [JPC$^+$12]. We have solely addressed eavesdropping attacks in the previous work. However, when node compromise attack is considered among autonomous sensor nodes, the noncooperative and malicious nodes behaviors must be studied to cope with the noncooperative network environment, which is our focus in this work.

5.2.2 Sensitivity Customization

The trajectory privacy sensitivity, noted as sensitivity through this paper, represents the privacy requirement level of a node in the specific area at particular time. Sensitivity is determined by both spatial and temporal information.

The network area is divided into small subareas. We categorize subareas into two types: open areas (OAs) and sensitive areas (SAs) according to the sensitivity required by a node in different subareas. When the node is traveling in its OAs, trajectories are relatively open to other network entities and there is no additional protections besides security techniques. On the other hand, when the node is traveling in its SAs, trajectories need to be highly protected from untrustworthy or compromised nodes. For example, OAs can be users' working places since such information is easy to obtain from public resources in most cases. SAs can be the restaurants or hospitals that users occasionally visit. The above customization only involves spatial information which is insufficient in practice. Given an example that if a user goes to the office after working hours, he/she may want to keep such information as privacy which makes that office a SA. Therefore, the sensitivity of an area needs to be assigned along with the specific time span.

The sensitivity is a customized TPP parameter regarding individual nodes. In an autonomous network, before nodes join/rejoin the network or while on the move, users can determine the sensitivity and make the selection on OAs and SAs that comply with a certain trajectory privacy level θ. θ is defined as the ratio of the traveling time in SAs over the overall traveling time of each node. The overall traveling time is the life time or a refreshing period of a node. Individual nodes can have different θ values. However, there is a maximum requirement defined by the network administrator, denoted as Θ. For example, node i's trajectory privacy level $\theta_i \leq \Theta$. From the individual user's point of view, greater θ value indicates higher trajectory privacy. From the network's point of view, the user's trajectory privacy level needs to be limited to Θ since nodes in SAs prioritize their trajectory privacy and may prefer hiding from any other node rather than cooperating in TPP activities. It is worth noting that the trajectory privacy level θ of a node does not involve any private trajectory information. Therefore, neighboring nodes can easily retrieve this information through queries to peers.

5.3 Trajectory Privacy Preservation Game

The TPP game is conducted between two neighboring nodes, node i and node j, in the network. Node i is in SAs and needs cooperation from node j in preserving i's trajectory privacy during data transmissions. However, node i does not know if node j will cooperate, defect, or even attack (node j may be an compromised node) due to the unknown type of node j. We use a Bayesian game formulation to model the interactions between two nodes in TPP activities. The set of players is $\mathcal{N} = \{i, j\}$.

Table 5.1: Payoff matrix of the TPP game

	i	
j	Trust	Not Trust
Cooperate	$G_r - C_p, G_p - C_p$	$G_r, 0$
Defect	$0, -C_p$	$0, 0$

(a) node j is open, $t_j = O$

	i	
j	Trust	Not Trust
Cooperate	$G_r - C_p - G_\theta, G_p - C_p$	$G_r - G_\theta, 0$
Defect	$0, -C_p$	$0, 0$

(b) node j is selfish, $t_j = S$

	i	
j	Trust	Not Trust
Cooperate	$G_r - C_p, G_p - C_p$	$G_r, 0$
Attack	$G_p - C_A, -G_p - C_p$	$-C_A, 0$

(c) node j is malicious, $t_j = M$

i has one type which is a regular node in SAs. That means $t_i \in \mathcal{T}_i = \{S\}$. i can choose whether to trust j in cooperating in TPP activities. Actions available to i are $a_i \in \mathcal{A}_i$ =\{*Trust, Not Trust*\}. j has three types which are regular nodes in OAs, denoted as Open nodes; regular nodes in SAs, denoted as Selfish nodes, and Malicious nodes. That means $t_j \in \mathcal{T}_j = \{O, S, M\}$. We differentiate selfish nodes from open nodes to remark that nodes in SAs areas prioritize their trajectory privacy. Actions available to j are $a_j \in \mathcal{A}_j$ =\{*Cooperate, Defect, Attack*\}. Open nodes and selfish nodes can play either *Cooperate* or *Defect*, while malicious can choose to either *Cooperate* or *Attack*. The payoff matrices of the game is presented in Table 5.1. G_r denotes the cooperation payoff of j (e.g. cooperative credit gain) when it cooperates in TPP. C_p is the participation cost. G_p is the trajectory privacy gain of i when j cooperates.G_θ is the trajectory privacy leakage of j when $t_j = S$ and j cooperates. C_A is the cost of a malicious node to *Attack*. We also denote β as the attack success rate. Except that the type of j is uncertain to i, other information is known to each player and both players know this fact. Due to the high priority of trajectory privacy of selfish nodes, we restrict that $G_\theta > G_r$. Additionally, we focus on discussing TPP issues in this project. Therefore, we assume that G_p is far greater than other payoff elements. C_p and C_A are comparable and less than other payoff elements. Next, we present the equilibrium analysis of the TPP game.

5.4 Equilibrium Analysis in the Strategic Form

We begin the equilibrium analysis with considering this TPP game as a static Bayesian game. We assumed that both players are rational. Their objectives are to maximize their own expected payoffs. Bayesian Nash Equilibria (BNE) specify actions or randomized strategies of each type of player, which would be maximizing the expected payoffs for each player in the strategic form [Mye97]. Therefore, our equilibrium analysis here is to find existing BNE. In the TPP game, the type of i is certain to both players and i is uncertain about j's type. i has the initial belief about j's type, which is the probability distribution over all possible types of j. j is malicious with the probability of $(1 - P)$, selfish with the probability of $P\theta_j$ and open with the probability of $P(1 - \theta_j)$.

From the payoff matrices of the game, it is not difficult to eliminate some strategies from possible BNE by dominance. *Defect* is dominated for $t_j = O$ and *Cooperate* is dominated for $t_j = S$. j in turn has two possible pure-strategy BNE: $\sigma_j \in$ $\{$(*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Cooperate* if $t_j = M$), (*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Cooperate* if $t_j = M$)$\}$.

- Case 1: $\sigma_j = (\textit{Cooperate} \text{ if } t_j = O,\ \textit{Defect} \text{ if } t_j = S,\ \textit{Cooperate} \text{ if } t_j = M)$.

  In this case, the expected payoff of i if $\sigma_i = \textit{Trust}$ is

$$\mathscr{U}_i(T) \quad = \quad (G_p - C_p)P(1-\theta_j) + (G_p - C_p)(1-P) + (-C_p)P\theta_j. \quad (5.1)$$

  $(G_p - C_p)P(1-\theta_j) + (G_p - C_p)(1-P)$ is the payoff of obtaining cooperation from j to gain trajectory privacy. $(-C_p)P\theta_j$ is the payoff when j is selfish and defects. The expected payoff of i if $\sigma_i = \textit{Not Trust}$ is

$$\mathscr{U}_i(NT) = 0. \quad (5.2)$$

  Therefore, if 5.1 >5.2, i.e. $P < \frac{G_p - C_p}{G_p\theta_j}$, the best response for i is always *Trust*. In this case, if $t_j = M$, j will deviate from *Cooperate* to *Attack* since j will breach i's privacy and get higher expected payoffs. Thus, there is no pure-strategy BNE when $P < \frac{G_p - C_p}{G_p\theta_j}$. If 5.1 <5.2, i.e. $P > \frac{G_p - C_p}{G_p\theta_j}$, the best response for i is always *Not Trust*. Correspondingly, j's best response is *Cooperate* if $t_j = M$. Hence, $(\sigma_j, \sigma_i) = \{(\textit{Cooperate} \text{ if } t_j = O,\ \textit{Defect} \text{ if } t_j = S,\ \textit{Cooperate} \text{ if } t_j = M),\ \textit{Not trust}\}$ is a possible pure-strategy BNE if $P > \frac{G_p - C_p}{G_p\theta_j}$.

- Case 2: $\sigma_j = (\textit{Cooperate} \text{ if } t_j = O,\ \textit{Defect} \text{ if } t_j = S,\ \textit{Attack} \text{ if } t_j = M)$.

