PRIVACY-PRESERVING COMMUNICATION PROTOCOLS FOR AUTHENTICATED LOCATION-BASED SERVICES IN MOBILE NETWORKS

by

Qin Zhang

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTERS OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2011

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____
Qin Zhang

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

_____ _____
Loukas Lazos
Assistant Professor

Date

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my academic advisor Dr. Loukas Lazos. Your passion for your work encouraged me to pursue my own goals, while your attention to details instilled in me a unique perspective on which to approach problems. I am deeply grateful for your taking me on as a student. Our academic work together has been a truly rewarding and enriching experience, while the lessons learned will help guide me throughout the course of my life in whatever avenue I pursue.

I would also like to thank the members of my defense committee, Dr. Marwan Krunz and Dr. Bane Vasic, for both supporting my degree goals and the valuable time spent in the classroom.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

ABSTRACT

The pervasiveness of mobile devices equipped with positioning capabilities has led to the emergence of numerous location-based applications and services. Using mobile network infrastructures, mobile users can rapidly gain access to a wealth of information by connecting to a variety of services. A large fraction of the information sought is related to the current user's position. This includes queries for nearby medical services, specialized stores, social activities and groups, and others. In general, location-based service operators are assumed to be trusted parties that preserve the user's privacy. However, due to the sensitive nature of the information accessed by these parties and repeated information leakages that have been recorded, the privacy of users that access location-based services is at risk.

User's privay can be breached by linking one's identity, location, and query content. On certain scenarios, knowing one's location is sufficient to derive his identity (e.g. if this location is the user's residence, office, etc.). In this thesis, we address the problem of preserving the location privacy and user anonymity of mobile users accessing authenticated location-based services. We design novel communication protocols that preserve the user privacy without relying on any trusted entity. At the same time, our protocols allow the service provider to authenticate any authorized requesting location-based services.

CHAPTER 1

Introduction

1.1 Motivation and Scope

The pervasiveness of mobile devices in today's social and commercial activities has led to the proliferation of numerous user-centric applications and services. Using various wireless technologies including cellular [28], WiFi [29], and WiMax [1] to name a few, users can remain continuously connected to a web of services and information banks irrespective of their location. Moreover, equipping the mobile devices with positioning capabilities, either by embedding a GPS receiver, or exploiting the reception of radio signals of known origin [29], has enabled the precise determination of ones's location. Sharing this location with service providers has given rise to a wealth of personalized services that exploit knowledge of one's position, typically referred to as, Location-Based Services. These services include but are not limited to the discovery of points of interest, localized traffic and weather updates, provision of navigation instructions and social networking [15, 31, 52].

In a location-based service scenario, the user consents to disclose his location, in exchange for receiving location-relevant information such as listings of nearby restaurants, medical care providers, or lists of people with specific interests. Because of the generality of the types of information that can be requested, oftentimes this information can be highly sensitive, the disclosure of which can lead to a major breach of a user's privacy rights. For instance, a user requesting listings of highly specialized medical care providers may implicitly reveal an existing medical condition to the information service provider. Thus breach of privacy can occur on many levels, including the privacy of the information sought by users and the privacy of

the users' whereabouts.

Typically, location service providers are assumed to be trusted entities that have entered into legal agreement with users for the non-disclosure of private data. However, the providers themselves may exploit private data to profile users and improve on their marketing strategies, or fall victims of security breaches in which case users' records are illegally retrieved by unauthorized parties [3, 20].

Due to the increased privacy concerns raised by authenticated information retrieval, several location-based services are provisioned anonymously. The concept of anonymity states that an individual shall not be identifiable among the set of users. Intuitively, protecting the sender's anonymity by excluding his identity from this communication with a location-based server (LBS) should be sufficient to disassociate the sender's identity from the nature and contents of his queries.

However, even if the user's identity is omitted from the LBS-user interaction, knowledge of one's location may be sufficient to link the user to his identity and eventually, his queries. For example, a user submitting a query from his private residence becomes uniquely identifiable if his position is known with high accuracy.

Furthermore, for location-based services that implement a subscription or pay-per-view pricing model, implicit or explicit authentication is required. Anonymity and authentication requirements are seemingly antithetic goals since one requires the concealment of one's identity and the other requires the disclosure. *In this thesis, we address the problem of preserving the location privacy and user anonymity of mobile users accessing authenticated location-based services.* Previously proposed solutions for protecting location privacy rely on spatial cloaking mechanisms [21]. The basic idea of spatial cloaking is to "blur" the user's exact location to a cloaking region (CR) which meets the user's privacy requirements. These requirements are expressed by the number of users within the CR and the size of the CR. The former metric reflects the anonymity degree k, i.e., the size of the set of users for which the user submitting a query becomes indistinguishable. The latter express the level of

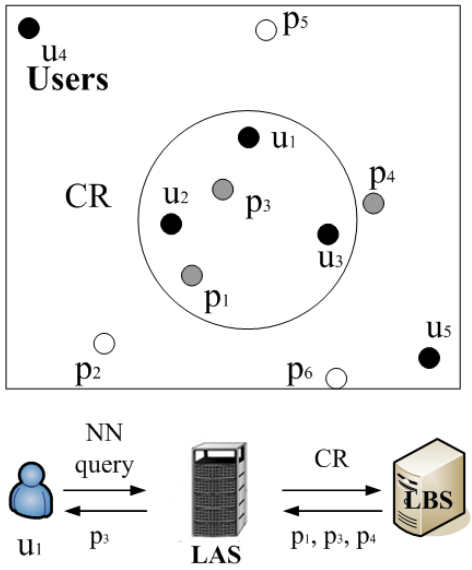

Figure 1.1: An example of achieving spatial cloaking via a LAS. When the user u_1 sends a query to LBS, a cloaking region with three users u_1, u_2, u_3 is formed and sent to LBS via the LAS. The LBS sends the answer set $\{p_1, p_3, p_4\}$ back to the LAS. The LAS distributes the corresponding answers to the users that submitted the queries.

accuracy with which a user's identity becomes known to the LBS.

The majority existing methods for spatial cloaking and query anonymization rely on a trusted third party known as the location anonymizer server (LAS) [2, 21, 33, 37, 48]. The LAS collects queries from multiple users, removes all identity related information from the queries, computes the spatial cloaking region and submits the queries to the LBS. The LBS serves the queries without being able to match the users' locations with their queries and their identities. The LBS computes a candidate set of responses to the submitted queries (e.g. a set of points-of-interests (POI)) and sends the responses to the LAS [24]. The LAS filters the responses according to the exact location of the users and provides them with the appropriate information. An example of the implementation of a location-based service via a LAS is shown in Figure 1.1.

Using an intermediate trusted party for query anonymization purposes has several drawbacks. First the LAS remains as a single point of failure. The breach of the security of the LAS exposes users' privacy. Methods relying on the existence of a LAS move the problem of trust from the LBS to the LAS. The users' location privacy is not preserved with respect to that party. Moreover, the LAS becomes the bottleneck in terms of system scalability, since it must process frequent updates of users' locations, anonymize all queries and filter the results.

To address the above shortcomings, several researchers have proposed decentralized anonymization schemes that do not rely on the existence of LASs [10, 11, 19, 26, 35]. Decentralized schemes remove the placement of trust on a third party by collectively creating a spatial cloaking region and anonymizaing queries in a peer-to-peer(P2P) fashion. However, note that users need to disclose their locations to their group in order to perform peer discovery and CR generation. Users' location privacy is not preserved among group members. Moreover, existing methods do not address the problem of collusion among the participating parties and how to authenticate users. We develop methods that preserve the location privacy even among the group participants and solve the authentication problem at the same time. In particular, we make the following contributions.

1.2 Main Contributions and Thesis Organization

We design a novel location privacy preserving and query anonymization scheme called MAZE. Our scheme does not require the existence of a LAS, but achieves its privacy properties in a decentralized manner using P2P groups. With respect to privacy, MAZE achieves the following properties:

(a) the location of any user cannot be defined to an accuracy greater than a predefined level.

(b) user queries are k-anonymized.

(c) the LBS can authenticate every user requesting service for charging purposes.

MAZE guarantees properties (a) and (b) against the LBS, the participating users or any colluding set of system participants. Moreover, it allows for user authentication, a property seemingly antithetic with the notion of k-anonymity. To achieve this goal, we obfuscate users' location to a proper CR without asking users' exact location and we disassociate users' identity with from the contents of their queries. The properties of MAZE are analytically shown. Furthermore, we evaluate the performance of MAZE in terms of communication overhead and verify our analysis via extensive simulations.

The remainder of the thesis is organized as follows. Chapter 2 highlights the related work. The problem statement and model assumptions are presented in Chapter 3. The details of the MAZE protocol are described in Chapter 4. In Chapter 5, we present L-MAZE, version of MAZE that is resistant to colluding adversaries. In Chapter 6, we analytically evaluate the communication overhead of MAZE and L-MAZE. We experimentally verify our analysis in Chapter 7, and summarize our conclusions in Chapter 8.

CHAPTER 2

Related Work

In this chapter, we present related work on the problem of user privacy in mobile environments.

2.1 Definition of Privacy

User privacy refers to the protection of users from unauthorized accessing, storing, repurposing and displaying of personal information from a third party. Threats on privacy can take many forms including the disclosure of personally identifiable data, the identification and tracking of one's whereabouts, profiling of user habits, and even disclosing one's communication patterns (when and/or with whom communication took place).

Users that receive mobile services are particularly vulnerable to privacy threats, since their communication is carried out via the open wireless medium. An adversary capable of taping the wireless medium breach the privacy of the information transmitted over the air, pinpoint a user's location by intercepting signals that contain low-level identifiers [20], and identify a user based on his location [21]. Privacy concerns become more prominent in pervasive computing environments where a pre-deployed and typically trusted infrastructure is absent. In such environments, users have to collaborate in order to communicate, thus entrusting their private information to other users [20]. While information privacy can be protected using cryptographic methods, location privacy, communication privacy and anonymity are not easy to guarantee. In fact, the latter is oftentimes antithetic to notions of

security such as authentication, which requires the identification of users for the provision of services.

2.2 Privacy Protection Methods

There are several existing approaches to protect a user's privacy. The purpose of these approaches is to prevent the disclosure of user's information, which primarily refers to the individual's identity and location information. The different strategies that exist for protecting one's location privacy can be classified into three categories: regulatory, privacy policies, and anonymity strategies [14].

Regulatory strategies - normally government rules that dictate on the use of personal information, which call and enforce regulations for fair use of private information [14].

Privacy policies - trust-based agreements between individuals and the private party that gains access to private information [14].

Anonymity - anonymous access of services that prevent the disclosure of the user's identity in a direct or indirect manner.

This thesis focuses on the provision of anonymous but authenticated services. Next we provide a detailed description of the three privacy protection strategies.

2.2.1 Regulatory Strategies

The five principles of fair information practices given below are the core of most privacy regulations, (originally developed as the basis of the US privacy legislation [4, 14]). First, individuals must be aware of who is collecting personal information about them and for what purpose. Second, individuals must consent to personal information being collected for particular purposes and ensure that the use of personal information is limited to those purposes. Individuals must be able to access stored personal data which refers to them, and may request at any time for any errors to

be corrected. Collectors must ensure that personal data is accurate and up-to-date and protect against unauthorized access, disclosure, or use. Finally, collectors must be accountable for any failures to comply with the other principles.

The concept of fair information practices is also applied specifically to location privacy. For location privacy, except the above five principles, the basic principles for handling location information are given a document, drafted by the Organization of Economic Co-operation and Development(OECD) [27]. The main privacy guidelines are:

Collection Limitation: only necessary information should be provided for obtaining required services.

Consent: A LBS has to seek the agreement of the user before his location can be collected. This is called the opt-in principle where the user has to give his consent before data is collected or disclosed.

Usage and disclosure: The processing and disclosure of location data shall be limited to what consent is given for. If there's no need for LBS to know the user's true identity, pseudonyms should be used.

Security safeguard: After using LBS services, location data should be erased or be anonymized so that individuals cannot be identified.

These principles serve as the basic guideline for LBS to handle location data. Currently, two important legislations related to privacy are in effect. In the US, privacy laws are outlined in the Privacy Act of 1974 [4]. In the European Union, similar laws are described in Derective 95/46/EC [13]. The US Privacy Act of 1974 was designed for information privacy. It gives legal substance to the idea of fair information practives including openness and transparency. For example, no secret record keeping, individual participation, collection and use limitations, reasonable security and accountability [38].

Regulatory strategies state how to handle intrusions and give a powerful mean of governing privacy, however, they cannot prevent intrusion of privacy. Development

of regulations may lag behind technological progress. Moreover, it is hard to keep a balance between new technical features and privacy protection goals. It is also hard to restrict everyone to follow and implement these principles in practise. LBS are used across different countries. Therefore, regulatory strategies are not enough.

2.2.2 Privacy Policies

In general, all policy-based approaches require that users place trust on the system [14]. Policy is the agreement the system promises to follow when providing the LBS service. In order to protect a user's privacy, the user should have control over who may know his whereabouts. However, when this information is revealed to the LBS, the user has lost his control. Thus, before location information can be used by a LBS, the LBS's policies need to be verified for compliance with the user's preferences and privacy setting requirements.

The essential elements of setting privacy policies are studied in [38]. These are: the location of an object, the identity of an object, the time this observation occurs, and the speed of the object [38]. For example, a nurse named Alice who carries a mobile device allows her boss to locate her during work time with a resolution of 100 meters in order to let her boss control how fast she is able to arrive when a patient needs help. When she is off duty, she does not allow her company to locate her. She may use location-based services to receive reports of traffic conditions within one mile around her position on her way home and let her friends know her location when she is out for dinner. This example shows that in different conditions, a user has different preferences and has different trust levels on different entities. Thus, the amount of information disclosed differs.