  In this case, the expected payoff of i if $\sigma_i = \textit{Trust}$ is

$$\begin{aligned}\mathscr{U}_i(T) \quad = \quad & (G_p - C_p)P(1-\theta_j) + (-C_p)P\theta_j \\ & +(-G_p - C_p)(1-P)\beta + (-C_p)(1-P)(1-\beta). \quad (5.3)\end{aligned}$$

  $(-G_p - C_p)(1-P)\beta$ is the payoff of being successfully attacked by j. $(-C_p)(1-P)(1-\beta)$ is the payoff when j fails in attacking. The expected payoff of i if σ_i =*Not Trust* is

$$\mathscr{U}_i(NT) = 0. \quad (5.4)$$

  So if 5.3 <5.4, i.e. $P < \frac{G_p\beta + C_p}{G_p(1-\theta_j) + G_p\beta}$, the best response for i is always *Not Trust*. In this case, if $t_j = M$, j will deviate from *Attack* to *Cooperate* to get higher expected payoffs. Thus, there is no pure-strategy BNE when $P < \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}$. If 5.3 >5.4, i.e. $P > \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}$, the best response for i is always *Trust*. Correspondingly, j's best response is *Attack* if $t_j = M$. Hence (σ_j, σ_i) ={(*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Attack* if $t_j = M$), *Trust*} is a possible pure-strategy BNE if $P > \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}$.

Now we further analyze the conditions under which the pure-strategy BNE exists. Given $0 < P < 1$ and $0 < \theta_j < 1$, the condition that $P > \frac{G_p-C_p}{G_p\theta_j}$ can establish is $1 - \frac{C_p}{G_p} < \theta_j < 1$. On the other hand, the condition that $P > \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}$ can establish is $0 < \theta_j < 1-\frac{C_p}{G_p}$. θ_j is retrieved by j through sending queries to i if there is no previous local records.

From the above analysis, we can summarize the existence of pure-strategy BNE: the TPP game has one pure-strategy BNE when $P > P_0 \in \{\frac{G_p-C_p}{G_p\theta_j}, \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}\}$. If $1 - \frac{C_p}{G_p} < \theta_j < 1$, there exists a pure-strategy BNE (σ_j, σ_i) = {(*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Cooperate* if $t_j = M$), *Not trust*} when $P > \frac{G_p-C_p}{G_p\theta_j}$. If $0 < \theta_j < 1 - \frac{C_p}{G_p}$, there exists a pure-strategy BNE (σ_j, σ_i) = {(*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Attack* if $t_j = M$), *Trust*} when $P > \frac{G_p\beta+C_p}{G_p(1-\theta_j)+G_p\beta}$. This conclusion verifies that in order to have stable status that encourages cooperation among nodes in TPP activities, θ_j needs to be restricted to an upper bound of Θ.

We have analyzed the pure-strategy BNE of the TPP game. However, under pure strategies, i either cannot gain trajectory privacy by trusting other nodes or can frequently be attacked by malicious nodes where high expected payoffs encourage malicious nodes to attack. Therefore, we need to find the mixed-strategy BNE of the TPP game. Such BNE exists when $P < P_0$. Let ϕ be the probability of σ_i = *Trust*. Let ψ be the probability of σ_j = *Attack* when $t_j = M$. The mixed-strategy BNE is derived as follows. The expected payoffs of i when σ_i = *Trust* and when

$\sigma_i = \mathit{Not\ Trust}$ are respectively:

$$\begin{aligned}\mathscr{U}_i(T) &= (G_p - C_p)P(1-\theta_j) + (-C_p)P\theta_j + (1-\psi)(G_p - C_p)(1-P) \\ &\quad +\psi(1-P)((-G_p - C_p)\beta + (-C_p)(1-\beta)). && (5.5) \\ \mathscr{U}_i(NT) &= 0. && (5.6)\end{aligned}$$

The expected payoffs of j when $\sigma_j = \mathit{Cooperate}$ and when $\sigma_j = \mathit{Attack}$ are respectively:

$$\begin{aligned}\mathscr{U}_j(C) &= (G_r - C_p)\phi + G_r(1-\phi). && (5.7) \\ \mathscr{U}_j(A) &= (G_p - C_A)\phi\beta + (-C_A)\phi(1-\beta) + (-C_A)(1-\phi). && (5.8)\end{aligned}$$

To derive a mixed-strategy BNE, j's attacking rate needs to satisfy $\mathscr{U}_i(T) = \mathscr{U}_i(NT)$ if $t_j = M$. Thus, j's equilibrium strategy is to attack with probability $\psi^* = \frac{G_p(1-P\theta_j)-C_p}{G_p(1-P)(1+\beta)}$. Similarly, i's equilibrium strategy is to trust with probability $\phi^* = \frac{G_r+C_A}{G_p\beta+C_p}$. The static TPP game has a mixed-strategy BNE when $(\sigma_j, \sigma_i) =($*Cooperate* if $t_j = O$, *Defect* if $t_j = S$, *Attack* with probability ψ^* if $t_j = M$, *Trust* with probability ϕ^*).

5.5 Dynamic TPP Game and Perfect Bayesian Equilibrium

Thus far, we have analyzed the one-stage TPP game. The challenge of applying such a game model is assigning i a proper initial belief of j's type. In mWSNs, nodes are highly distributed. Relying on the centralized administrator to provide the regular nodes' rate P is costly. Therefore, i needs to dynamically update its belief of j's type in a distributed manner in the multi-stage TPP game.

5.5.1 Belief System

We consider a dynamic Bayesian game which is a repeated one-stage TPP game with no discount factor to model the multi-stage TPP game. The game is infinite since players cannot predict when neighboring nodes leave the network. The static TPP game is repeated in each time slot. We continue to use notations in the static TPP game with minor revisions. $a_j(t)$ denotes j's action at stage t. $\tilde{a}_j(t)$ denotes i's observation of j's action. i observes j's actions with the observation rate α_t, the action false active rate γ_t and the attack false alarm rate β_t. Additionally, let $\mu_i(t_j|h_j^t)$ be i's belief of j, where h_j^t is the action history profile j at the beginning of stage t, i.e. $h_j^t = (a_j(0), a_j(1), ..., a_j(t-1))$. Given j's action history profile h_j^t and type t_j, $P(a_j(t)|t_i, h_j^t)$ is the probability that $a_j(t)$ is observed at stage t. Based on Bayes' rule, i's posterior belief of j is calculated by:

$$\mu_i(t_j|(h_j^t, \tilde{a}_j(t))) = \frac{\mu(t_j|(h_j^t))P(\tilde{a}_j(t)|h_j^t, t_j)}{\sum_{\tilde{t}_j \in \mathcal{T}_j} \mu_i(\tilde{t}_j|h_j^t)P(\tilde{a}_j(t)|h_j^t, \tilde{t}_j)}. \tag{5.9}$$

where $\mu_i(\tilde{t}_j|h_j^t) > 0$. In our game model, with an altering of notation, we have the following:

$$\begin{aligned}
P(\tilde{a}_j(t) = Cooperate|t_j = O) &= \alpha_t(1-\beta_t) \\
P(\tilde{a}_j(t) = Defect|t_j = O) &= 1-\alpha_t \\
P(\tilde{a}_j(t) = Attack|t_j = O) &= \alpha_t\beta_t \\
P(\tilde{a}_j(t) = Cooperate|t_j = S) &= \gamma_t(1-\beta_t) \\
P(\tilde{a}_j(t) = Defect|t_j = S) &= 1-\gamma_t \\
P(\tilde{a}_j(t) = Attack|t_j = S) &= \gamma_t\beta_t \\
P(\tilde{a}_j(t) = Cooperate|t_j = M) &= \alpha_t(1-\beta_t)(1-\psi) \\
P(\tilde{a}_j(t) = Defect|t_j = M) &= 1-\alpha_t \\
P(\tilde{a}_j(t) = Attack|t_j = M) &= \alpha_t(\psi+(1-\psi)\beta_t)
\end{aligned} \tag{5.10}$$

Formula 5.9 and 5.10 form the belief system [Osb04] for i to update its belief of j as the game is played sequentially. With both this belief system and the initial belief that i holds, i is able to compute updated belief. It is worth noting that in applications requiring high trajectory privacy levels, i can apply the Grim Trigger strategy once an attack is observed. However, keeping a dynamic updated belief on all types of nodes for further possible cooperation is plausible in sparse networks.