Especially in pervasive computing environments, where personalized computing power is freed from a stationary position, enabling information access everywhere, anytime, and on demand and providing location services while protecting the user's privacy is a big challenge. Since location based services can be accessed from any-

place, there is no single point that needs to be protected. Second, the devices in pervasive environments may be managed by different administrators. Policies may changes over places and over time. Every time the policy changes, the system needs to ask the user to give approval to the new policy and to release the location information. This overhead appears a nuisance to the user. In [38], the problem meeting the needs of users for protecting location privacy while minimizing the required interactions is addressed. A system is developed using machine-readable privacy policies and user preferences to automate the privacy management decision-making process [38]. An access control mechanism based on digital certificates is developed in [23]. For example, If Bob is allowed to retrieve Alice's location information, then there must be a certificate specifying whether and what type of information Bob can retrieve. When Bob tries to locate Alice, a service receiving his location request needs to check the existence and validity of a certificate permitting this access. In [30], Langheinrich proposes a Privacy Awareness System (PAWS) for ubiquitous computing environments. It provides users with a privacy enabling technology. A language named APPEL, an XML-language, is used for specifying a user's privacy requirements when browsing the web. The P3P (Platform for Privacy Preference Project) [12], framework enables the encoding of privacy policies into machine-readable XML code. These policies are used to describe the requirements for privacy issues [38] in PAWS. Then using a trusted device, the user negotiates his privacy preferences with the UbiCom environment.

All the policies we discussed in this section assume that the LBS will strictly follow the policies they provide to the users. However, privacy policies are not effective against malicious providers who to break the privacy policies.

2.2.3 User Anonymity

Instead of trusting the policies setforth by the LBS, there is a more effective way of preventing private information disclosure before it is sent out to the LBS which is

based on communication anonymity. The basic idea is hiding the information that can lead the traceback of the users, such as the user identity. Instead, communication can occur using pseudonyms. Intuitively, using a pseudonym can disassociate a user with real identity. However, after tracking a user with the same pseudonym for sufficient amount of time, adversaries are still able to profile and link pseudonym to the victim's real identity. Beresford and Stajano [3] proposed the idea of frequently changing the pseudonym of each user, such that the chance of being tracked by an adversary is reduced. Users cannot change their identities anytime, since this could lead to the linkage between old and new pseudonyms. The change of pseudonym should be spatially and temporary restricted inside a region called mix-zone where there are other users changing pseudonyms as well. However, pseudonyms are costly to acquire. Moreover, changing pseudonyms is not effective against global adversaries. If an adversary can hear all communications in the network, then the movement of a user can be predicted according to position, speed, movement pattern etc., thus different pseudonyms can be linked with high probability [5]. Furthermore, for LBS, even it is able to hide a user's identity, it still cannot solve the problem of location privacy. Adversaries are able to link a user's pseudonym to its real identity when he sends out a query from a sensitive location like his private residence.

Thus, location information also should be anonymized by reducing the temporal and spatial resolution of the user's location, which is called spatial cloaking. The basic idea of spatial cloaking technique is to "blur" the user's exact location into a cloaked area with at least $(k-1)$ other users. This is a concept called k-anonymity which was first developed for protecting published medical data [46]. k anonymity is defined as "A dataset is said to be k-anonymized, if each record is indistinguishable from at least k-1 other records with respect to certain identifying attributes [19, 43]". This notation was first used for preserving location privacy in [21].

Several works improved upon the method for computing the spatial cloaking region in mobile environments. These works can be classified to centralized schemes

[2, 21, 33, 37, 48] , and decentralized ones [10, 11, 19, 26, 35].

In centralized schemes, query anonymization and spatial cloaking are achieved at a trusted third party known as the *location anonymizer server* (LAS) [2, 21, 33, 37, 48]. The LAS is responsible for collecting users' queries including their exact locations, and transforming those locations to a cloaked region that satisfies the desired k-anonymity requirement. The LAS then forwards the queries along with the cloaked region to the LBS which responds with multiple replies based on the entire cloaked area. The LAS filters the replies and distributes the optimal answers to the users. The size of the cloaked region and degree of anonymity establish a tradeoff with the accuracy of the returned results.

To improve upon this accuracy, several cloaking algorithms have been proposed [2, 17, 19, 37]. In [19, 37], the proposed cloaking algorithms focus on computing the smallest area that satisfies the k-anonymity requirement. In [2], two dynamic grid cloaking algorithms, a bottom-up and a top-down algorithm are proposed, aiming at achieving a high degree of accuracy in a resource-efficient manner. The CliqueCloak algorithm assumes that users may have different k-anonymity requirements [17]. It then combines these requirements to form clique graphs of users that can share the same cloaked region. The Casper scheme introduced a privacy-aware query processor embedded in the LBS, capable of handling queries that include spatially cloaked regions and not exact locations [37].

Compared to MAZE, centralized schemes rely on the existence of a trusted entity, therefore placing explicit trust on a single party. On the other hand, in MAZE, none of the system participants is assumed to be trusted. Moreover most previous works focus on constructing efficient methods for computing the cloaking region. Our work is complimentary, attempting to satisfy the antithetic requirements of privacy and authentication.

Decentralized schemes remove the requirement for the existence of a trusted LAS [10, 11, 35]. P2P models have been proposed in [10, 11]. In these modes, users form

a P2P group that meets the desired k-anonymity requirement. The spatial cloaking region is collaboratively computed based on the users' exact locations. One of the P2P group participants is randomly selected to act as an agent, responsible for forwarding the group queries to the LBS and distributing the responses to individual users. Since the exact users' locations are needed to compute the spatial cloaking region, previous P2P methods do not preserve location privacy within the P2P group. Moreover, these methods do not consider the problem of authenticated services. Only the id of the agent user becomes known to the LBS, thus preventing the authentication of the entire P2P group submitting queries. In [35], the authors proposed a distributed privacy preserving scheme for location-base queries that considers P2P as potential threats to users' privacy. In this scheme, users submit queries via a set of peers that act like mixes and re-encrypt a message before sending it to the LBS provider. The authors also discuss the idea of homomorphisms for the purpose of aggregating multiple queries into single messages. However, the scheme in [35] does not take into account the collusion of the LBS with users of a P2P group. Moreover, because accurate location information is included with every query, the user identity is disclosed if that location is sensitive (such as one's home).

Several other methods employing cryptographic techniques have been proposed in the broader context of privately answering questions. The scheme proposed in [18] supports private location dependent queries, based on the theoretical work on Private Information Retrieval (PIR) [9]. The protocol based on PIR, allows users to privately retrieve information from a database, without this information being revealed to the database server. In this scheme, no third-trusted party is required and the user's identity (or location) remains secret. However, communication and computation requirements are high and large portions of the server's data are exposed to the user with successive queries. The associated resource overhead is alleviated in [39], [49] by developing a two-level PIR system, and the combination of PIR with an oblivious transfer protocol, respectively. Other cryptographic techniques

for anonymizing queries and achieving user authentication employ group signatures [8] and ring signatures [42]. Group signature techniques rely on a central authority for setting up and managing the group membership and associated pseudonyms, thus adding considerable communication overhead [16]. Ring signatures allow users to use a ring of pseudonyms for creating their own privacy cloaks. A graph-theoretic model was developed in [16] in order to evaluate different ring construction strategies.

CHAPTER 3

Problem Statement and Model Assumptions

In this chapter, we state the problem addressed in this work and list our model assumptions. For clarity, the notation adopted in the rest of the thesis is presented in Table 3.1.

Table 3.1: Notation.

u_i	:	identity of user i
q_i	:	query submitted by u_i
r_i	:	a response to a query q_i
l_i	:	location of u_i
k_i	:	anonymity level of u_i
k-ASR	:	k-anonymizing spatial region
α	:	k-ASR area radius
l_c	:	the center of k-ASR
d	:	user density
$\hat{u_i}$	:	the group initiator
ϕ	:	privacy resolution tolerance
$P_i = \{k_i, \phi_i\}$	:	privacy profile of u_i
$\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$	:	A user set of size k
$\mathcal{Q} = \{q_1, q_2, \ldots, q_k\}$	:	A set of queries submitted by $\mathcal{U}$
$\mathcal{R} = \{r_1, r_2, \ldots, r_k\}$	:	A set of responses to a query set $\mathcal{Q}$
$\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$	:	A set of transformation applied by $\mathcal{U}$
$\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$	:	A set of transformation applied by the LBS

3.1 Problem Statement

We address the problem of preserving the privacy of users who seek location-based services from untrusted servers. The main goal of our system is to disassociate the

identity of a user u_i submitting a query q_i to a LBS, from the contents of q_i and the location ℓ_i of u_i. To quantify the privacy level achieved by our solution, we adopt the metric of k-anonymity [40]:

Definition 1. *k-anonymity: A dataset is said to be k-anonymized, if each record is indistinguishable from at least $(k-1)$ other records with respect to identifying attributes of interest. In our context, a query submitted by a user u_i is said to be k-anonymized, if u_i is indistinguishable from the identities of at least $(k-1)$ other users.*

To prevent the association of a location-based query q_i with the identity of a user u_i by exploiting the correlation of his location with his identity u_i (e.g., a query issued from a user's private residence leaks his identity), the user's exact location is obfuscated to a larger area known as the *k-anonymizing spatial region* (k-ASR). This region is defined as the minimum area with a radius α that contains at least $(k-1)$ other users, so that the k-anonymity of any user within the k-ASR is preserved. However, for areas of high user density the k-ASR can acquire very small values, thus compromising the location privacy of the user. For this purpose, we also adopt the metric of *privacy resolution tolerance* ϕ, defined as follows:

Definition 2. *Privacy resolution tolerance (PRT): The privacy resolution tolerance ϕ is defined as the set of candidate locations ℓ_i of a user u_i submitting a query q_i to an LBS.*

From the definitions of k-ASR and PRT, it holds that $\alpha \geq \phi$. In other words, the k-ASR cannot become smaller than the location resolution tolerated by the user. The parameter ϕ is user customizable.

Based on the aforementioned privacy metrics, we design our system to satisfy the following requirements:

1. The queries submitted by any user u_i must be k-anonymous.

2. The location of any user u_i cannot be determined beyond the PRT level ϕ.

3. The LBS can authenticate all queries originating from legitimate users (subscribers).

4. The LBS can charge a user u_i, for obtaining service to a submitted query q_i.

Note that our system must be designed to satisfy seemingly antithetic goals. On one hand, the anonymity of the user must be preserved in order to protect his privacy. On the other hand, a user requesting any service must be authenticated and uniquely identified in order to be charged.

3.2 User Privacy Profile

The privacy profile P_i of a user u_i consists of the 2-tuple $< k_i, \phi_i >$. Ideally, each user can select exact values for k_i, and ϕ_i. However, from a usability perspective, the majority of users will not be able to correlate the profile parameters to a mental perception of privacy. For this reason, we assume that privacy profile can be customized to a distinct and finite set of privacy levels $\mathcal{P} = \{P^a, P^b, \ldots, P^w\}$, corresponding to privacy values $\{< k^a, \phi^a >, < k^b, \phi^b >, \ldots, < k^w, \phi^w >\}$. For instance, a user u_i can set his privacy profile to one of three privacy levels $\{LOW, MED, HIGH\}$.

We assume that $\mathcal{P}$ is an ordered set. For two privacy levels P^a and P^b with $a < b$, it holds that $k^a < k^b$, and $\phi^a \leq \phi^b$. We further assume that a user u_i with a privacy level preference of P_i^a is willing to accept a privacy level $P_j^b > P_i^a$ of a user u_j. This decision can be automated via a pre-selection option in the user's device. This flexibility allows the participation of users with different privacy settings to the same P2P group.

3.3 Model Assumptions

Network Model–We consider a set of users which obtain location-based services from one or several LBSs. Users are assumed to form an overlay P2P network for obtaining anonymous location-based services. This network can be facilitated by an existing infrastructure such as a set of base-stations, or can occur in an ad-hoc mode. In the latter case, the ad hoc network is responsible for relaying queries to the LBS in a multi-hop fashion. Figure 3.1 shows the two scenarios under consideration; the infrastructure-based P2P overlay and the ad hoc P2P network. The confidentiality and authenticity of P2P and peer-to-LBS communications is guaranteed via cryptographic methods such as symmetric or asymmetric cryptography. Using such methods, any pair of nodes in the network can establish pairwise symmetric keys, when necessary. Users are assumed to be capable of establishing pairwise symmetric keys for the purpose of preserving the confidentiality of pairwise P2P communications. Thus can be facilitated via the use of pre-existing public keys or via other methods such as a Diffie-Hellman key exchange [45]. Moreover, every user u_i holds a secret/public key pair denoted as (sk_i, pk_i).

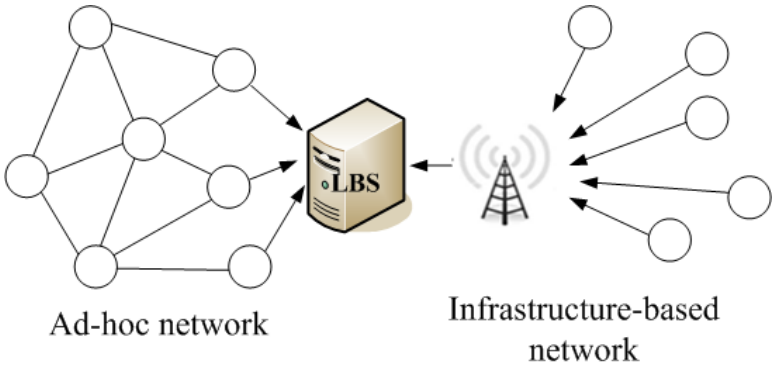

Figure 3.1: System architecture.

Adversary Model–We assume that none of the network participants is trusted. This includes users collaborating for the provision of anonymity as well as the LBS

units. Similar to [47], we further assume that the network participants including the LBS are *honest but curious*. The goal of the participants are to breach the location privacy of users submitting queries. This can be achieved by associating query q_i issued by a user u_i with the identity u_i of that user. Untrusted peers aim at achieving one of the following goals: (a) learn the contents of a query q_i, (b) associate q_i with the user's location ℓ_i, (c) associate a query q_i or a response r_i with a location ℓ_i, and (d) learn the contents of a response r_i. Despite their curious nature, all peers honestly participate in the system and do not launch any active denial-of-service attacks such as dropping, modifying, or misrouting packets.

We consider two possible adversary models. In the first model, each malicious entity is acting independently. Peers and/or LBSs independently attempt to breach the privacy of users based on the information they can collect. In the second model, the LBS may collude with untrusted users in order to breach the privacy of any other user. The two entities share their information in an effort to discover the identity of a user submitting queries, or correlate his identity with his location.

CHAPTER 4

The MAZE Protocol

4.1 System Architecture

In this section, we describe MAZE, a distributed privacy preserving protocol that relies on P2P collaboration. MAZE does not utilize a centralized trusted party such as a location anonymizer. Our goal is to provide the anonymization service in a distributed manner, and in complete lack of trust between the system's participants. As we show in Section 4.5, MAZE satisfies all the system requirements set forth in Section 3.1, under independent adversaries.

In MAZE, a set of users cooperate in order to anonymize their location-based queries to a desired degree of anonymity according to their user profile settings. Once queries have been anonymized, they are collectively submitted to the LBS for service. The LBS is able to authenticate that the queries originate from legitimate users (subscribers) and is able to charge the users submitting the requests. However, the LBS cannot link the submitted queries to individual users. The responses of the LBS propagate back to the group of users and are distributed accordingly. The MAZE protocol consists of the following three phases.

Group formation phase: During this phase, a set of users $\mathcal{U} = \{u_1, u_2, u_3, \cdots, u_k\}$ form a P2P anonymity group of size k. The group $\mathcal{U}$ satisfies privacy profile of each individual user $u_i \in \mathcal{U}$. Because peers are considered to be untrusted, during the group formation phase, users do not reveal their locations for forming $\mathcal{U}$.

Query anonymization phase: In this phase, users in $\mathcal{U}$ anonymize a set of queries $\mathcal{Q} = \{q_1, q_2, \cdots, q_k\}$ by applying a transformation $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$ to the query set $\mathcal{Q}$. The transformed set $\mathcal{F}(\mathcal{Q}) = \{f_1(q_1), f_2(q_2), \ldots, f_2(q_k)\}$, is shuffled between

the group participants (similar to a *mixnet* operation) so that the identity of the originator of each query cannot be determined by the users or the LBS. The goal of $\mathcal{F}$ is dual; to prevent the disclosure of the contents of a query q_i made by u_i to any user in $\mathcal{U}$, and to prevent the LBS from associating q_i with u_i.

Query service phase: In this phase, the set of transformed queries $\mathcal{F}(\mathcal{Q})$ is collectively submitted to the LBS by the group $\mathcal{U}$. The LBS authenticates all users in $\mathcal{U}$, and charges them an appropriate fee for the service it provides. It then obtains $\mathcal{Q}$ by applying the inverse transformation $\mathcal{F}^{-1} = \{f_1^{-1}, f_2^{-1}, \ldots, f_k^{-1}\}$ and prepares a response set $\mathcal{R} = \{r_1, r_2, \cdots, r_k\}$. Set $\mathcal{R}$ is hidden by the application of a transformation $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$ on $\mathcal{R}$. Because the LBS cannot associate q_i with u_i, the set of transformed responses $\mathcal{G}(\mathcal{R})$ is sent to all members of $\mathcal{U}$. Each member is able to extract r_i from $\mathcal{G}(\mathcal{R})$ by applying an inverse transformation g_i^{-1}, but cannot learn the response to any other query.

Figure 4.1 shows the MAZE protocol architecture. A group of five users has formed an anonymity group $\mathcal{U}$. The anonymity group satisfies the privacy requirements in terms of k_i, ϕ_i for all $u_i \in \mathcal{U}$. Members of $\mathcal{U}$ send a transformation $\mathcal{F}(\mathcal{Q})$ of their query set $\mathcal{Q}$ to the LBS. The LBS responds with the transformed response set $\mathcal{G}(\mathcal{R})$. Each member is able to extract the individual response r_i. We now describe the three phases of MAZE in detail.

4.2 Phase I: Group Formation Phase

The group formation phase is initiated by any user who wants to submit a query to the LBS, and does not already belong to an anonymization group. We denote the group initiator as $\hat{u}_i$. Based on his privacy profile, $\hat{u}_i$ selects the values of the anonymity level k, and the k-ASR area radius α. Parameter α can be roughly calculated by $\alpha \approx \sqrt{k/\pi \times d}$, where d is the user density assuming that $\alpha \geq \phi$. Otherwise α can be set equal to ϕ. The group initiator also selects a random point ℓ_c within a disc $C(\ell_i, \alpha)$, where ℓ_i denotes the center of the disc (and also the

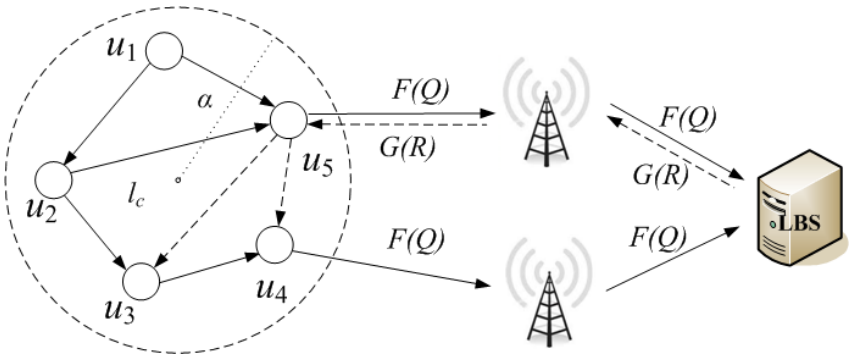

Figure 4.1: The MAZE protocol architecture. A set of users form a P2P group $\mathcal{U} = \{u_1, u_2, u_3, u_4, u_5\}$ with an anonymity degree k=5 and a spatial cloaking region of size A=(l_c, α). The users in $\mathcal{U}$ exchange a transformation $\mathcal{F}(\mathcal{Q})$ of their query set in order to anonymize it. $\mathcal{F}(\mathcal{Q})$ is submitted to the LBS who responds with a set of hidden responses $\mathcal{G}(\mathcal{R})$. $\mathcal{G}(\mathcal{R})$ is flooded among the P2P users with each $u_i \in \mathcal{U}$ obtaining only the response r_i.

location of $\hat{u}_i$) and α denotes its radius. The point ℓ_c serves as the center of the k-ASR and is randomly selected to prevent the *center-of-ASR* attack [25]. Disk $C(\ell_c, \alpha)$ becomes the k-ASR for group $\mathcal{U}$ initiated by $\hat{u}_i$. To form a group $\mathcal{U}$, the following steps are executed.

Step 1: The group initiator broadcasts a group formation message.

$$m_f : g_{id} \parallel (\ell_c, \alpha) \parallel \hat{u}_i \parallel t_s \parallel P_i.$$

Message m_f contains the group id g_{id}, the k-ASR boundaries, the identity $\hat{u}_i$ of the group initiator, a timestamp and user's privacy profile P_i.

Step 2: A user u_j located within the k-ASR boundaries replies with a join message m_j

$$m_j : g_{id} \parallel join \parallel u_j$$

if $P_i > P_j$ (i.e. $k_i \geq k_j$, $\alpha_i \geq \phi_j$). And then u_j rebroadcast m_j out.

Step 3: If the number of join messages received by $\hat{u}_i$ is less than $(k-1)$, u_i increases the size α of the k-ASR and repeats Steps 1, 2. To avoid duplicate join requests, users within the k-ASR do not respond to requests with the same g_{id}.

Step 4: The group initiator $\hat{u}_i$ randomly selects $(k-1)$ users that replied with a join message and forms group $\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$ identified by g_{id}. It then notifies all group members of $\mathcal{U}$ by broadcasting an accept message,

$$m_a : \{u_1, u_2, \cdots, u_k\} \parallel g_{id} \parallel \hat{u}_i \parallel accept$$

The dissemination mechanism for the messages necessary for the formation of $\mathcal{U}$ depends on the underlying network architecture. We describe two mechanisms, one for infrastructure-based networks and one for ad hoc networks.

Infrastructure-based networks–In infrastructure-based networks, communication among peers is realized via base stations (BS). The group initiator $\hat{u}_i$ sends the group formation message m_f to the BS that he is associated with. The BS is responsible for relaying m_f to all users within the area specified by the k-ASR (disk of radius α centered at ℓ_c). Note that depending on the BS deployment, multiple BSs may be needed to participate in the relay of m_f and also in the subsequent phases of the query anonymization. An example of the group formation phase for an infrastructure-based network is shown in Figure 4.2. The group initiator u_1 broadcasts message m_f with a cloaking region k-ASR $C(l_c, \alpha)$ via the BS. Only users with privacy profile $P_i < P_1$ and located within $C(l_c, \alpha)$ will reply with a join message m_j. Group initiator u_1 will then respond with message m_a specifying $\mathcal{U}$, thus completing the group formation phase.

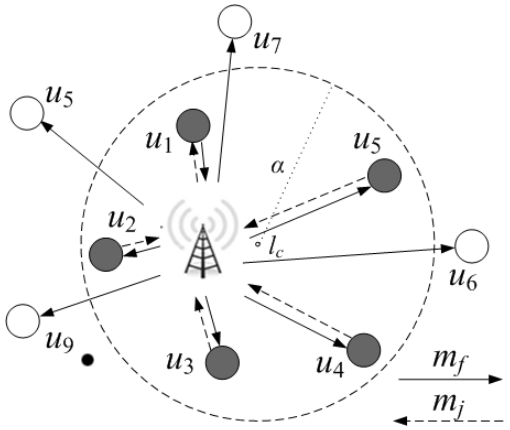

Figure 4.2: Group formation phase in an infrastructure-based network.

Ad hoc networks–In ad hoc networks, the group formation phase takes place in a distributed manner. Here, we employ a flooding mechanism for disseminating the group formation message m_j. In Step 1, the group initiator $\hat{u}_i$ broadcasts m_f. Any user within the k-ASR that receives m_f for the first time re-broadcasts it. Note that users can identify duplicate requests based on the unique g_{id} and timestamp t_s, and therefore, avoid broadcasting m_f multiple times. Nodes within k-ASR receiving m_f continue to relay it. The propagation of m_f terminates at the boundaries of the k-ASR since nodes outside that region ignore m_f. Any user u_j that has received an m_f that satisfies his privacy profile as it is outlined in Step 2, unicasts message m_j to $\hat{u}_i$. The group initiator randomly selects $(k-1)$ replies and forms group $\mathcal{U}$ by sending the accept message m_a to the group members. At this stage, group members form an overlay network using a broadcast routing algorithm for ad hoc networks (e.g. [32]). The resulting broadcast routing tree is utilized for the dissemination of group messages among the group members for the purpose of query anonymization.

In the case where Step 3 is necessary due to an insufficient number of users within the the initial selection of $C(l_c, \alpha)$, the group initiator does not sent an accept message m_a. Instead, he broadcasts a new m_f with the same g_{id}, but different timestamp t_s. Users within the new k-ASR, flood m_f similarly to the flooding of

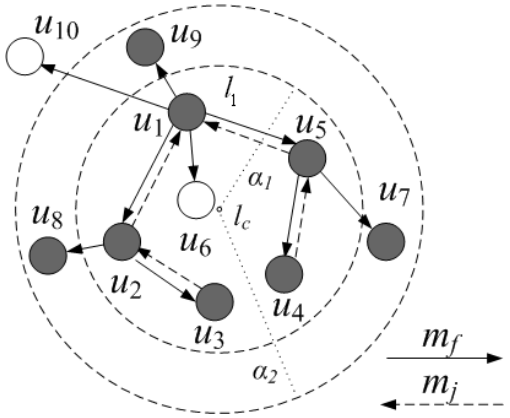

Figure 4.3: Group formation phase in an ad hoc network.

the original request. However, only users that did not reply in the original request, unicast m_j to $\hat{u}_i$. Once $\hat{u}_i$ obtains at least $(k-1)$ replies, it can then broadcast m_a to the corresponding members.

The group formation phase for an ad hoc network of 10 users is shown in Figure 4.3. In this example, assume that u_1 is the group initiator. User u_1 broadcasts a group formation message m_f, indicating the k-ASR boundaries $C(l_c, \alpha_1)$ and P_1. Users u_2 and u_5 rebroadcast m_f since they are located within the k-ASR. Similarly, users u_3 and u_4 rebroadcast the requests received from u_2 and u_5 respectively. On the other hand, users u_7, u_8, u_9 and u_{10} do not propagate m_f since they are outside the k-ASR.

Out of the five users within the k-ASR that received m_f, only four reply with a join message. User m_6 does not join the group due to its high privacy requirement. Since at leat six nodes are needed to satisfy the anonymity value k, user u_1 expands the k-ASR to $C(l_c, \alpha_2)$ and rebroadcasts m_f. Under the new k-ASR, users u_7, u_8, and u_9 reply with a join message. User u_1 randomly picks five users out of the seven that replied with a join request. The group formation phase terminates in a P2P anonymity group of size k, satisfying the privacy settings of all group members.