5.5.2 Perfect Bayesian Equilibrium (PBE)

We have found BNE of the static TPP game in previous sections. However, when the game involves sequential multiple stages, Nash Equilibrium needs to be strengthened with the notion of subgame perfection. The relevant notion of equilibrium will be PBE. PBE requires each player's strategy to specify optimal actions, given the player's beliefs and the strategies of all other players, and the beliefs are consistent with Bayes' Rule whenever it is applicable. It specifies a feasible strategy profile for players to optimize the expected payoffs in the multi-stage game. We now show that there exist PBE in the dynamic TPP game.

We first prove that the proposed dynamic TPP game satisfies the Bayesian condition B(i)-B(iv) and P [FT91]. Then we determine the PBE in such games.

Lemma 5.5.1 *The proposed dynamic TPP game satisfies Bayesian conditions B(i)-B(iv) and P:*

B(i): Posterior beliefs are independent, and all types of player i have the same beliefs.

B(ii): Bayes' rule is used to update beliefs whenever possible.

B(iii): Players do not signal what they do not know.

B(iv): Posterior beliefs are consistent for all nodes with a common joint distribution on the type of another player given h^t.

P: For each player i, type t_i, player i's alternative strategy σ_i' and history h^t,

$$\mathscr{U}_i(\sigma|h^t, t_i, \mu(\cdot\,|h^t)) \geq \mathscr{U}_i((\sigma_i', \sigma_{-i})|h^t, t_i, \mu(\cdot\,|h^t)). \tag{5.11}$$

Proof. B(i) is satisfied because i only has one type. The proposed belief update system was derived according to Bayes' rule. Thus, B(ii) is satisfied. B(iii)is satisfied because j's signal is j's action which is observed by i, and B(iv) is satisfied since this is a two-player game.

According to the rationality of the players, given i's updated belief of j, $\mu_i(t_j|h_j^t)$, and h_j^t, i's optimal behavior strategy σ_i^* is to maximize his expected payoff based on i's belief. Therefore, σ_i^* satisfies:

$$\mathscr{U}_i(\sigma_j, \sigma_i^*)|h_j^t, t_i, \mu_i(t_j|(h_j^t, a_j(t))) \geq \mathscr{U}_i((\sigma_j, \sigma_i'|h_j^t, t_i, \mu_i(t_j|(h_j^t, a_j(t))). \tag{5.12}$$

Similarly, j's optimal behavior strategy σ_j^* satisfies:

$$\mathscr{U}_j(\sigma_j^*, \sigma_i)|h_i^t, t_j, \mu_j(t_i|(h_i^t, a_i(t))) \geq \mathscr{U}_j((\sigma_j', \sigma_i|h_i^t, t_j, \mu_j(t_i|(h_i^t, a_i(t))). \tag{5.13}$$

σ' denotes the alternative strategy of the player. In this two-player game, formula 5.12 and 5.13 show the sequential rationality of each player, which satisfies P. □

In this paragraph, we derive the PBE of the dynamic TPP game. At stage t, recall ϕ denotes the probability of $\sigma_i = \mathit{Trust}$ and ψ denotes the probability of $\sigma_j = \mathit{Attack}$ when $t_j = M$.

$$\begin{aligned} \mathscr{U}_i(a_i(t) = T) &= (G_p - C_p)\mu_i(t_j = O|h_j^t) + (-C_p)\mu_i(t_j = S|h_j^t) \\ &\quad +(1-\psi)(G_p - C_p)\mu_i(t_j = M|h_j^t) \\ &\quad +\psi(\mu_i(t_j = M|h_j^t)((-G_p - C_p)\beta + (-C_p)(1-\beta)). \end{aligned} \tag{5.14}$$

$$\mathscr{U}_i(a_i(t) = NT) = 0. \tag{5.15}$$

$$\mathscr{U}_j(a_j(t) = C) \;=\; (G_r - C_p)\phi + G_r(1-\phi). \tag{5.16}$$

$$\mathscr{U}_j(a_j(t) = A) \;=\; (G_p - C_A)\phi\beta + (-C_A)\phi(1-\beta) + (-C_A)(1-\phi). \tag{5.17}$$

This mixed-strategy equilibrium needs to satisfy the condition that different strategies cannot be differentiated by each player from the expected payoffs. Therefore, we derive the PBE pair based on the equivalence of equations 5.14 and 5.15, and the equivalence of 5.16 and 5.17. Thus, we have the following:

$$\begin{aligned} \psi_t^* &= \frac{G_p(1-(1-\mu_i(t_j = M|h_j^t))\theta_j) - C_p}{G_p\mu_i(t_j = M|h_j^t)(1+\beta)} \\ \phi_t^* &= \frac{G_r + C_A}{G_p\beta + C_p}. \end{aligned} \tag{5.18}$$

We now discuss the existence of pure-strategy PBE. In the case that 5.14 >5.15, i always plays *Trust* and j always plays *Attack*. In this case ψ_t^* satisfies $\psi_t^* < \frac{G_p(1-(1-\mu_i(t_j=M|h_j^t))\theta_j)-C_p}{G_p\mu_i(t_j=M|h_j^t)(1+\beta)}$ and $\psi_t^* = 1$. The condition for such a case to exist is that $\mu_i(t_j = M|h_j^t) < \frac{G_p(1-\theta_j)-C_p}{G_p(1+\beta-\theta_j)}$ and $\theta_j < 1 - \frac{C_p}{G_p}$. Similarly, the pure-strategy pair of i always plays *Not Trust* and malicious j always plays *Cooperate* exists under the condition that $\mu_i(t_j = M|h_j^t) < \frac{G_p(\theta_j-1)+C_p}{G_p\theta_j}$ and $\theta_j > 1 - \frac{C_p}{G_p}$, which does not hold in this TPP game model. Therefore, there exists one pure-strategy PBE pair $(\psi_t^* = 1, \phi_t^* = 1)$ when $\mu_i(t_j = M|h_j^t) < \frac{G_p(1-\theta_j)-C_p}{G_p(1+\beta-\theta_j)}$ and $\theta_j < 1 - \frac{C_p}{G_p}$. In sum, given the belief $\mu_i(t_j|(h_j^t, a_j(t)))$ which can be derived by equation 5.9, the PBE pair for the dynamic TPP game is (ψ_t^*, ϕ_t^*).

5.6 Simulation Results and Analysis

In this section, we provide simulation results to illustrate the properties of equilibrium strategies in the dynamic TPP game. We implement PBE strategies in mobile nodes. Nodes are moving in the 200X200 m^2 area within DHDN/3-degree

Gauss-Kruger zone 2 (EPSG code: 31466). Nodes' trajectories are generated by using the Random Street model of BonnMotion [Bon]. The maximum transmission range of each node is set to be 40 m, referenced to the average maximum transmission range in our real experiments on MEMSIC sensors equipped with MTS420 boards. The default values of the payoff matrix and system parameters are $G_p = 50, G_r = 5, C_p = 1, C_A = 2, P = 0.2, \alpha_t = 0.8, \beta = 0.9, \beta_t = 0.2, \gamma_t = 0.2$.

We first analyze the probability for a node to play *Trust* in the two-node dynamic TPP game. Figure 5.1 shows that when attackers have a higher successful attacking rate, users need to decrease trust. The trajectory privacy gain G_p has more obvious impact on trust. When G_p has greater values, it indicates that the user weighs trajectory privacy more critically and also encourages the attacker to attack more frequently. Therefore, users' trust of others dramatically decreases.

Figure 5.2 and Figure 5.3 show the node i's belief update and node j's corresponding attacking rate when consecutive *Attacks* are observed by i in each stage t. $\theta_j = 0.5$. The figures suggest that regardless the initial belief that i holds, observing consecutive *Attacks* gives a very fast convergence of i's belief that j is a malicious node. As a result, the attacker has to reduce the attacking rate fast. These results are obtained when we assume that two nodes are always one-hop neighboring nodes. As a comparison, Figure 5.5 and Figure 5.6 show the belief update and attacking rate when the malicious node plays *Attack* rationally according to its PBE strategy. The data are collected from different scenarios where the malicious nodes A and B have different trajectory similarities with i. Trajectories of node i, A and B are illustrated in Figure 5.4. Node A is always closely following i and node B is i's one-hop neighbor in half of the time. The result is based on the average value from 1,000 iterations. i's belief of node A converges slightly slower compared with the results in Figure 5.2 because node A also plays *Cooperate* and *Attack* is observed less frequently. Node A's attacking rate also reduces slower in order to gain more payoffs by attacking in longer time. i's belief of node B converges much slower because at some stages the TPP game cannot be conducted. In this case, the belief remains the same as the previous stage.