The group formation phase culminates in a set of k users satisfying their privacy

settings profile and forming a P2P group. Note that users do not reveal their location while joining a group. The only information that is provided is that every user is within the k-ASR. Because a user u_j responds to the group formation request originating from $\hat{u}_i$ only if $\alpha_i \geq \phi_j$ the location privacy of u_j is preserved.

4.3 Phase II: Query Anonymization Phase

In this phase, the P2P group applies a transformation $\mathcal{F}$ to their set of queries $\mathcal{Q}$ in order to anonymize them. The transformation involves cryptographic operations on $\mathcal{Q}$ as well as exchange of messages between members of $\mathcal{U}$.

To anonymize the set of group queries $\mathcal{Q}$ and at the same time allow the LBS to authenticate and charge each group member, we employ All-Or-Nothing (AONT) transformations. Such transformations were originally proposed by Rivest to slow down brute force attacks against block encryption algorithms [41].

Definition 3. AONT*: A transformation f: $\{\mathbb{F}_u\}^n \longrightarrow \{\mathbb{F}_u\}^{n'}$, mapping a message $y = \{y_1, y_2, \ldots, y_n\}$ to a set of pseudo-messages $s = \{s_1, s_2, \ldots, s_{n'}\}$ is set to be an AONT if:*

1. *f is a bijection.*
2. *It is infeasible to obtain any part of y, if one of the s_i is unknown.*
3. *f and its inverse f^{-1} are efficiently computable.*

In this definition, $\mathbb{F}_u$ denotes the alphabet of message blocks y_i, s_i and n' denotes the number of output pseudo-messages given the input messages n, with $n' \geq n$.

Here, the main idea of an AONT is to prevent the recovery of any part of y, if any pseudo-message s_i is not known. In essence, AONTs can be considered to be an (n',n')-threshold scheme, where a secret is split to n' shares all of which are required to recover the secret [41]. Several AONTs have been proposed in the literature

including the package transform [41] and the linear AONT [44]. The latter provides unconditional security[1] while preverving the size of the original message.

To anonymize the set of queries $\mathcal{Q}$, the P2P group executes the following steps.

Step 1: Each user generates a random symmetric key K_{r_i}. This key will be used by the LBS for the encryption of the response r_i corresponding to query q_i.

Step 2: Each user transforms $y_i : q_i \| K_{r_i} = \{y_i^1, y_i^2, \ldots, y_i^{k'}\}$ to pseudo-messages $s_i = \{s_i^1, s_i^2, \ldots, s_i^k\}$ by applying AONT f: $\{\mathbb{F}_u\}^{k'} \longrightarrow \{\mathbb{F}_u\}^k$. Here, $\{\mathbb{F}_u\}$ is the alphabet of the input blocks y_i^j and $k' \leq k$ denotes the number of input blocks needes such that the number of output pseudo-messages is equal to the anonymity degree k.

Step 3: Each user selects a random permutation $\pi_i(\mathcal{U})$ of the user set $\mathcal{U}$ and sends pseudo-message $s_i^j \| s_{ID}^i$ to user $\pi(\mathcal{U})(j)$. Here, s_{ID}^i denotes a unique identifier for s_i, so that pseudo-messages belonging to the same message can be correlated at the LBS. Messages $s_i^j \| s_{ID}^i$ are encrypted with the symmetric key shared between user u_i and user $\pi_i(\mathcal{U})(j)$.

Step 4: Each user u_i encrpyts all received pseudo-messages with the pairwise key shared between u_i and the LBS. It then sends to LBS

$$E_{K_{u_i},LBS}(s_1^{\pi_1^{-1}(\mathcal{U})(i)} \| s_{ID}^1, s_2^{\pi_2^{-1}(\mathcal{U})(i)} \| s_{ID}^2, \ldots, s_k^{\pi_k^{-1}(\mathcal{U})(i)} \| s_{ID}^k) \parallel u_i \parallel g_{id} \parallel (l_c, \alpha). \tag{4.1}$$

The four steps of the anonymization phase are shown in the example of Figure 4.4. In this example, the P2P group consists of three users. Queries $q_i \| K_{r_i}$ are split to three pseudo-messages. The recipient of each pseudo-message is selected

[1] Given any $n' - 1$ pseudo-messages, all messages y are equally likely.

according to the random permutation $\pi_i(\mathcal{U})$ generated by each user. Users forward all received pseudo-messages to the LBS including their identity.

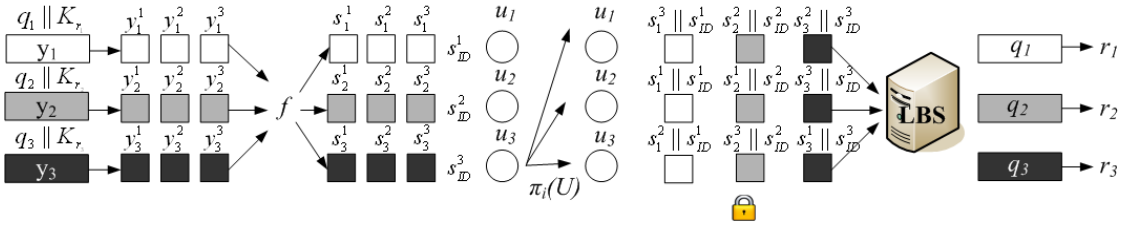

Figure 4.4: Anonymization under independent adversaries.

4.4 Phase III: Query Service Phase

In this phase, the LBS responds to the queries received from the P2P group and charges each of the group participants. To achieve this, the following steps are executed:

Step 1: The LBS authenticates and decrypts every message carrying group id g_{id}.

Step 2: The LBS recovers all transformed queries s_i, $\forall u_i \in \mathcal{U}$, using the message ids s^i_{ID} of each pseudo-message. It then reconstructs each query q_i, $\forall u_i \in \mathcal{U}$ by applying the inverse AONT f^{-1} on $f(\mathcal{Q})$.

Step 3: The LBS prepares response set $\mathcal{R} = \{r_1, r_2, \ldots, r_k\}$ where each r_i corresponds to a query q_i.

Step 4: The LBS sends $g(\mathcal{R}) = \{E_{K_{r_1}}(r_1), E_{K_{r_2}}(r_2), \ldots, E_{K_{r_k}}(r_k)\}$ to all users of the P2P group $\mathcal{U}$.

Step 5: Each $u_i \in \mathcal{U}$ obtains its own response r_i by decrypting $E_{K_{r_i}}(r_i)$ with K_{r_i}.

At the termination of this phase, each user in $\mathcal{U}$ has received its individual response r_i. Moreover, the LBS has authenticated and individually charged every user participating in $\mathcal{U}$.

4.5 Privacy Analysis

In this section, we analyze the privacy properties of MAZE.

4.5.1 Correctness

We first verify that execution of MAZE leads to the service of each query in $\mathcal{Q}$. This can shown in a straightforward manner by inspecting the query anonymization and query service phases. If every participant follows the stpes of MAZE, the LBS will obtain all pseudo-messages corresponding to the queries in $\mathcal{Q}$. Because f is a bijective function set $\mathcal{Q}$ is uniquely recovered. Each user will then be able to obtain this response from the response set $g(\mathcal{R})$ by using his randomly generated symmetric key.

4.5.2 Location Privacy in $\mathcal{U}$

We first show that MAZE preserves the location privacy requirements of any users participating in the P2P anonymization process.

Proposition 1. *For every user $u_i \in \mathcal{U}$, with profile $P_i = \langle k_i, \phi_i \rangle$, no user $u_j \in \mathcal{U}$, $j \neq i$ can determine u_i's position l_i, at an accuracy smaller than ϕ_i.*

Proof. During the group formation phase, the group initiator $\hat{u}_i$ selects k-ASR $C(l_c, \alpha)$ by randomly selecting l_c within $C(l_i, \alpha)$. Therefore, all points within $C(l_c, \alpha)$

area equally likely as candidate position of $\hat{u_i}$. The group formation message m_f contains only $C(l_c, \alpha)$ thus localizing $\hat{u_i}$ within an accuracy $\alpha \geq \phi_i$. Any users u_j replying to m_f acknowledges to be within $C(l_c, \alpha)$, thus allowing its localization with an accuracy α. Since it is assumed that u_j will only reply if $P_i > P_j$, it follows that $\alpha \geq \phi_j$. Therefore, the location privacy of every user in $\mathcal{U}$ with respect to other users in $\mathcal{U}$ is maintained. □

4.5.3 Location Privacy at the LBS

The reconstructed query set at the LBS contains the specification of the k-ASR via the inclusion of $C(l_c, \alpha)$. Because a user u_j participates in $\mathcal{U}$ only if $\alpha \geq \phi_j$, if follows that the LBS cannot learn the location of a user in $\mathcal{U}$ at an accuracy greater than the one specified in everyone's user profiles. Hence, the location privacy requirements of all users in $\mathcal{U}$ are preserved.

4.5.4 Query Privacy in $\mathcal{U}$

We now analyze the query privacy properties of MAZE with respect to the users in $\mathcal{U}$.

Proposition 2. *Collusion of up to* $(k-1)$ *users in* $\mathcal{U}$ *does not reveal the query of the* k^{th} *user.*

Proof. This is a direct consequence of the use of AONTs. During the anonymization phase, a query q_i is partitioned in k pseudo-messages, $(k-1)$ of which are distributed to $(k-1)$ other users. Hence, the collusion of $(k-1)$ users can recover $(k-1)$ pseudo-messages which according to the definition of an AONT do not reveal any information about q_i.

Similarly, because every user u_i with holds one pseudo-message from the group $\mathcal{U}$, the collusion of $(k-1)$ users cannot recover K_{r_i}. Therefore, no user but u_i can decrypt the query response and obtain r_i. Note here that MAZE not only preserves

k-anonymity, but prevents users in $\mathcal{U}$ from learning the contents of queries and responses. □

4.5.5 Query Anonymization at the LBS

We now show that under MAZE, the LBS cannot associate the identity of a user u_i with his query q_i.

Proposition 3. *MAZE preserves k-anonymity with respect to the LBS.*

Proof. After the anonymization phase, every user in $\mathcal{U}$ sends a message y_i to the LBS. This message contains one pseudo-message from every query in $\mathcal{Q}$ and the associated query ids s^i_{ID} corresponding to each query q_i. Using s^i_{ID}, the LBS can reconstruct $\mathcal{Q}$, but it cannot link a particular q_i to a u_i, since the s_{ID} of each query is not related to the user identity. The probability that a query q_i reconstructed for pseudo-messages $s_i = \{s^1_i, s^2_i, \ldots, s^k_i\}$, where each of s^j_i is provided by one user in $\mathcal{U}$ is equal to $1/k$. Hence, all users in $\mathcal{U}$ are equally likely to have generated q_i, and the k-anonymity of users in $\mathcal{U}$ is preserved. □

However, MAZE does not preserve the query anonymity when the LBS colludes with one of the nodes in $\mathcal{U}$. Using the query id s^i_{ID} the LBS can reconstruct every query q_i, $\forall u_i \in \mathcal{U}$. Moreover, individual users in $\mathcal{U}$, can associated any query id s^i_{ID} with the corresponding user u_i. This is because during the steps of the anonymization phase, a user u_i reveals the s_{ID} of his query to the user that receives his pseudo-message. A user u_j receiving a message $s^j_i \| s^i_{ID}$ from u_i, knows that u_i is the message originator and hence can link a message s^i_{ID} to user u_i. Therefore a user colluding with the LBS can associate q_i with u_i via the s^i_{ID}. In the following section we modify the query anonymization phase to prevent the breach of privacy due to the collusion of LBS with users in $\mathcal{U}$.

CHAPTER 5

Collusion-Resistant Query Anonymization

As shown in Chapter 4, MAZE is not resistant to the collusion of the LBS with users in $\mathcal{U}$.In this chapter, we develop L-MAZE, an anonymization protocol that is resistant to the collusion of up to $(L-1)$ users with the LBS.

To preserve anonymity, we employ and L-stage decryption mixnet[7] that anonymizes the originator of pseudo-messages with a given message id. The dis-association of the user u_i from the message s^i_{ID} effectively addresses the collusion problem, in which the LBS exploits its knowledge of the link between a query and a message id to identify a user. In L-MAZE, the group formation phase and query service phase remain identical to those of MAZE. The query anonymizaiton phase is modified as follows.

5.1 L-MAZE: Query Anonymization Phase

The steps of the collusion-resistant query anonymization phase of L-MAZE are as follows:

Step 1: Each user generates a random symmetric key K_{r_i}. This key will be used for the encryption of the response r_i corresponding to query q_i, at the LBS.

Step 2: Each user transforms $y_i : q_i \| K_{r_i} = \{y_i^1, y_i^2, \ldots, y_i^{k'}\}$ to pseudo-messages $s_i = \{s_i^1, s_i^2, \ldots, s_i^k\}$ by applying AONT f: $\{\mathbb{F}_u\}^{k'} \longrightarrow \{\mathbb{F}_u\}^k$.

Step 3: Each user u_i generates a mixnet matrix $M^i_{k\times L}$ with the following properties. Every column of $M^i_{k\times L}$ is a permutation of $\mathcal{U}$. Every row of $M^i_{k\times L}$ is a sub-permutation of $\mathcal{U}$. The initial column permutation is selected at random. The remaining column permutations satisfy the row sub-permutation requirement. One such $M^i_{k\times L}$ for $k = 5$ and $L = 3$ is shown in Figure 5.1.

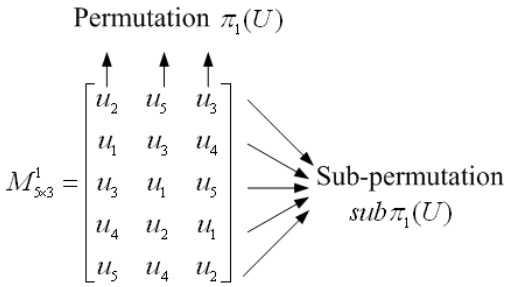

Figure 5.1: MIXNET matrix $M^i_{5\times3}$.

Step 4: Every user u_i with pseudo-messages $s_i = \{s^1_i, s^2_i, \ldots, s^k_i\}$ encrypts each pseudo-message $s^j_i \| s^i_{ID}$ L times using the sequence of public keys of the set of users denoted by row j of $M^i_{k\times L}$.

$$E_{pk_{M^i_{k\times L}(j,1)}}(E_{pk_{M^i_{k\times L}(j,2)}}(\ldots(E_{pk_{M^i_{k\times L}(j,L)}}(s^j_i \| s^i_{ID})))). \tag{5.1}$$

Mixnets are cryptographic systems that implement an anonymous channel between two parties, a sender and a receiver[8]. These systems involve a combination of cryptographic operations such as an encryption/decryption operation with a shuffling and/or permutation operation in order to prevent the tracing of a message to its origin. The cryptographic and shuffling/permutation operations may be repeated a number of times L, until messages are sufficiently anonymized. In this case, we say that the mixnet consists of L stages.

In our context, the role of the sender is assumed by a user u_i that wants to anonymize query q_i. The role of the receiver is assumed by the set of colluding

users and the LBS.

Step 5: Every encrypted pseudo-message is broadcasted to the entire group $\mathcal{U}$.

Step 6: At each stage, an intended receiver decrypts one layer of encryption from each pseudo-message it receives. User waits until it receives $(k-1)$ pseudo-messages. It then repeats steps 5, 6 a total of $(L-1)$ times (L stages in total).

Step 7: After all layers of encryption have been removed, each user encrypts all received pseudo-messages and his id with the symmetric key shared between the user and the LBS. It then sends all encrypted pseudo-messages to the LBS.

5.2 Privacy Analysis

5.2.1 Correctness

We first show that the execution of the L-MAZE protocol allows the correct reconstruction of all queries at the LBS. This is shown via the following proposition.

Proposition 4. *At every stage of the mixing operation, every user in* $\mathcal{U}$ *holds exactly one pseudo-message from each query.*

Proof. Consider a message $y_i = q_i \| K_{r_i}$ transformed in k pseudo-messages $s_i = \{s_i^1, s_i^2, \ldots, s_i^k\}$ via an AONT. At every mixnet stage these messages are sent to a permutation of $\mathcal{U}$ (columns of $M_{k\times L}^i$ are permutations of $\mathcal{U}$). Hence, at the each stage, every user in $\mathcal{U}$ receives exactly one s_i^j. □

There are a total of k queries, each one being split to k pseudo-messages. According to Proposition 4, at the end of the L^{th} stage, each user receives one pseudo-message from each query. In all, each user holds k pseudo-messages, each one

belonging to one of the k queries. In Step 7, all pseudo-messages are forwarded to the LBS. The LBS reconstructs the s_is, $\forall i$ using each s^i_{ID} and recovers the query q_i, $\forall i$ by applying the inverse AONT.

5.2.2 Query anonymization under collusion

We now analyze the resistance of L-MAZE to the collusion of the LBS with up to $(L-1)$ users. Our purpose is to show that collusion of any number $x \leq (L-1)$ users in $\mathcal{U}$ with the LBS reduces the anonymity of the remaining $(k-x)$ users to $(k-x)$. That is, the queries submitted by the $(k-x)$ non-colluding users are indistinguishable. This is the best case scenario since the colluding users already reveal their queries to the LBS.

To illustrate the collusion resistance property, we model the mixnet represented by a matrix $M^i_{k\times L}$ as a set of paths on a complete graph G_k where the set of vertices corresponds to the users of $\mathcal{U}$. An example of such a graph for the mixnet of Figure 5.1 is shown in Figure 5.2.

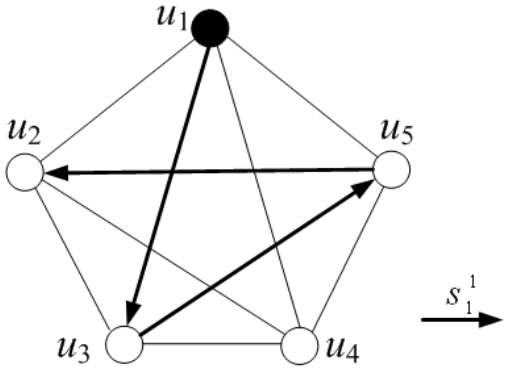

Figure 5.2: G_5 for the user u_1.

Using this graph model, the mixnet operations applied to a pseudo-message s^j_i are represented as a path $P^j_i = \{i, M^i_{k\times L}(j,1), M^i_{k\times L}(j,2), \ldots, M^i_{k\times L}(j,L)\}$. This path has the following properties.

Proposition 5. *Each path $P_i^j, j = 1, \ldots, k$ corresponding to the mixnet operation applied to the pseudo-messages $s_i = \{s_i^1, s_i^2, \ldots, s_i^k\}$ contains at least one non-colluding user.*

Proof. Every path P_i^j consists of exactly L vertices corresponding to the L stages of the mixnet. These vertices are denoted by each of the rows of $M_{k\times L}^i$. By construction, each row of $M_{k\times L}^i$ is a sub-permutation of $\mathcal{U}$ of size L. Therefore, P_i^j contains exactly L distinct vertices. Given that at most $(L-1)$ users are assumed to collude, every path P_i^j will contain at least one non-colluding user. □

Using Proposition 5, we can now show that L-MAZE preserves $(k - L - 1)$ anonymity.

Proposition 6. *L-MAZE preserves the query anonymity of the $(k-x)$ non-colluding users when $x \leq (L-1)$ users collude with the LBS.*

Proof. According to Proposition 4, at each stage of L-MAZE, every user obtains exactly one pseudo-message from each user in $\mathcal{U}$ corresponding to the k queries generated in $\mathcal{U}$. Moreover, according to Proposition 5, the pseudo-messages of a transformed query s_i follow paths that contain at least one non-colluding user. When the encrypted form of pseudo-message s_j^i reaches one non-colluding user, it is mixed with $(k-1)$ other encrypted pseudo-messages. Out of the total k messages received by the non-colluding user, x of them are known to the LBS since they belong to the colluding users. The remaining $(k - x)$ messages were generated by non colluding users. These unknown $(k-x)$ pseudo-messages are encrypted with the public key of the non-colluding user that received these k messages. When this layer of encryption is removed at the non colluding user, the incoming $(k-x)$ encrypted pseudo-messages become uncorrelated to the $(k - x)$ outgoing ones, since the secret key used of the non-colluding user for decryption is not known to others. Note that the message id s_{ID}^i appended to each pseudo-message is randomly generated for each query. Therefore, even if a user submits the exact query twice, the incoming and outgoing

cipher texts are randomized. Since the colluding users cannot correlate the $(k-x)$ incoming pseudo-message with the $(k-x)$ outgoing ones when mixed at a non-colluding node, the linking of an s_{ID}^{i} to u_i becomes a random guess with probability of success equal to $1/(k-x)$.

We emphasize that in Step 5 of L-MAZE, the encrypted pseudo-messages are broadcasted to the entire group $\mathcal{U}$. This is because a recipient of an encrypted pseudo-message is not aware of which user is supposed to receive the pseudo-message at the next stage of the mixnet. Even if the intended recipient is reached before the broadcast is completed (recall that users in $\mathcal{U}$ may span in multiple hops), the relay of the pseudo-message is contained until all users in $\mathcal{U}$ receive it. Therefore, when a pseudo-message leaves a colluding user and is destined to a non-colluding one, the set of colluding users cannot identify which of the non-colluding users are the intended recipients. In combination with the mixing operation of the incoming pseudo-messages, occurring at the non-colluding users that property of $(k-x)$-anonymity is shown. □

To illustrate Proposition 6 consider the mixnet shown in Figure 5.3. Assume that users u_2 and u_5 are colluding and attempt to correlate a pseudo-message originating from u_1 with his identity. At stage 1, u_2 identifies that the encrypted pseudo-message he received originated from u_1, since the first transmission of any user must contain pseudo-messages generated from that user[1].

User u_2 will remove one layer of encryption and forward the pseudo-message to the next stage of the mixnet. Since u_4 is a non-colluding node, users (u_2, u_5) cannot identify which of the (u_2, u_3, u_4) received the pseudo-message. Moreover, u_4 receives 4 more pseudo-messages, only two of which are known to (u_2, u_5). User u_4

[1]Because the encrypted pseudo-messages do not contain any identifier with repect to the recipient, a user has to try decrypting all received pseudo-messages with his private key and keep the k ones that he is able to decrypt. To indicate the success of the decryption, the pseudo-messages are accompanied by a message authentication code.

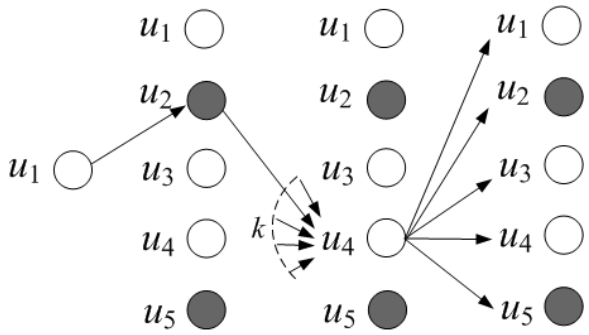

Figure 5.3: In 3-MAZE, when two users u_2 and u_5 collude, they are not able to trace back to the query sender u_1.

will remove one layer of encryption and forward all 5 recieved pseudo-messages to the next hop. Out of the five pseudo-messages forwarded to stage three, the three that belong to u_1, u_3, u_4 are indistinguishable. Therefore, (u_2, u_5) can correlate those messages to their originators with probability 1/3.

CHAPTER 6

Analysis of the Communication Overhead of MAZE

In this chapter, we analytically evaluate the communication overhead of MAZE. We express this overhead in terms of the number of messages that need to be transmitted in order to complete each of the phases of MAZE. We consider two cases, (a) number of messages transmitted by users participating in the P2P group without taking into account the multi-hop nature of the network, and (b) number of messages transmitted by any of nodes of the network. For case (b), we further divide our classification to infrastructure-based and ad hoc networks.

6.1 Communication Overhead of MAZE

We first compute the number of messages that need to be transmitted in MAZE and L-MAZE without taking into consideration how these messages are relayed to the appropriate destinations. Here, we assume that the users of the P2P group form an overlay network and hence, are connected via one-hop virtual connections. The total communication overhead can be expressed as the sum of the communication overheads for completing each of the three phases of MAZE.

$$O_{Total} = O_{gfp} + O_{qap} + O_{qsp}, \tag{6.1}$$

where O_{Total} denotes the total communication overhead and O_{gfp}, O_{qap} and O_{qsp}, denote the overhead of the group formation, query anonymization and query service phases, respectively. For the group formation phase it holds that

$$O_{gfp} = k + 1. \tag{6.2}$$

Eq. (6.2) accounts for the group formation message m_f, the group acceptance message m_a and the individual join messages m_j sent from each user located within the k-ASR. Here, we have assumed that the size of the k-ASR has been selected to contain exactly k users. Therefore, $(k-1)$ users reply to the group initiator with a join request.

In the query anonymization phase, every user transforms his query to a set of k pseudo-messages, $(k-1)$ of which are distributed to other users. After the query anonymization, every user sends one message to the LBS. Hence, the total number of messages transmitted during this phase is equal to

$$O_{qap} = k^2. \tag{6.3}$$

Note that each pseudo-message has a length equal to $\frac{1}{k}^{th}$ of the length of the query message y. Thus, the $(k-1)$ pseudo-messages transmitted for the query anonymization can be thought of to be equivalent to the transmission of one message equal to the query q_i, plus the random symmetric encryption key. In this case, the number of messages of the anonymization phase becomes equal to $O_{qap} = 2k$. In the case of L-MAZE, the anonymization phase has L stages, with each stage involving the transmission of k pseudo-messages, or the equivalent of a single query. Thus, the communication overhead of the anonymization phase becomes equal to $(L+1)k$.

Finally, during the query service phase, the LBS sends one reply message for every query it receives accounting for a total of $O_{qsp} = k$ messages. Summing the communication overhead of all three phases yields

$$O_{Total} = k + 1 + 2k + k = 4k + 1. \tag{6.4}$$

That is, the communication overhead of MAZE increases linearly with the k-anonymity requirement. For L-MAZE, the total communication overhead is equal to $(L+3)k+1$.

6.2 Network Overhead of MAZE

In this section, we evaluate the communication overhead imposed by MAZE on the entire network. In our analysis, we account for the number of messages that need to be relayed by the network in order to complete the different phases of MAZE.

6.2.1 Infrastructure-based Networks

In infrastructure-based networks, communication among the peers of group $\mathcal{U}$ takes place via BSs. In our analysis, we assume all users participating in a P2P group are served by a single BS. Hence, the transmission of each message from any of the group participants, or the LBS requires a total of two messages to be relayed to the appropriate destination; one message from the source to the BS and one message from the BS to the intended destination. Utilizing eq. (6.5), the required network overhead for realizing MAZE is

$$O_{Total} = 2(4k + 1). \tag{6.5}$$

For L-MAZE the network overhead becomes equal to $O_{Total} = 2(L + 3)k + 2$.

6.2.2 Ad hoc Networks

When the communication among the peers of the P2P group is realized via an ad hoc architecture, the network overhead of MAZE is topology dependent. To make our analysis tractable, we make the following simplifying assumptions. We assume that users are uniformly distributed within the network area with a density d[1]. Moreover, every user can reach the LBS via a single transmission.