Finally, we simulate a mWSN composed of 200 mobile nodes and analyze the network trajectory privacy gain under different strategies of the nodes seeking TPP cooperation. There are 40 malicious nodes and 160 regular nodes, including selfish nodes and open nodes. Each regular node randomizes its trajectory privacy level with the upper boundary at $\Theta = 0.7$. Malicious nodes are programmed to play the PBE strategy. Regular nodes are programmed to play one of the following three strategies: the pure strategy, always *Trust*; the mixed BNE strategy; and the PBE strategy respectively. The results are presented in Figure 5.7 and Figure 5.8. We show the average data based on 10 groups of trajectory data and 1,000 iterations for each group. When regular nodes playing mixed BNE strategy, both regular nodes and attackers get low average payoffs in the actual game within 20 stages. This is because the mixed strategy suggests regular nodes a fair probability to play *Trust* no matter how malicious nodes act. This strategy neither encourages nor discourages malicious nodes to *Attack*. On the other hand, if regular nodes always play *Trust*, it greatly encourages malicious nodes to *Attack*. Therefore, this pure strategy gives very high average payoffs to the attacker. Although malicious nodes follow the PBE strategy and reduce the attacking rate gradually, regular nodes get high average payoffs by always getting cooperation from open nodes. Finally, when both players follow the PBE strategy pair, regular nodes get even higher average payoffs but malicious nodes' payoffs dramatically reduce along with the belief convergence. This is because regular nodes take actions according to the dynamically updated beliefs of other peer nodes. The PBE strategy allows regular

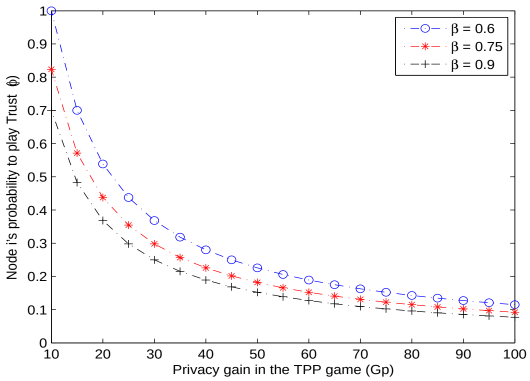


Figure 5.1: Users' trust in the TPP game

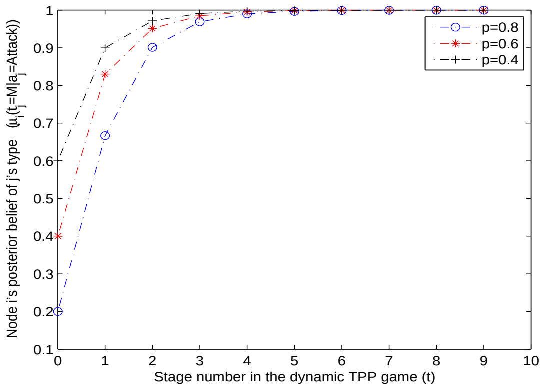


Figure 5.2: Node i's posterior belief given the observation of consecutive *Attacks*

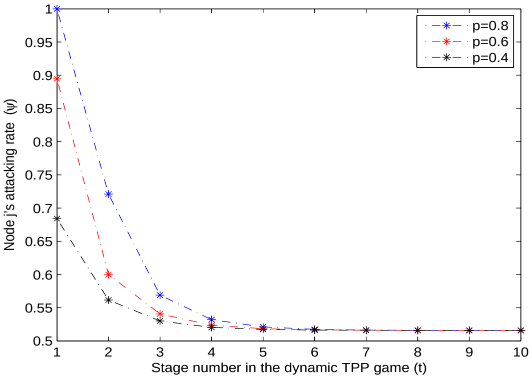


Figure 5.3: Node j's attacking rate given the observation of consecutive *Attacks*

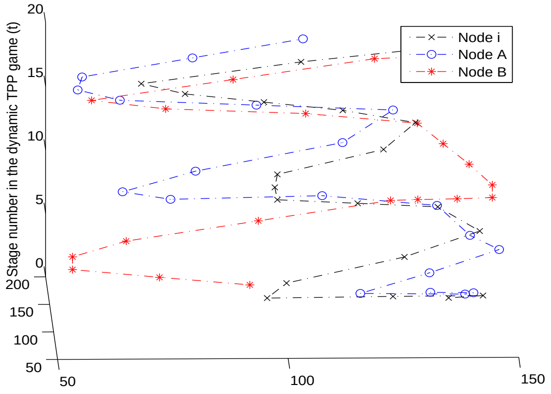


Figure 5.4: The illustration of selected nodes' trajectories in the network

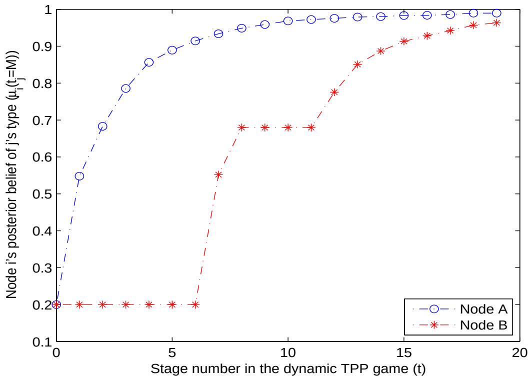


Figure 5.5: Node i's belief update of the selected malicious node

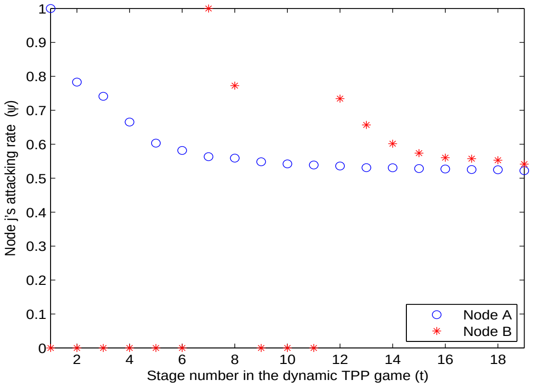


Figure 5.6: The attacking rate of the selected malicious node

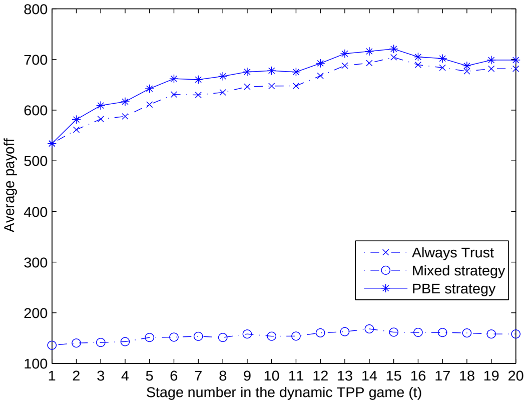


Figure 5.7: The actual average payoff of regular nodes in the network

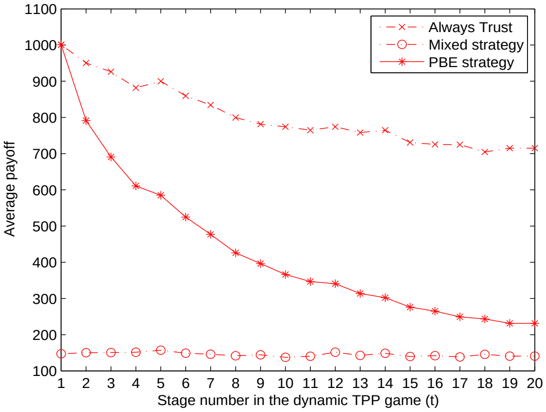


Figure 5.8: The actual average payoff of malicious nodes in the network

nodes to catch more opportunities to *Trust* open nodes to gain trajectory privacy while often playing *Not Trust* with malicious nodes.

5.7 Summary

In this chapter, we revisited the trust issue in TPP cooperation. We formulated TPP activities as a two-player Bayesian game and modeled cooperative, selfish and malicious nodes' behaviors in the TPP game. The game was analyzed in both strategic and dynamic forms. We derived the pure and mixed strategies in the BNE and PBE, respectively. We suggested the trust degree for mobile nodes seeking TPP cooperation. The limitations and future directions will be discussed in the next Chapter.

CHAPTER 6

DISCUSSIONS, FUTURE WORK AND CONCLUSION

This dissertation described an in-network TPP framework that incorporates the trajectory privacy-aware routing protocol design, the internal attack detection algorithm development, nodes' behavior mathematical modelling and performance evaluations. This chapter reflects the contributions, discusses the limitations and future work, and concludes.

6.1 TPriv: Privacy-aware Routing Protocol to Prevent External Attacks

We designed the TPriv to prevent the eavesdropping attack in mWSNs, where attackers may deploy static eavesdroppers near APs to estimate mobile nodes' trajectories. We defined the passive trajectory privacy attack model in mWSNs. Based on our network model, we presented TPriv, including BTPriv and M-hop STPriv, to hide the trajectory of data source nodes from privacy adversaries. We defined the proper evaluation metrics to evaluate the trajectory privacy performance of the above methods. The effectiveness was presented by simulation results.

One limitation of TPriv is the extra power consumption for query-and-reply during the privacy-aware routing phase. Although the development of both micro-computing and nanotechnology is resolving the power issue to a large extent, real applications are still lagging. One possible improvement is to embed this query-and-reply operation into route discovery. In this case, one-time pad VID can be implemented during route discovery in mWSNs where the network topology is highly dynamic. However, the lifetime of VID needs to be redesigned to comply with the lifetime of routes.

6.2 USAS: Software-based Attestation Solution to Detect the Internal Attack

We designed the USAS to protect trajectory privacy in the presence of the internal attack. We applied software-based attestation to detect compromised nodes in mWSNs. We have simulated the promising USAS and shown the effectiveness for detecting any falsification of program memory of sensor nodes. By deploying the dynamic node attestation chains, the attestation computation time and power consumption are improved from the previous work. With the unpredictable I-node designation, the security level of mWSNs is increased.

We have thus far only considered nodes one hop away from base stations here. However, with the proper design of the dynamic node chain and the number of generations, USAS could be applied in multi-hop communications in the entire mWSN. Another improvement is to relax the responsibility of the base station and adapt USAS into more distributed network architectures.

6.3 TPP Game: Modeling Cooperative, Malicious and Selfish Nodes' Behaviors

We applied Bayesian game theory to model nodes' behaviors in TPP activities in mWSNs. We formulated the characteristics of autonomous nodes, including selfish, malicious and cooperative, in the TPP game, and evaluated the trustworthiness of the unknown type node. The equilibrium strategies of the game have been derived and analyzed in theory and simulation results.

In the current TPP game model, the post-detection strategy has not been discussed here. For example, the regular node can take the Grim Trigger strategy to cut off any cooperation once it observes an *Attack* from a node or has 100 percent belief in a node's malice. Another direction to improve this work is to specify the landscape and combine it with the sensitivity customization. If done, the generality of this framework will be degraded with the tradeoff to gain higher performance payoff by avoiding seeking cooperation from selfish nodes. Additionally, the current framework is restricted by the two-node game, which cannot be applied to TPP techniques that require multiple entities to cooperate in TPP activities.

6.4 Future Directions

Regardless of the above improvements, there are other potential directions to enhance the current work and develop a more comprehensive TPP framework.

6.4.1 A Context-based Trajectory Data Model

The context-based trajectory data model is aimed to illustrate the physical, spatiotemporal, symbolic, absolute, relative or uncertain context of trajectories in mWSNs. To generate this model, there are two tasks that need to be done: transform spatiotemporal instances into the basic context-based trajectories; and balance the tradeoffs between trajectory privacy, cost and data accuracy that result from the degree of uncertainty and granularity of trajectory data.

As we noted earlier, the current trajectory data model is defined as an ordered list of location samples at specific instances in time [GS05]. From this list, the movement of the object is inferred and information is provided for LBS. However, a more thorough description, representation and manipulation of trajectory modeling could provide a more physical, symbolic, absolute or relative context, which in turn could aid in privacy protection. For example, when analyzing a single trajectory, a low-active pattern of movement within a certain area, the sudden appearance of an outlier, or high frequency of querying for the same online service, may imply that an adversary is trying to survey other users. Besides examining a single trajectory, this model need to consider trajectories with certain spatial and temporal relations (such as co-location in space, co-existence in time, lagged co-incidence, etc.). For example, a high correlation between the trajectories of neighboring nodes could imply that one or multiple nodes are potential malicious followers. These scenarios illustrate the benefits of incorporating context trajectory information into TPP mechanisms.

The trajectory model can incorporate trajectory context information using various techniques including integrating the network's scene knowledge [WTG06], the attributes of moving carriers, and trajectory uncertainty into the trajectory context information. Scene knowledge (e.g. roads, entrances, exits, bus stops, buildings) provides the background knowledge for realistic trajectory segmentation, i.e., in order to develop a trajectory segmentation method, which efficiently detects suspicious attacks, we must embed scene knowledge into the method. For example, a high correlation between two trajectories on the freeway should not be considered suspicious. On the other hand, a high correlation between two trajectories, in a particular small field or building, should be considered suspicious. Attributes of a moving carrier (e.g. size, velocity, shape, sound) can be used to identify users in heterogeneous networks. For example, an outlier in a cluster should not be considered a compromised node if the outlier is identified to be an ambulance as a result of its emergency siren.

To balance the tradeoffs between trajectory privacy, the cost associated with the data sampling rates and the accuracy of the data, application-dependent trajectory uncertainty needs to be defined, analyzed and evaluated. The inherent uncertainty of trajectory data is primarily due to sensors' physical and technical capabilities, which limit the performance of the data collection and storage processes. On one hand, accurate data at detailed levels of granularity is preferred for data collection, management and analysis for services. On the other hand, less accurate and coarse granularity [EE07] of spatiotemporal data is necessary for preventing privacy attacks and is less expensive. Furthermore, the attributes of a moving carrier can reveal the identity of the carrier. Therefore, a method to properly define the granularity levels of trajectory data must be developed.

In applications of mWSNs involving trajectory-based services, the trajectory granularity requirements of a service provider/administrators and adversaries would typically be different. Service providers/administrators require synoptic information, while attackers are interested in elementary information. Given this distinction, we would first develop different trajectory granularity levels for different entities in the network. To do so, we must first find solutions to the following open questions: What trajectory data are needed by the service provider/administrators? What information is most critical for an adversary attempting a trajectory privacy breach? We must also define an appropriate trajectory granularity level that meets the requirements of service providers/administrators without releasing sensitive data. Furthermore, in the course of trajectory data interpolation, sample suppression and compression, we must balance the tradeoff between sending critical information to service providers/administrators and risking the release secret information to potential attackers. Last, we must develop an approach to compute and evaluate the similarity between an educated trajectory guess and the real trajectory, so that we can determine the threshold of the suppression/compression limit. Once we accomplish these goals, we will develop a trajectory data model that properly defines trajectory granularity and uncertainty levels.

Our current work, the context-based trajectory privacy-aware routing protocol, is limited in that it is based solely on the dissimilarity context of trajectories. Furthermore, we have yet to define an optimal granularity that allows accurate data transmissions and prevents attackers from tracking the target node. If a context-based trajectory data model were developed and the context of trajectories were known, then suspicious trajectories of sensors (e.g. sensors that reside around a data gathering point for prolonged periods) could help detect potential attackers. Such information could then be used to build up basic trust for private data transmissions among neighboring sensors, which would also lead to less power consumption. Moreover, trajectories with properly defined granularity and uncertainty levels could prevent undetected attackers from reconstructing valid trajectories of the target. A context-based trajectory data model will fundamentally enrich the definition of trajectory and provide the groundwork for the further development of TPP mechanisms.

6.4.2 A Lightweight Energy-efficient Secure Framework

The lightweight energy-efficient secure framework is a comprehensive framework which integrates overhead and power-consumption optimization, security techniques and the TPP mechanisms.

The major constraints of mobile sensor nodes are limited storage and power support. Therefore, lightweight energy-efficient algorithms are preferred in TPP and security algorithm designs. This framework needs to consider the inherent limitations and characteristics of mWSNs. It will utilize the existing overhead and power consumption optimization techniques and security techniques to benefit the comprehensive network performance and build up a robust mWSN.

Due to the inherent characteristics of online algorithms, limited trajectory data storage needs to provide adequate trajectory information, such as historical and multiple trajectories, for online analysis and manipulation. Therefore, trajectory compression is a very promising field for developing a lightweight system framework in terms of efficient data storage utilization. Moreover, the uncertainty caused by compression can also benefit trajectory privacy protection.