[1]Our analysis can be easily extended to the case where users with different profiles are deployed with different densities.

Group Formation Phase: In the group formation phase, the group initiator broadcasts a group formation message m_f that must be relayed to every node within the k-ASR. Given the radius α of the k-ASR, the relay of m_f requires on average, the transmission of $d\pi\alpha^2$ messages, which is equal to the expected number of nodes within an area of size $\pi\alpha^2$. Note that all nodes within the k-ASR relay m_f irrespective of whether the privacy profile included in m_f satisfies their own privacy requirements.

In Step 2 of the group formation phase, each user within the k-ASR that meets the privacy requirements set by the group initiator, unicasts a join message m_j to the group initiator. Depending on the network topology, each m_j may be relayed to the group originator via a multi-hop route. To compute the overhead of this phase, we divide the k-ASR to several zones $Z_1, Z_2, \ldots, Z_h$. The number of zones is equal to $h = \lceil \frac{\alpha+\epsilon}{r} \rceil$, where ϵ denotes the distance between the group initiator and the randomly selected center of the k-ASR, and r denotes the communication range of each user.

As shown in Figure 6.1, each zone Z_i is the intersection between a ring of width r and the k-ASR. We make the approximation that a join request originating from a user within zone Z_i requires a total of i transmissions before it is received by the group originator. Let the area of each zone that intersects with the k-ASR be denoted by $A_{Z_i}(\epsilon)$. The total number of messages that are transmitted by the network for the completion of step two of MAZE is equal to

$$\sum_{i=1}^{h(\epsilon)} i d A_{Z_i}(\epsilon). \tag{6.6}$$

To compute the size of the areas $A_{Z_i}(\epsilon)$ for any arbitrary i, we consider the geometry of Figure 6.1. In Figure 6.1, disc $C_r(A(x_A, y_A), r)$ denotes the communication area of the group originator and disk $C_\alpha(B(x_B, y_B), \alpha)$ denotes the k-ASR. Based on the selection strategy of point B (the center of k-ASR), the distance $AB = \epsilon$ may

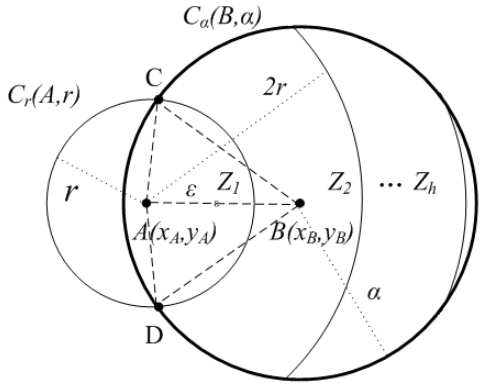

Figure 6.1: Intersection area of disc $C_r(A, r)$ and $C_\alpha(B, \alpha)$.

vary to any value from 0 to α. In fact, it is easy to show via elementary geometric arguments that the probability distribution for the distance AB, given that A is uniformly selected to be within the circle $C_\alpha(B, \alpha)$ is given by

$$\Pr[AB \leq \epsilon] = \frac{1}{\alpha}, \quad 0 \leq \epsilon \leq \alpha \tag{6.7}$$

For a fixed distance ϵ, the area of zone Z_1 is equal to the area of intersection between discs C_r and C_α (we have assumed that $\alpha > r$). Assume $C_r(A, r)$ and $C_\alpha(B, \alpha)$ intersect in two points which is labeled C and D as shown in Figure 6.1. The area of zone Z_1 can be computed to be equal to

$$A_{Z_1}(\epsilon) = \begin{cases} \frac{1}{2}r^2 CBD - \frac{1}{2}r^2 \sin(CBD) & \\ +\frac{1}{2}\alpha^2 CAD - \frac{1}{2}\alpha^2 \sin(CAD) & \text{if } \epsilon \geq \alpha \text{ and } \epsilon \geq r \\ \pi r^2 & \text{if } r \leq \epsilon \leq \alpha \end{cases} \tag{6.8}$$

where

$$\epsilon = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$
$$CBD = 2 \times \arccos\left(\frac{r_r^2 + (\epsilon)^2 - r_\alpha^2}{2 \times r_r \times \epsilon}\right)$$
$$CAD = 2 \times \arccos\left(\frac{r_\alpha^2 + (\epsilon)^2 - r_r^2}{2 \times r_\alpha \times \epsilon}\right)$$

Similarly, the area of zone Z_2 can be computed as the intersection between a disk of radius $C_{2r}(A, 2r)$, and the k-ASR minus the area of zone Z_1.

$$A_{Z_2}(\epsilon) = area(C_{2r}(A, 2r) \cap C_\alpha(B, \alpha)) - A_{Z_1}(\epsilon). \tag{6.9}$$

Generalizing to the area of zone Z_i it follows that,

$$A_{Z_i}(\epsilon) = area(C_{ir}(A, ir) \cap\ C_\alpha(B, \alpha)) - \sum_{j=1}^{i-1} A_{Z_j}(\epsilon),\ i \geq 2. \tag{6.10}$$

By substituting eq. (6.10) to eq. (6.6), we obtain an approximation of the communication overhead associated with Step 2 of MAZE, for a given distance ϵ. Averaging over all possible values of ϵ based on the probability density function in eq. (6.7), we obtain the value of the expected communication overhead of Step 2 as

$$\int_0^\alpha \sum_{i=1}^{h(\epsilon)} i dA_{Z_i}(\epsilon) \frac{1}{\alpha} d\epsilon. \tag{6.11}$$

In Step 3 of the group formation phase, the group initiator sends an accept message m_a to $(k-1)$ nodes located within the k-ASR. To facilitate the communication among the group members, we have assumed that communication takes place via a broadcast routing tree. The overhead associated with this type of communication is topology-dependent and equal to the number of non-leaf nodes of the broadcast tree topology. To estimate this communication overhead, we evaluate the number of users that are needed for covering the k-ASR with a connected network. The justification for this estimated value lies in the selection process of the users that participate in the P2P group. Out of all users within the k-ASR, the k users that compose $\mathcal{U}$ are selected at random. Hence, these users can be located anywhere within the k-ASR. A connected network of users that covers the k-ASR is guaranteed to relay information to all users within it. Therefore, by finding the minimum number of users necessary to cover the k-ASR, we calculate a conservative estimate on the overhead broadcast routing operation.

The number of users needed to cover the disk of radius α with a connected network is given by the following theorems [34, 50].

Theorem 1. *[34]: Consider a two-dimensional infinite plane where nodes of coverage range r_0 are deployed uniformly at density d. The area coverage A_c of such network is*

$$A_c = 1 - e^{-d\pi r_0^2} \tag{6.12}$$

Theorem 1 is a result obtained from the discipline of stochastic geometry [22]. Using Theorem 1, we can compute the required density d_{min} such that the k-ASR is covered almost surely [34]. This can be achieved by setting $A_c = \pi a^2$ and solving for d_{min}. That is,

$$d_{min} = -\frac{\ln(1 - \pi\alpha^2)}{\pi r_0^2}. \tag{6.13}$$

In our scenario, the coverage range of the nodes that cover the k-ASR corresponds to the communication range r of the users. If a set of users with communication range r_0 covers the k-ASR, then their transmission is guaranteed to reach every node within the k-ASR. However, coverage is not a sufficient condition to obtain a connected network of users that would correspond to a broadcast routing tree. To satisfy the connectivity requirement, we rely on the following theorem.

Theorem 2. *[50] Let a set of nodes have a coverage range r_0 and a communication range r. If this set of nodes at least 1-covers a convex region A, the communication graph is connected if $r \geq 2r_0$.*

Combining Theorem 2, with eq. (6.13) yields the desired minimum density such that the network of nodes that 1-covers the k-ASR is also connected. One example of 1-cover connected network that covers the k-ASR is shown in Figure 6.2. This is achieved by setting the coverage range equal to half the communication range, i.e., $r_0 = \frac{r}{2}$

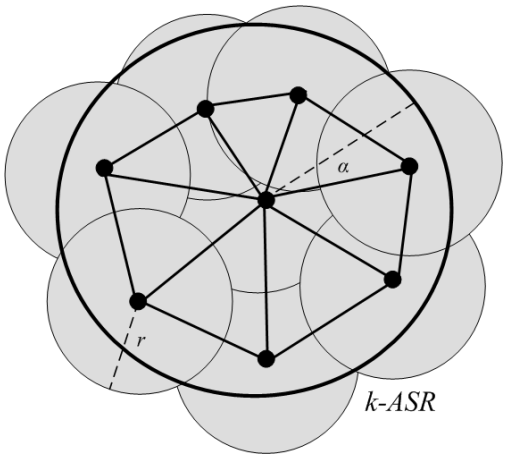

Figure 6.2: Connected-coverage. The shaded area represents the connected-covered area.

$$d_{min} = -\frac{4\ln(1-\pi\alpha^2)}{\pi r^2}. \tag{6.14}$$

The number of nodes that need to transmit in order to cover the k-ASR is found by multiplying the density with the size of the k-ASR. This number is equal to $d_{min}\pi\alpha^2 = -\frac{4\ln(1-\pi\alpha^2)}{\pi r^2}\pi\alpha^2$. We note that the infinite plane assumption of Theorem 1 is satisfied in our case, since the k-ASR is a small part of the entire network deployment and hence, border effects do not apply. The deployment density is kept constant through the entire k-ASR.

Combining all steps of the group formation phase, the average network overhead for this phase is estimated to

$$O_{gfp} = d\pi\alpha^2 + \int_0^{\alpha} \sum_{i=1}^{h(\epsilon)} i dA_{Z_i}(\epsilon)\frac{1}{\alpha}d\epsilon + d_{\min}\pi\alpha^2. \tag{6.15}$$

Query Anonymization Phase: In the query anonymization phase, the members of $\mathcal{U}$ exchange pseudo-messages using the established broadcast routing tree. We

note that according to Step 5 of the query anonymization phase, any pseudo-message is broadcasted to the entire group $\mathcal{U}$, in order to keep the identity of the recipient secret. Hence, the network overhead of the query anonymization phase is equal to the number of messages that need to be transmitted by the users in $\mathcal{U}$ times the number of transmissions necessary to broadcast a message to all nodes, as expressed by eq. (6.14). Here, we consider the transmission of k pseudo-messages to be equivalent to the transmission of one query, since the pseudo-messages have a length equal to $\frac{1}{k}^{th}$ of the query messages. For MAZE, this network overhead is estimated to

$$O_{qap} = kd_{\min}\pi\alpha^2 + k. \tag{6.16}$$

The second component accounts for the k messages that have to be relayed from each of the users in $\mathcal{U}$ to the LBS, after the pseudo-messages have been mixed. In the case of L-MAZE, there are L mixing stages of the k pseudo-messages that correspond to each of the k queries. Hence, the network overhead for this phase increases to

$$O_{qap} \leq Lkd_{\min}\pi\alpha^2 + k. \tag{6.17}$$

CHAPTER 7

Experiment Evaluation

In this chapter, we experimentally evaluate the communication overhead of MAZE via simulations. We compare the overhead of MAZE with the P2P anonymization protocol in [11]. We note however that the two protocols cannot be considered to be equivalent since the protocol in [11] does not preserve location privacy in $\mathcal{U}$, and is not resistant to collusion.

7.1 Experimental Setup

We randomly deployed n users within an area of $1,000m \times 1,000m$, yielding a user density of $n/1,000,000$ $users/m^2$. Users are assumed to form multihop ad hoc network. The communication range of each user was assumed to be $250m$. For the query anonmization phase, a broadcast tree was built using the broadcast routing algorithm in [32]. We note that nodes of the ad hoc network may participate in the broadcast tree without necessarily being a member of $\mathcal{U}$. Each experiment was repeated for 40 random network topologies and the results were averaged.