Lightweight design can only extend sensors' lifetime to some extent. Further power consumption optimization is necessary. Therefore, we suggest a cross-layer localization system to assist in trajectory privacy protection. Thus far, we have overlooked the advantages of utilizing the lower layers to collect relative location information. One improvement is to deploy Cognitive Radio (CR)-based sensor networks. By deploying additional sensors for spectrum sensing, Dynamic Spectrum Access (DSA) can be enabled in sensor networks, where secondary users are allowed to operate in spectrum bands allocated to primary users on a non-interfering basis [CGCC07]. CR technology has great potential to enhance the performance of sensor networks in terms of extending the transmission range without increasing transmission power [CDWC08]. By applying this model, energy consumption will dramatically decrease.

Furthermore, privacy-aware routing algorithms will be collaborated with lower layers, which can be used to provide location information. There are techniques based on Received Signal Strength Indication (RSSI), radio hop count, Time Difference of Arrival (TDoA), Angle of Arrival (AoA), which accurately orient nodes with respect to a global coordinate system. In our system, these localization techniques will be used to sense trajectory information of neighbors in range in a privacy-aware manner. Combined with the DSA operation, this cross-layer model will serve as the basis of the energy-efficient trajectory privacy-aware localization system.

As a comprehensive framework for mWSNs, it is inevitable to consider security issues in the system due to the high vulnerability of sensors in unattended environments. Moreover, many security techniques can be used to protect trajectory privacy. We will begin with integrating security techniques, in particular, authentication and node identity encryption, into TPP to improve the security level under a variety of security attacks. Authentication is aimed to securely identify the entities in the network and prevent external online trajectory privacy attacks. Node identity encryption is a straightforward solution to hide node identities among untrustworthy peer nodes. The challenge is to develop a lightweight distributed cryptographic mechanism to make the node identities untraceable for the adversary while synchronized with the administration entity. Encryption synchronization between nodes and the administration entity is important and necessary, as the administrator needs to trace back node identities for the purposes of communications and intrusion attack detection. When a powerful adversary monitors the traffic flow all over the network, the security techniques above are of great importance to integrate into the system to achieve better security protection.

Furthermore, considering that the value of private information largely relies on the roles that describe users' functionalities in organizations and communities, we will provide the role-based service customization mechanism in this comprehensive framework. Next, the framework will be able to dynamically adjust the privacy requirements based on different users' roles, which is of great importance in the sense that users are generally identified by multiple roles in different trajectory contexts. Meanwhile, improper privacy breach behaviors of authorized users could also be detected, which is a crucial misbehavior detection method to internal attacks.

6.4.3 Privacy Preservation Features and Challenges in GeoSNs

En route to the current stage of the study, we realize the important extension of TPP in GeoSNs, along with the fast spreading of GeoSNs. Mobile devices are developed in the current of combining multiple functions of sensing environments, collecting information, processing data, communicating, entertaining and socializing. In this future, we would like to investigate the TPP issue in GeoSNs.

TPP in GeoSNs is particularly important to social network users due to the fact that trajectory information available from GeoSNs is always associated with users' social activities and such activities can be easily identified from public or semi-public resources, such as urban patterns [FRMZ11], open social events and interactions among users. As we reviewed in Chapter 2, trajectory privacy was not considered in the original design of GeoSNs and there exists a tremendous leakage. There is an ungent need to develop TPP techniques in GeoSNs.

As summarized in [VFBJ11], GeoSNs have several features in the inherent settings which may render existing TPP techniques in LBS and databases to be directly applied. 1) Some applications, such as check-ins, require exact locations with high granularity. 2) Some applications require "real-time" response, such as proximity services. 3) Users could be tagged passively, such as in photo posts and co-location check-ins. 4) Users could be re-identified through linking available background knowledge and external information.

In addition, we would like to emphasize the following two aspects: 1) The massive interaction among users could be a resource of trajectory privacy leakage. Regardless of passive tagging by friends or group members, users might reveal their trip plans or daily routines during the interaction with friends. Although the involved friends maybe trustworthy, the access to such interaction is commonly ignored by users. 2) Profiles of the same user from multiple GeoSN applications might lead to a clear "picture" of the user for re-identification. Active social network users likely join more than one GeoSNs. Although users may be aware of privacy and set privacy preferences carefully, it is very possible for attackers to link information from multiple GeoSN accounts and extract more characteristics of the target.

As indicated in the existing literature [FRVM$^+$10, VFBJ11], TPP in GeoSNs need to consider location privacy, absence privacy and co-location privacy. As the conclusion of this section, we summarize the challenges in terms of connections that need to be considered in the design of TPP techniques in GeoSNs as follows:

- The connection among users – It includes the social relationships and the interaction among users during online social activities, as we discussed above.
- The connection between the social pattern and available traces – As an entity in society, participating in social activities is very common, which forms the certain pattern in corresponding environments. On the other hand, it is possible to infer private trajectory information from available traces and the social pattern.
- The connection among different GeoSNs and other online service sites – This is an issue which has been ignored in the existing literature. However, it is a possible and effective method for attackers to obtain private data since most of the online sites require users to register services by a unique identification, and many users use the same identity for registration, such as email accounts and phone numbers. Particularly, collecting information from multiple sites, facilitated by effective searching engines, like Google and Bing, is cost-efficient in terms of time and energy consumption. Moreover, different access control policies provided by a variety of GeoSNs hinder effective preservation methods.

- The connection between the geographic map and available traces – This is a consistent issue to consider along with TPP in different applications. However, in GeoSNs, the geographic map could contain rich social contexts. Combining background knowledge and social contexts may distinguish the target from other users who have similar available traces. For example, there is an art exhibition in the museum, which is located near a church, on Sunday morning. It is likely that an art student is going to the museum and a religious person is going to the church, given that they are both heading to this region. Geographic maps associated with social contexts may further ease the inference of users' private trajectories.

6.5 Conclusion

The vast applications of location-aware mobile sensing devices will accomplish severe missions in human-unattainable environments, as well as bring significant convenience to our daily lives. However, these applications will remain elusive unless the trajectory privacy of mobile nodes is properly protected from unauthorized entities. This dissertation stemmed from the urgent need of TPP mechanisms and described an in-network TPP framework in mWSNs. The framework designed the TPriv routing protocol to prevent the external attack, developed the USAS algorithm to detect the internal attack, applied Bayesian game theory to model nodes' behaviors and evaluated the trust degree in TPP cooperations. We hope the future applications of mobile sensing devices will benefit from our research outcomes and our study will inspire better privacy preservation solutions.

BIBLIOGRAPHY

[ABN08] O. Abul, F. Bonchi, and M. Nanni. Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In *International Conference on Data Engineering*, pages 376–385, 2008.

[ACG09] C. A. Ardagna, M. C., and G. Gianini. Landscape-aware location-privacy protection in location-based services. *Journal of Systems Architecture - Embedded Systems Design*, 55(4):243–254, 2009.

[AEM$^+$07] A. Amir, A. Efrat, J. Myllymaki, L. Palaniappan, and K. Wampler. Buddy tracking - efficient proximity detection among mobile friends. *Pervasive Mob. Comput.*, 3(5):489–511, October 2007.

[Atm] Atmel. 8-bit avr microcontroller with 16k bytes in-system programmable flash. `http://www.hth.com/filelibrary/pdffiles/atmega163.pdf`. Accessed: 09/18/2013.

[Bon] Bonnmotion: A mobility scenario generation and analysis tool. `http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/`. Accessed: 06/01/2013.

[BS03] A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46 – 55, jan-mar 2003.

[BS04] A.R. Beresford and F. Stajano. Mix zones: user privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127 – 131, march 2004.

[CA12] A. K. Chorppath and T. Alpcan. Trading privacy with incentives in mobile commerce: A game theoretic approach. *Pervasive and Mobile Computing*, August 2012.

[CCL13] Z. Cheng, J. Caverlee, and K. Lee. A content-driven framework for geolocating microblog users. *ACM Trans. Intell. Syst. Technol.*, 4(1):2:1–2:27, February 2013.

[CDWC08] D. Cavalcanti, S. Das, Jianfeng Wang, and K. Challapali. Cognitive radio based wireless sensor networks. In *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on*, pages 1–6, 2008.