7.2 Impact of the Anonymity Level k

7.2.1 Varying Cloaking Region Size

In the first set of experiments, we varied the anonymity level requirement of the group initiator from $k = 10$ to $k = 30$, while the cloaking region size was varied according to the anonymity level k. Here, the cloaking region size was set to the minimum value that satisfied k-anonymity. To find that value, we initially set the

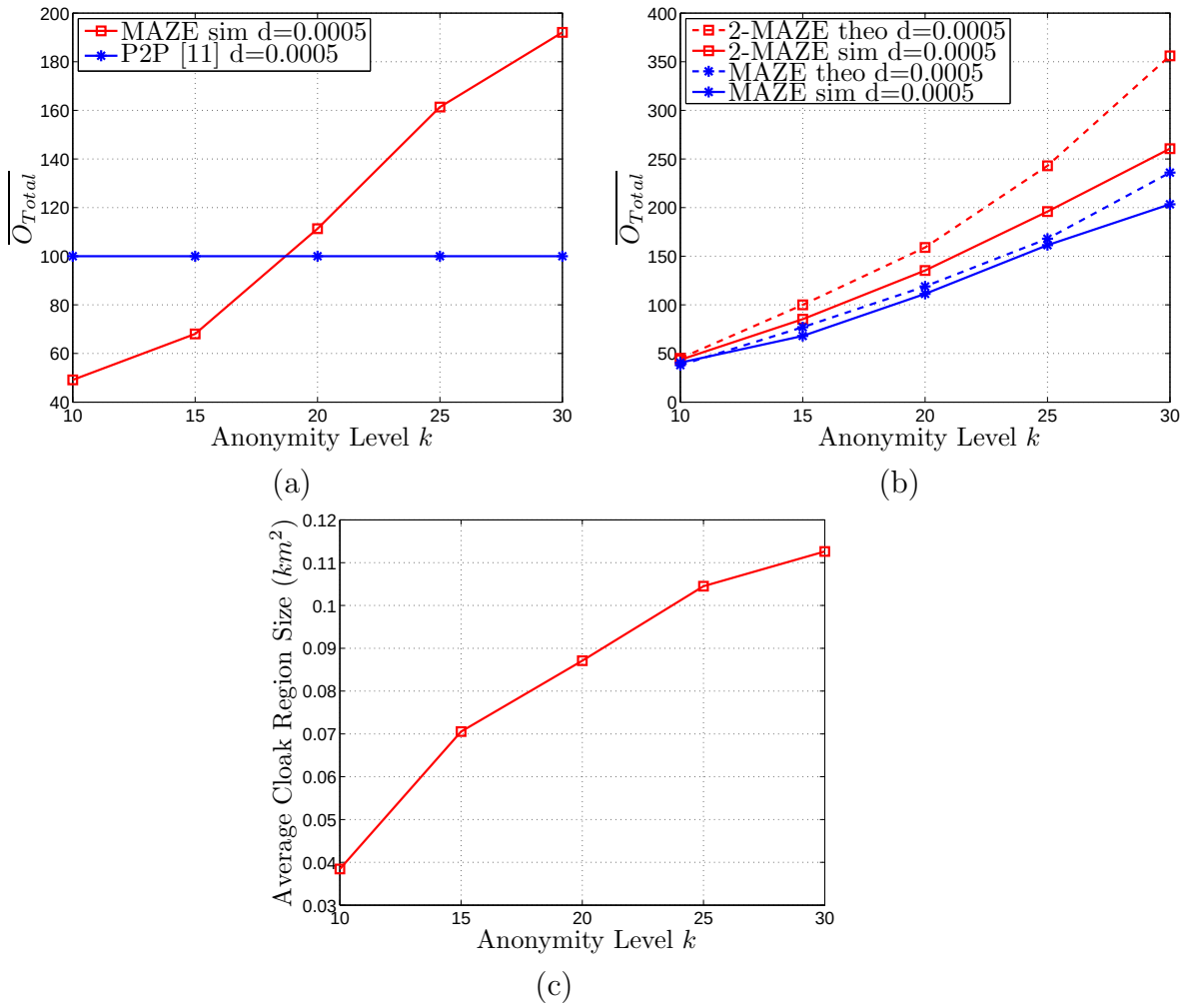

Figure 7.1: (a) Average communication overhead as a function of the anonymity level k, (b) Comparison of the average communication overhead between MAZE and 2-MAZE as a function of the anonymity level k, (c) Average cloaking region size as a function of the anonymity level k.

radius of k-ASR to $\alpha = \sqrt{\frac{k}{\pi d}}$. If k user were not found, α was increased by a factor $\beta = 0.1$. All users were assumed to have the same privacy profile.

Figure 7.1(a) shows the average communication overhead $\overline{O_{Total}}$ when the user density is set to $d = 0.0005$, compared to the communication of the P2P protocol in [11] as a function of k. We observe that for small values of k, MAZE has lower $\overline{O_{Total}}$ than the scheme in [11]. This communication overhead increases linearly with the anonymity level requirement. On the other hand, the protocol in [11] incurred a fixed communication overhead since the search for group peers is limited to broadcasts within one hop as long as the user density is sufficient to satisfy k. Note that nearly all communication overhead of the P2P scheme in [11] is incurred during the group formation phase, since there is no mechanism that protects the privacy of users within $\mathcal{U}$. MAZE, on the other hand, incurs most of its overhead during the query anonymization phase where pseudo-messages are mixed among the users of the group in order to preserve the query anonymity. Therefore, the larger k the value of the larger the number of message exchanged during anonymization.

In Figure 7.1(b) we compared the communication overhead of the version of MAZE that preserves the query privacy when users do not collude with the LBS, with L-MAZE, the version of MAZE that is resistant to LBS-user collusion. Here we set $L = 2$. As expected, the communication overhead of the collusion-resistant MAZE protocol is increased L-fold compared to that of MAZE under independent adversaries. This is due to the fact that L-MAZE involves L rounds of mixing during the query anonymization phase (the two protocols incur the same overhead during the group formation phase). The collusion resistance property comes at the expense of a proportional increase in the number of messages.

In the same graph, we also show the theoretical values of the network overhead of MAZE, as computed based on our analysis in Chapter 6. We observe that our analytical estimates are conservative, and the gap between the theoretical and simulated values increases as a function of k. This can be justified by the conservative

estimate of the number of messages needed for completing a broadcast operation using the broadcast routing tree. Because this broadcast operation is repeated k times during the query anonymization phase, any error in the estimate of this overhead accumulates with k in a linear fashion.

Finally, in Figure 7.1(c) we show the size of the cloaking region as a function of k. Based on the node density d, we expect that the cloaking region will grow according to

$$A_{CR}(k) = \frac{k}{d}. \tag{7.1}$$

Figure 7.1(c) shows the almost linear increase of $A_{CR}(k)$ with the increase of k.

7.2.2 Fixed Cloaking Region

In the second set of experiments, we kept the size of the cloaking region fixed and independent of k. Thus scenario arises when the privacy profile of the user requires a minimum privacy resolution tolerance that is larger than the minimum k-ASR size that satisfies the k-anonymity requirement. Figure 7.2 shows the average communication overhead of MAZE as a function of the anonymity level k, when the radius of the k-ASR is fixed to $\alpha = 160m$ and $\alpha = 180m$ respectively. We observe that the communication overhead of MAZE increases linearly with k and also becomes larger with α. This is justified by the fact that the k users of $\mathcal{U}$ are spread over a large area and thus communication among the group peers becomes more expensive.

Moreover, comparing to the MAZE overhead for the case of a varying k-ASR we identify that for lower values of k, keeping the k-ASR area large leads to a higher overhead. However, as the value of k increases the overhead with varying k-ASR becomes larger. This is because if k is not satisfied with the initial value of α, the peer discovery phase has to be repeated multiple times. Setting the value of α tot a large value almost guarantees the discovery k peers thus limiting the group formation phase to a single iteration.

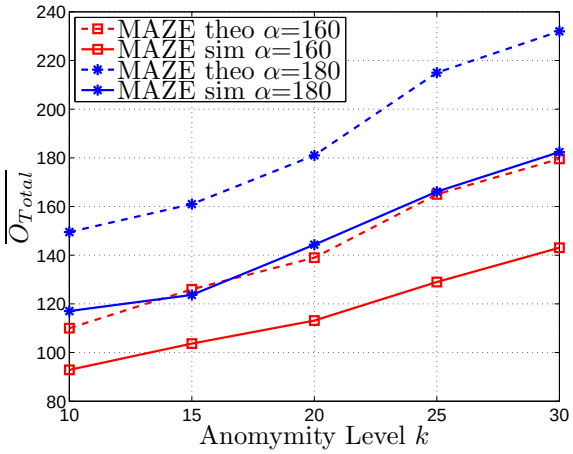

Figure 7.2: Anonymity level impact with fixed cloaking region size.

7.3 Impact of User Density

In this set of experiments, we evaluated the impact of the user density on the communication overhead of MAZE. Figure 7.3(a) shows the average communication overhead $\overline{O_{Total}}$ as a function of d compared to the P2P protocol in [11], for $k = 10$. We observe that for the P2P protocol, the overhead increases linearly according to the user density value. This because the number of users within one hop of the group initiator, who reply to a group formation message are a linear function of d. For MAZE protocol, $\overline{O_{Total}}$ decreases with d approaching an asymptotic value. This behavior is explained by the expected network topology of the ad hoc network as d increases. For large values of d, users of $\mathcal{U}$ are located within a one-hop range of each other (they form a one-hop network). Thus smaller cloaking region are needed to satisfy the k-anonymity requirement, as shown in Figure 7.3(c). Therefore, a transmission by any user is received by all other users. Under such a topology, the overhead costs of the group formation and query anonymization phases are fixed and cannot be further reduced by the increase of d.

The exact overhead cost of MAZE when applied to a one-hop ad hoc network is equal to

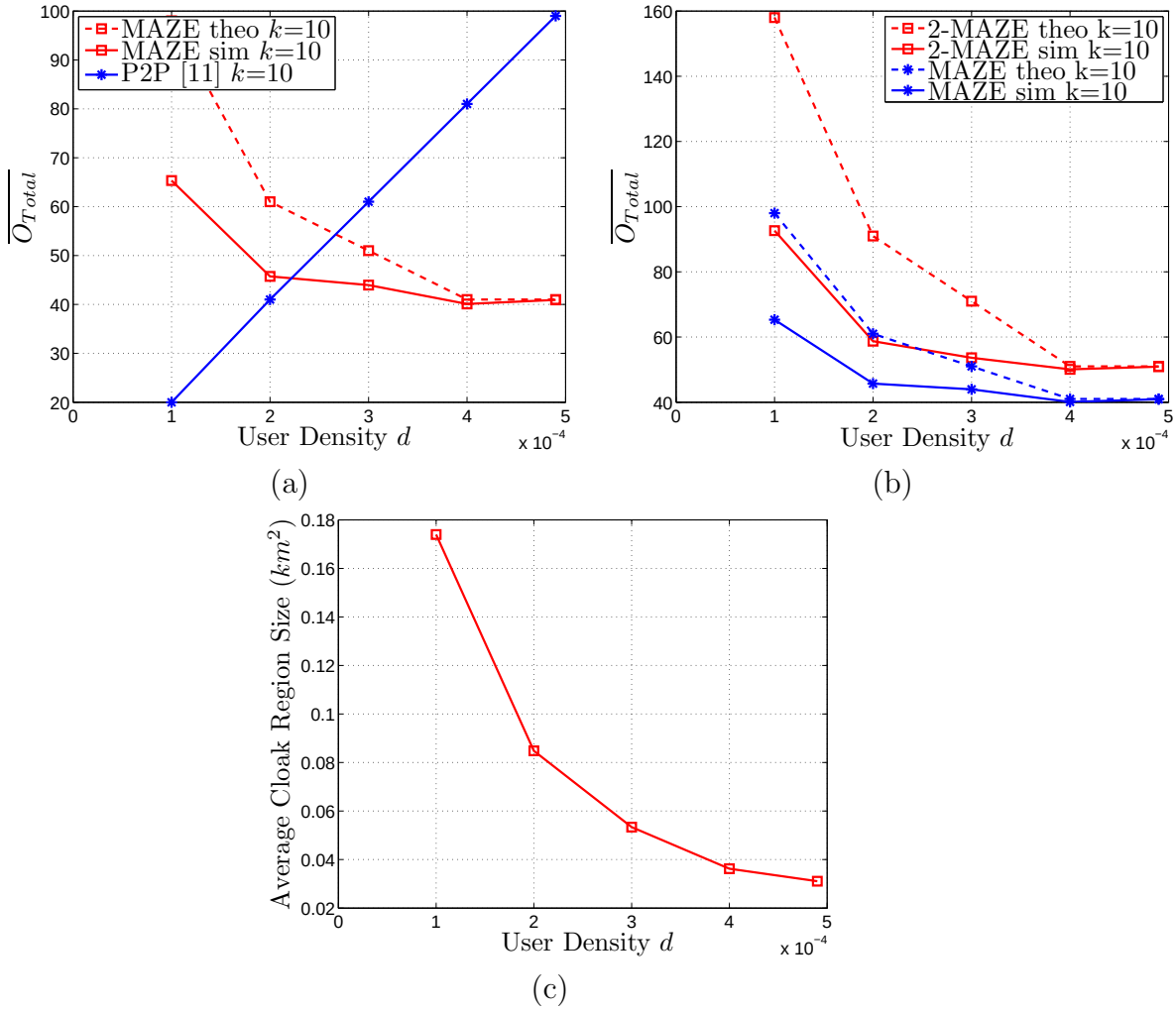

Figure 7.3: (a) Average communication overhead versus user density, (b)Comparison of average communication overhead between MAZE and 2-MAZE versus user density, (c) Average cloaking region size versus user density.

$$O_{total} = 2d\pi\alpha^2 + k - 1, \tag{7.2}$$

where α is the radius of the k-ASR. As expected, L-MAZE increase the communication overhead of the anonymization phase L-fold compare to MAZE, while the overhead of the group formation phase is decrease, as shown in Figure 7.3(b).

7.4 Impact of the k-ASR Expansion Factor β

In this set of experiments, we study the impact of the k-ASR expansion factor β. This parameter denotes the factor by which the k-ASR radius is extended if k users are not discovered within the initial k-ASR. Smaller values of β lead to a more accurate determination of the minimum k-ASR size that satisfies the k requirement at the expense of repeating the group formation process multiple times until k users are discovered. On the other hand, large values of β require fewer stages until k users are discovered but lead to the spreading of those k users over a larger area.

In Figure 7.4(a), we show the size of the cloaking region as a function of β for different densities. We observe that larger β result in larger k-ASR. We note that the k-ASR was increased only if k users were not discovered within our initial k-ASR area. Figure 7.4(b) shows the associated overhead of MAZE for anonymizing k queries as a function of β. We observe that β plays an important role only when the user density is not sufficient to satisfy the anonymity level requirement with the selection of the initial k-ASR size.

7.5 Impact of $\hat{u_i}$'s Privacy Requirement

As described in Section 3.2, assuming $\hat{u_i}$ can set his privacy profile to one of three privacy levels $\{LOW, MED, HIGH\}$, the higher the user's privacy requirement level is, the easier he is able to find other peers to join the group, since they can all accept his privacy requirement. To study the impact of $\hat{u_i}$'s privacy requirement, we

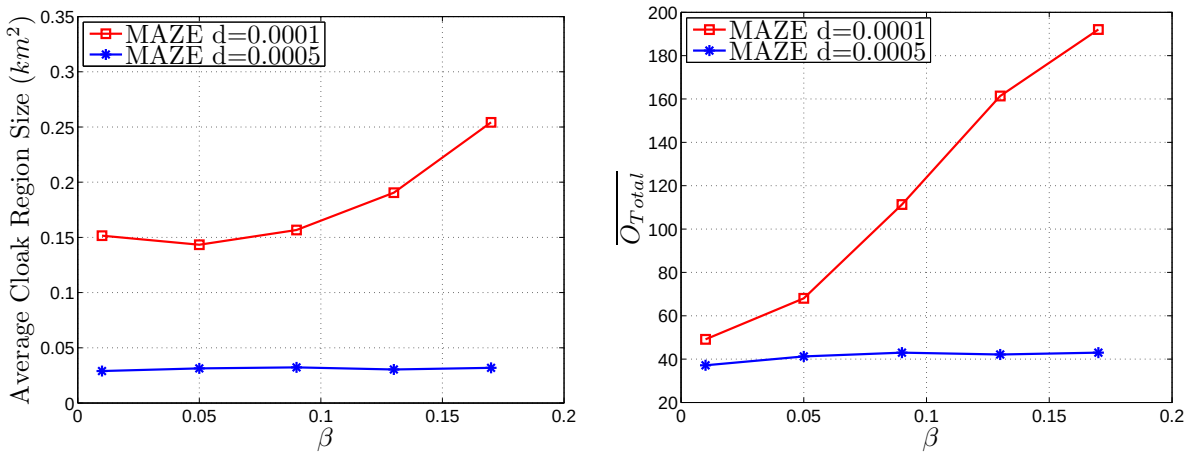

Figure 7.4: (a) Average cloaking region size as a function of parameter β, (b) Average communication overhead of MAZE as a function of parameter β.

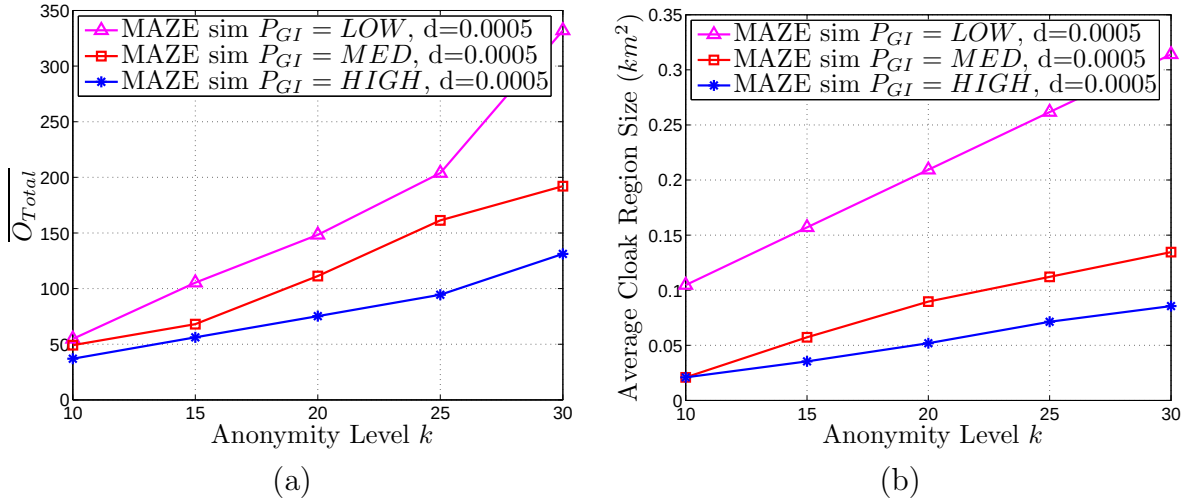

Figure 7.5: (a) Average communication overhead versus anonymity level for three different $\hat{u_i}$'s privacy requirement level, (b) Average cloaking region size versus anonymity level for three different $\hat{u_i}$'s privacy requirement level.

randomly set the privacy level of users to one of the three privacy levels and then compute the overhead of MAZE when the group initiator's privacy requirement is LOW, MED and $HIGH$ respectively. The results are shown in Figure 7.5. MAZE requires a larger cloaking region and incurs more communication overhead when his privacy requirement level is set to LOW. This is because users with a MED or $HIGH$ privacy level do not reply to $\hat{u}_i$'s request. On the other hand, when the privacy profile of the group initiator is set to $HIGH$, users of any level responds to u_i's request thus reducing the required size of the cloaking region and the communication overhead.

CHAPTER 8

Conclusions

8.0.1 Future Work

In this thesis, we addressed the problem of preserving the location privacy and user anonymity when receiving authenticated location-based services. We developed a privacy-preserving communication protocol called MAZE, that allows users to place queries to a location based server without revealing their identity, or current location beyond a certain accuracy. In addition, we showed that MAZE allows the location-based service provider to authenticate and charge any user that receives a location-based service. Our security analysis showed that MAZE preserves the location privacy and query anonymity against any of the system participants, including users and the location-based server, when these participants act independently. Furthermore, we proposed the L-MAZE protocol that protects the user privacy even if up to $(L-1)$ users collude with the location-based server.

The privacy properties of MAZE are achieved in a decentralized manner by forming P2P anonymity groups. Hence, MAZE does not employ third trusted parties such as a location anonymization server. The P2P group formation process is described for both infrastructure-based and ad hoc network architectures. To preserve the location privacy, the position of each user is blurred to a cloaking region of a desired area. To preserve anonymity, user queries are divided into several pseudo-messages using all-or-nothing transformations and then mixed among the P2P group participants using a mixnet operation and onion encryption. The application of a mixnet disassociates a user's identity from the query it submits. All pseudo-messages are received by the location-based server, that can reconstruct all

queries using the inverse all-or-nothing transformation. The server responds to the queries, with each response being sent to the intended user in a confidential manner.

Finally, we evaluated the communication and network overhead of MAZE and L-MAZE analytically and via extensive simulations. We showed that while both protocols are not the most efficient for all possible privacy requirements, they incur acceptable overhead in exchange for resilience to collusion of system participants and provision of authenticated location based services.

8.0.2 Implementation

Implementation of a mobile application for providing location based services with privacy protection by our MAZE protocol is practical.

First, technologies for building location-based service platform exists and is getting maturely developed. This platform is divided into client side and server side [51]. Client side of this platform software can be developed based on the following techniques: mobile Scalable Vector Graphics (SVG), a language for describing two-dimensional vector and mixed vector/raster graphics in XML, Bluetooth, mobile media, map slicing, map layering, and Java 2 Micro-Edition (J2ME). Sever side is developed by XML, J2ME, and MySQL [51]. With this platform, terminal user can send location-based query through his client application embedded in the terminal while GPS can determine the position of the user. Wireless communication provides transmission between terminal and server, and database stores the maps, path information, user's profile and etc. Many valuable LBS can be enabled on this platform by some suitable modification [51].

With the help of location-based service platform, more features are needed, like finding and communicating with people nearby, and settings to filter people that are not satisfy the requirements. There are various applications nowadays that can achieve these features. [36] indicates that since people are carrying mobile phones with variety of sensing components (e.g. GPS, proximity sensors, microphone, cam-

era, etc.) mobile phones can create mobile sensor networks that is capable of sensing information like where are people? And the expanding sensing capabilities of mobile phones combined with the open programming environments and platforms, typified by the Android platform and the Apple iPhone SDK, is accelerating the development of new people-centric sensing applications and systems [6]. For example, an application called CenceMe application is designed and implemented in [36] that can let sensor-enabled mobile phones automatically infer peoples sensing presence (e.g., dancing at a party with friends) and then shares this presence through social network portals such as Facebook. Other examples are, mModes Find Things or People Nearby and a new application called Blendr that combines location based social services and online dating services together. People can set up pre-filtering settings that only people meet their requirements will show up in their application. Moreover, people can send quick messages to one another through the application easily.

With the location-based service platform that consists of Terminal user, GPS, Client software, Wireless network, Database [51] and nearby people sensing features, we are able to implement our MAZE protocol for protecting user's privacy when using location-based services.

REFERENCES

[1] J. Andrews, A. Ghosh, and R. Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking*. Prentice Hall PTR, 2007.

[2] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceedings of the 17th International Conference on World Wide Web*, pages 237–246, 2008.

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

[4] A. Bushkin, S. Schaen, and U. States. *The privacy act of 1974: A Reference Manual for Compliance*. System Development Corp., 1976.

[5] L. Buttyán, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Proceedings of the 4th European Conference on Security and Privacy in Ad-Hoc and Sensor Networks*, pages 129–141. Springer-Verlag, 2007.

[6] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn. The rise of people-centric sensing. *Internet Computing, IEEE*, 12(4):12–21, 2008.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[8] D. Chaum and E. Van Heyst. Group signatures. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, pages 257–265, 1991.

[9] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *FOCS*, page 41, 1995.

[10] C. Chow, M. Mokbel, and X. Liu. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. *GeoInformatica*, pages 1–30.

[11] C. Chow, M. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proceedings of the 14th annual ACM International Symposium on Advances in Geographic Information Systems*, pages 171–178, 2006.

[12] L. Cranor. *Web privacy with P3P*. O'Reilly Media, 2002.

[13] E. Directive. 95/46/ec-the data protection directive. *Official Journal of the European Communities*, 1995.

[14] M. Duckham and L. Kulik. Location privacy and location-aware computing. *Dynamic Mobile GIS: Investigating Change in Space and Time*, pages 34–51, 2006.

[15] C. Dwyer, S. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of facebook and myspace. *Americas The*, 123, 2007.

[16] J. Freudiger, M. Raya, and J. Hubaux. Towards self-organized location privacy in mobile networks. Technical report, 2008.

[17] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *25th IEEE International Conference on Distributed Computing Systems*, pages 620–629. Ieee, 2005.

[18] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 121–132, 2008.

[19] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *Proceedings of the 16th International Conference on World Wide Web*, pages 371–380, 2007.

[20] A. Görlach, A. Heinemann, and W. Terpstra. Survey on location privacy in pervasive computing. *Privacy, Security and Trust within the Context of Pervasive Computing*, pages 23–34, 2005.

[21] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 31–42, 2003.

[22] P. Hall. *Introduction to the Theory of Coverage Processes*. 1988.

[23] U. Hengartner and P. Steenkiste. Protecting access to people location information. *Security in Pervasive Computing*, pages 222–231, 2004.

[24] H. Hu and D. Lee. Range nearest-neighbor query. *IEEE Transactions on Knowledge and Data Engineering*, pages 78–91, 2006.

[25] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, pages 1719–1733, 2007.

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

[27] T. Kölsch, L. Fritsch, M. Kohlweiss, and D. Kesdogan. Privacy for profitable location based services. *Security in Pervasive Computing*, pages 164–178, 2005.

[28] A. Küpper. *Location-based services*. Wiley Online Library, 2005.

[29] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. *Pervasive Computing*, pages 301–306, 2005.

[30] M. Langheinrich. Privacy by designprinciples of privacy-aware ubiquitous systems. In *Ubicomp 2001: Ubiquitous Computing*, pages 273–291. Springer, 2001.

[31] A. Lenhart and M. Madden. Social networking websites and teens: An overview. *Pew Internet & American Life Project*, 3, 2007.

[32] D. Li, X. Jia, and H. Liu. Energy efficient broadcast routing in static ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 3(2):144–151, 2004.

[33] P. Li, W. Peng, T. Wang, W. Ku, J. Xu, and J. Hamilton Jr. A cloaking algorithm based on spatial networks for location privacy. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 90–97.

[34] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 475–483, 2004.

[35] E. Magkos, P. Kotzanikolaou, S. Sioutas, and K. Oikonomou. A distributed privacy-preserving scheme for location-based queries. In *2010 IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 1–6, 2010.

[36] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, 2008.

[37] M. Mokbel, C. Chow, and W. Aref. The new casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Databases*, pages 763–774, 2006.

[38] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, 2(1):56–64, 2003.

[39] M. Naor and B. Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, 2005.

[40] A. Pfitzmann and M. Kohntopp. Anonymity, unobservability, and pseudonymitya proposal for terminology. In *Designing Privacy Enhancing Technologies*, pages 1–9, 2001.

[41] R. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Encryption*, pages 210–218. Springer, 1997.

[42] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. *Advances in CryptologyASIACRYPT 2001*, pages 552–565, 2001.

[43] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, pages 1010–1027, 2001.

[44] D. Stinson. Something about all or nothing (transforms). *Designs, Codes and Cryptography*, 22(2):133–138, 2001.

[45] D. Stinson. *Cryptography: theory and practice*. CRC press, 2006.

[46] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[47] H. Takabi, J. Joshi, and H. Karimi. A collaborative k-anonymity approach for location privacy in location-based services. In *Collaborative Computing: Networking, Applications and Worksharing, 2009. CollaborateCom 2009. 5th International Conference on*, pages 1–9. IEEE, 2009.

[48] N. Talukder and S. Ahamed. Preventing multi-query attack in location-based services. In *Proceedings of the third ACM Conference on Wireless Network Security*, pages 25–36, 2010.

[49] R. Vishwanathan and Y. Huang. A two-level protocol to answer private location-based queries. In *IEEE International Conference on Intelligence and Security Informatics*, pages 149–154, 2009.

[50] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1:36–72, 2005.

[51] J. Zhao, C. Zheng, and D. Zhou. Design and implementation of a location-based service platform. In *ICACT 2008. 10th International Conference on Advanced Communication Technology, 2008.*, volume 1, pages 529–533, 2008.

[52] Y. Zheng, Y. Chen, X. Xie, and W. Ma. Geolife2. 0: A location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 357–358. IEEE, 2009.