[CGCC07] C. Cordeiro, M. Ghosh, D. Cavalcanti, and K. Challapali. Spectrum sensing for dynamic spectrum access of tv bands. In *in Second International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, 2007.

[CM07] Chi-Yin Chow and M. F. Mokbel. Enabling private continuous queries for revealed user locations. In D. Papadias, D. Zhang, and G. Kollios, editors, *Advances in Spatial and Temporal Databases*, volume 4605 of *Lecture Notes in Computer Science*, pages 258–275. Springer Berlin Heidelberg, 2007.

[CO11] N. J. Croft and M. S. Olivier. Location privacy: Privacy, efficiency and recourse through a prohibitive contract. *Transactions on Data Privacy*, 4(1):19–30, 2011.

[COR12] J. Cuellar, M. Ochoa, and R. Rios. Indistinguishable regions in geographic privacy. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1463–1469, New York, NY, USA, 2012. ACM.

[CRRB12] B. Carbunar, M. Rahman, N. Rishe, and J. Ballesteros. Private location centric profiles for geosocial networks. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, pages 458–461, New York, NY, USA, 2012. ACM.

[CZBP06] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *In Proc. of the 6th Workshop on Privacy Enhancing Technologies*, pages 393–412. Springer-Verlag, 2006.

[DBS10] M.L. Damiani, E. Bertino, and C. Silvestri. The probe framework for the personalized cloaking of private locations. *Transactions on Data Privacy*, 3(2):123–148, 2010.

[DMNRDSHSH11] F. De Meneses Neves Ramos Dos Santos, M. Humbert, R. Shokri, and J. Hubaux. Collaborative Location Privacy with

Rational Users. In *2nd Conference on Decision and Game Theory for Security (GameSec)*, 2011. The original publication is available at www.springerlink.com.

[Du08] X. Du. Detection of compromised sensor nodes in heterogeneous sensor networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 1446 –1450, may 2008.

[EE07] F. Giannotti (Editor) and D. Pedreschi (Editor). *Mobility, Data Mining and Privac-Geographic Knowledge Discovery.* Springer, 2007.

[FMHP09] J. Freudiger, M. H. Manshaei, J. Hubaux, and D. C. Parkes. On Non-cooperative Location Privacy: A Game-theoretic Analysis. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[FRF⁺07] J. Freudiger, M. Raya, M. Felegyhazi, P. Papadimitratos, and J. Hubaux. Mix-Zones for Location Privacy in Vehicular Networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, Vancouver, 2007.

[FRMZ11] L. Ferrari, A. Rosi, M. Mamei, and F. Zambonelli. Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, pages 9–16, New York, NY, USA, 2011. ACM.

[FRVM⁺10] D. Freni, C. Ruiz Vicente, S. Mascetti, C. Bettini, and C. S. Jensen. Preserving location and absence privacy in geo-social networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 309–318, New York, NY, USA, 2010. ACM.

[FSH09] J. Freudiger, R. Shokri, and J. Hubaux. On the optimal placement of mix zones. In *in Privacy Enhancing Technologies, 2009*, pages 216–234, 2009.

[FT91] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, 1991.

[GDVM09] A. Gkoulalas-Divanis, V. S. Verykios, and M. F. Mokbel. Identifying unsafe routes for network-based trajectory privacy. In *SDM*, pages 942–953, 2009.

[Ghi09] G. Ghinita. Private queries and trajectory anonymization: a dual perspective on location privacy. *Transactions on Data Privacy (TDP)*, 2:3–19, 2009.

[GHP11] S. Gambs, O. Heen, and C. Potin. A comparative privacy analysis of geosocial networks. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '11, pages 33–40, New York, NY, USA, 2011. ACM.

[GKK$^+$08] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan. Private queries in location based services: anonymizers are not necessary. In *International Conference on Management of Data*, pages 121–132, 2008.

[GKS07] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In *Proceedings of the 10th international conference on Advances in spatial and temporal databases*, SSTD'07, pages 221–238, Berlin, Heidelberg, 2007. Springer-Verlag.

[GL08] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *Mobile Computing, IEEE Transactions on*, 7(1):1 –18, jan. 2008.

[Gol04] O. Goldreich. *The foundations of Cryptography C Volume 2.* Cambridge University Press, 2004.

[GS05] R. H. Guting and M. Schneider. *Moving Objects Databases.* Morgan Kaufmann, 2005.

[GX04] M. Gruteser and L. Xuan. Protecting privacy in continuous location-tracking applications. *Security Privacy, IEEE*, 2(2):28 – 34, mar-apr 2004.

[HBH05] C. Hartung, J. Balasalle, and R. Han. Node compromise in sensor networks: The need for secure systems. Technical report, 2005.

[HG05] B. Hoh and M. Gruteser. Protecting Location Privacy Through Path Confusion. In *International Workshop on Security*, 2005.

[HGXA07] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 161–171, New York, NY, USA, 2007. ACM.

[HHJC12] A. Hossain, A. A. Hossain, S. Jang, and J. Chang. Privacy-aware cloaking technique in location-based services. In *Mobile Services (MS), 2012 IEEE First International Conference on*, pages 9 –16, june 2012.

[HMFH10] M. Humbert, M. Manshaei, J. Freudiger, and J. Hubaux. Tracking games in mobile networks. In T. Alpcan, L. Buttyan, and J. Baras, editors, *Decision and Game Theory for Security*, volume 6442 of *Lecture Notes in Computer Science*, pages 38–57. Springer Berlin Heidelberg, 2010.

[HMYS05] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki. Enhancing wireless location privacy using silent period. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 2, pages 1187 – 1192 Vol. 2, march 2005.

[JJFZ11] P. Jagtap, A. Joshi, T. Finin, and L. Zavala. Privacy preservation in context aware geosocial networking applications. Technical report, University of Maryland, Baltimore County, May 2011.

[JLJ12] L. Jin, X. Long, and J. B. D. Joshi. Towards understanding residential privacy by analyzing users' activities in foursquare. In *Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security*, BADGERS '12, pages 25–32, New York, NY, USA, 2012. ACM.

[JPC$^+$12] X. Jin, N. Pissinou, C. Chesneau, S. Pumpichet, and D. Pan. Hiding trajectory on the fly. In *Communications (ICC), 2012 IEEE International Conference on*, pages 403 –407, june 2012.

[JPP$^+$10] X. Jin, P. Putthapipat, D. Pan, N. Pissinou, and S. K. Makki. Unpredictable software-based attestation solution for node compromise detection in mobile wsn. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 2059–2064, 2010.

[JPP+13] X. Jin, N. Pissinou, S. Pumpichet, C. A. Kamhoua, and K. Kwiat. Modeling cooperative, selfish and malicious behaviors for trajectory privacy preservation using bayesian game theory. In *Conference on Local Computer Networks (LCN), 2013 IEEE*, 2013.

[KGMP07] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719 –1733, dec. 2007.

[KPM11] C.A. Kamhoua, N. Pissinou, and K. Makki. Game theoretic modeling and evolution of trust in autonomous multi-hop networks: Application to network security and privacy. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1 –6, june 2011.

[KSE08] C. Krauß, M. Schneider, and C. Eckert. On handling insider attacks in wireless sensor networks. *Information Security Technical Report*, 13(3):165 – 172, 2008.

[KXTZ07] P. Kamat, W. Xu, W. Trappe, and Y. Zhang. Temporal privacy in wireless sensor networks. In *Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on*, page 23, june 2007.

[KYS05] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on*, pages 88 – 97, july 2005.

[KZTO05] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 599 –608, june 2005.

[LB07] B. Li and L. Batten. Using mobile agents to detect node compromise in path-based dos attacks on wireless sensor networks. In

Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, pages 2507 –2510, sept. 2007.

[LHX12] H. Li, H. Hu, and J. Xu. Nearby friend alert: Location anonymity in mobile geo-social networks. *Pervasive Computing, IEEE*, PP(99):1–1, 2012.

[LR10] Y. Li and J. Ren. Source-location privacy through dynamic routing in wireless sensor networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[LSHP06] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran. Swing and swap: user-centric approaches towards maximizing location privacy. In *In Proceedings of the 5th ACM WPES 06*. ACM Press, 2006.

[LZDT09] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Ad Hoc Netw.*, 7(8):1501–1514, November 2009.

[LZL$^+$06] X. Lin, H. Zhu, B. Lin, P. Ho, and X. Shen. Nis01-5: A novel voting mechanism for compromised node revocation in wireless ad hoc networks. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1 –6, 27 2006-dec. 1 2006.

[MAA$^+$10] A. Monreale, G. L. Andrienko, N. V. Andrienko, F. Giannotti, D. Pedreschi, S. R., and S. Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.

[MCA06] M. F. Mokbel, C. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *In VLDB*, pages 763–774, 2006.

[McS09] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 19–30, New York, NY, USA, 2009. ACM.

[MFB$^+$10] S. Mascetti, D. Freni, C. Bettini, X. Sean Wang, and S. Jajodia. Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. *CoRR*, abs/1007.0408, 2010.

[MGKV06] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, page 24, april 2006.

[MJ11] A. Masoumzadeh and J. Joshi. Anonymizing geo-social network datasets. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '11, pages 25–32, New York, NY, USA, 2011. ACM.

[MLW12] K. Mehta, D. Liu, and M. Wright. Protecting location privacy in sensor networks against a global eavesdropper. *Mobile Computing, IEEE Transactions on*, 11(2):320 –336, feb. 2012.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and Computing:Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[Mye97] R. B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, 1997.

[NAS08] M. E. Nergiz, M. A., and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *Workshop on Advances in Geographic Information Systems*, pages 52–61, 2008.

[Osb04] M. J. Osborne. *An introduction to Game Theory.* Oxford University Press, 2004.

[PL11] K. Pongaliur and X. Li. Maintaining source privacy under eavesdropping and node compromise attacks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1656 –1664, april 2011.

[PL12] A. Proano and L. Lazos. Hiding contextual information in wsns. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1 –6, june 2012.

[PLP08] S. Peter, P. Langendörfer, and K. Piotrowski. Public key cryptography empowered smart dust is affordable. *International Journal of Sensor Networks*, 4:130–143, 2008.

[PMM] F. Provost, D. Martens, and A. Murray. Finding similar users with a privacy-friendly geo-social design. `http://www.everyscreenmedia.com/everyscreenmedia/wp-content/uploads/2012/10/Finding_Similar_Users.pdf`. Accessed: 12/09/2013.

[PMV+12] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida. Beware of what you share: Inferring home location in social networks. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 571–578, 2012.

[PVA+12] T. Pontes, M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida. We know where you live: privacy characterization of foursquare behavior. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 898–905, New York, NY, USA, 2012. ACM.

[PWS+12] K. P. N. Puttaswamy, S. Wang, T. Steinbauer, D. Agrawal, A. E. Abbadi, C. Kruegel, and B. Y. Zhao. Preserving location privacy in geo-social applications. *IEEE Transactions on Mobile Computing*, 99(PrePrints):1, 2012.

[Q-s] Q-sensor. `http://www.affectiva.com/q-sensor/`. Accessed: 25/02/2013.

[RB12] D. Riboni and C. Bettini. Private context-aware recommendation of points of interest: An initial investigation. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 584–589, 2012.

[Rob] Please rob me. `http://pleaserobme.com/`. Accessed: 12/09/2013.

[SPvDK04] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. Swatt: software-based attestation for embedded devices. In *Security and*

Privacy, 2004. Proceedings. 2004 IEEE Symposium on, pages 272 – 282, may 2004.

[STLBH11] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux. Quantifying location privacy. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 247 –262, may 2011.

[STT$^+$12] R. Shokri, G. Theodorakopoulos, C. Troncoso, J. Hubaux, and J. Le Boudec. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 617–627, New York, NY, USA, 2012. ACM.

[Swe02] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[SXZC07] H. Song, L. Xie, S. Zhu, and G. Cao. Sensor node compromise detection: the location perspective. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, IWCMC '07, pages 242–247, New York, NY, USA, 2007. ACM.

[SYZC08] M. Shao, Y. Yang, S. Zhu, and G. Cao. Towards statistically strong source anonymity for sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 51 –55, april 2008.

[VFBJ11] C.R. Vicente, D. Freni, C. Bettini, and Christian S. Jensen. Location-related privacy in geo-social networks. *Internet Computing, IEEE*, 15(3):20–27, 2011.

[WCK09] W. Wang, M. Chatterjee, and K. Kwiat. Coexistence with malicious nodes: A game theoretic approach. In *Game Theory for Networks, 2009. GameNets '09. International Conference on*, pages 277 –286, may 2009.

[WTG06] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. In *Proceedings of the 9th European conference on Computer Vision - Volume Part III*, ECCV'06, pages 110–123, Berlin, Heidelberg, 2006. Springer-Verlag.

[XC09] T. Xu and Y. Cai. Location cloaking for safety protection of ad hoc networks. In *INFOCOM 2009, IEEE*, pages 1944 –1952, april 2009.

[XSS06] Y. Xi, L. Schwiebert, and W. Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 8 pp., april 2006.

[YBLW09] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and H. Wang. Anonymizing Moving Objects: How to Hide a MOB in a Crowd? In *International Conference on Extending Database Technology*, pages 23–26, March 2009.

[YCM06] Y.Liu, C. Comaniciu, and H. Man. A Bayesian game approach for intrusion detection in wireless ad hoc networks. In *ACM International Conference Proceeding Series*, 2006.

[YJHL08] M. Yiu, C.S. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 366 –375, april 2008.

[YSZ+08] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao. Towards event source unobservability with minimum network traffic. In *in sensor networks, The ACM Conference on Wireless Network Security (WiSec)*, 2008.

[YWZC07] Y. Yang, X. Wang, S. Zhu, and G. Cao. Distributed software-based attestation for node compromise detection in sensor networks. In *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*, pages 219 –230, oct. 2007.

[ZSJ03] S. Zhu, S. Setia, and S. Jajodia. Leap: efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and communications security*, CCS '03, pages 62–72, New York, NY, USA, 2003. ACM.

VITA

XINYU JIN

September 21, 1985	Born, Qinyang, China
2007	B.E., Telecommunication Engineering Xi'an University of Posts & Telecoms Xi'an, China
2012	M.S., Computer Engineering Florida International University Miami, Florida
2013	Ph.D. Candidate, Electrical Engineering Florida International University Miami, Florida

PUBLICATIONS AND PRESENTATIONS

Jin, X., Pissinou, N., Pumpichet, S., Kamhoua, and C., Kwiat, K., (2013). "Modeling Cooperative, Selfish and Malicious Behaviors for Trajectory Privacy Preservation using Bayesian Game Theory," in proceedings of *The 38th IEEE Conference on Local Computer Networks (LCN)*, Oct. 2013.

Pumpichet, S., Jin, X., Pissinou, N., (2013). "Sketch-based Data Recovery in Sensor Data Streams," *The 19th IEEE International Conference on Networks (ICON)*, Dec. 2013. (to appear)

Munavalli, S. C., Pissinou, N., Lagos, L. E., and Jin, X., (2013). "Structural Damage Detection of Nuclear Reactor Sites Using Sensor Networks," *IEEE Sensors 2013*, Nov. 2013. (to appear)

Jin, X., Pissinou, N., Chesneau, C., Pumpichet, and S., Pan, D., (2012). "Hiding Trajectory on the Fly," *The IEEE International Conference on Communications (ICC)*, pp. 403-407. Jun. 2012.

Pumpichet, S., Pissinou, N., Jin, X., and Pan, D., (2012). "Belief-based Cleaning for Trajectory Sensor Streams," *The IEEE International Conference on Communications (ICC)*, pp. 208-212. Jun. 2012.

Jin, X., and Pissinou, N., (2012). "Trajectory Privacy Preservation-an Inevitable Issue Towards Future Mobile Sensor Networks," *International Journal of Research and Reviews in Wireless Sensor Networks (IJRRWSN)*, vol. 2, no. 2, 2012.

Jin, X., Putthapipat, P., Pissinou, N., Pan, D., and Makki, K. S., (2010). "Unpre-

dictable Software-based Attestation Solution for Node Compromise Detection in Mobile WSNs," *The IEEE Global Communications Conference Workshops (GLOBECOM Wkshps)*, pp. 2059-2064. Dec. 2010.

Liu, Y., Fang, Q., and Jin, X., (2007). "Channel Characteristics of Stimulated Raman Scattering with Dispersion Effect in Optical Fiber Communication System," *Journal of Applied Optics*, 28(5) 608-613, ISSN: 1002-2082, 2007.