Privacy-preserving Cooperative Services for Smart Traffic

zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Dipl.-Inform. Martin Florian

aus Sofia

Tag der mündlichen Prüfung:	22. Juli 2016
Erste Gutachterin:	Prof. Dr. Martina Zitterbart Karlsruher Institut für Technologie (KIT)
Zweiter Gutachter:	Prof. Dr. Björn Scheuermann Humboldt-Universtität zu Berlin

Danksagung

DIE vorliegende Arbeit entstand in den letzten vier Jahren während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Telematik des Karlsruher Institut für Technologie (KIT). An erster Stelle gilt mein Dank Frau Prof. Dr. Martina Zitterbart, die mir durch die Anstellung an ihrem Institut die Promotion ermöglichte und mich während der gesamten Zeit mit hilfreichen Diskussionen und Ratschlägen begleitete. Ebenso bedanken möchte ich mich bei Herrn Prof. Dr. Björn Scheuermann, der sich trotz zahlreicher Verpflichtungen in Forschung und Lehre sofort bereit erklärte, das Korreferat für meine Promotion zu übernehmen.

Allen Kolleginnen und Kollegen des Instituts für Telematik möchte ich herzlich für die letzten vier Jahre danken. Mein besonderer Dank gilt Dr. Ingmar Baumgart für seine zahlreichen hilfreichen Ratschläge, die hervorragende wissenschaftliche Zusammenarbeit und dafür, dass er stets ein offenes Ohr für mich hatte. Weiterhin gilt mein Dank Dr. Bernhard Heep für die wegweisende gemeinsame Arbeit am Anfang meiner Promotion, sowie Dr. Sören Finster, für die konstruktive Kritik und die ermunternden Gespräche. Danken möchte ich auch Dr. Fabian Hartmann, Dr. Christian Hübsch, Dr. Christoph Mayer, Dr. Sebastian Mies und Dr. Jérôme Härri, die mich während meiner Studienzeit als hervorragende Betreuer in die wissenschaftliche Arbeit einführten. Zahlreiche weitere Mitarbeiter waren durch ihre Freundschaft und Diskussionsbereitschaft eine tolle Unterstützung und haben maßgeblich zu dem angenehmen Arbeitsklima am Institut beigetragen.

Ein besonderer Dank gilt auch allen Studierenden, die als Bachelor-, Master-, Studien- oder Diplomarbeiter sowie als wissenschaftliche Hilfskräfte neue Ideen eingebracht und entworfene Konzepte kritisch hinterfragt haben. Mein besonderer Dank gilt hierbei Felix Pieper, Johannes Walter, Simeon Andreev und Timon Hackenjos.

Großen Anteil am Gelingen dieser Arbeit haben nicht zuletzt auch meine Freunde und meine Familie, die mich immer wieder motiviert und auch in schwierigen Phasen stets unterstützt haben. Mein größter Dank gilt meinen Eltern, die mir die Möglichkeit zu Studium gaben und mich auf meinem Weg zur Promotion stets als Vorbilder und mit wertvollen Ratschlägen begleiteten. Ein sehr großer Dank gilt auch meiner Freundin Andreyana Andreeva, die nie die Geduld mit mir verloren hat und mir immer wieder neue Kraft spendet.

Vielen Dank euch allen!

Karlsruhe, im Oktober 2016

Zusammenfassung

Moderne Fahrzeuge sind zunehmend intelligent und miteinander vernetzt. Dies Ermöglicht neue Qualitäten von Kooperation, indem Kooperation von der Aufmerksamkeit menschlicher Nutzer entkoppelt wird. Im Rahmen von *Smart-Traffic*, zum Beispiel, können smarte Fahrzeuge autonom kooperieren um den Verkehrsfluss zu optimieren, um Dienste zu nutzen und zu erbringen und um Sensorwerte auszutauschen. Zusätzlich zu Verbesserungen der individuellen Lebensqualität, haben solche *kooperative Dienste* das Potential, zur Lösung signifikanter gesellschaftlicher Herausforderungen beizutragen. Dazu zählen, zum Beispiel, die Verminderung schädlicher Umwelteinflüsse und die Meisterung stetig wachsender Mobilitätsanforderungen.

Das wachsende Volumen an Daten, die Nutzer smarter Dienste von sich preisgeben, ermöglicht jedoch auch weitgehende Eingriffe in die *Privatsphäre* von Nutzern. Wenn Aspekte des Privatsphärenschutzes nicht frühzeitig im Systementwurf berücksichtigt werden, kann die unrechtmäßige Erfassung privater Daten nicht effektiv verhindert werden.

Im Kontext von Smart-Traffic ist insbesondere der Schutz von mit Zeitstempeln versehenen *Lokationsdaten* relevant. Also von Daten, aus denen entnommen werden kann, wo ein bestimmter Nutzer sich zu einer bestimmten Zeit aufgehalten hat. Lokationsdaten werden für die Realisierung einer Vielzahl von Smart-Traffic-Diensten benötigt, von kooperativer Verkehrsüberwachung und Verkehrsflussoptimierung bis hin zu der Auslieferung von Nachrichten basierend auf geografischen Kriterien. Zur selben Zeit können Lokationsdaten, im Kontext der Privatheit, höchst sensibel sein. Aus Ihnen kann auf Gewohnheiten, Interessen, soziale Kontakte und sogar den Gesundheitszustand und das politische Engagement von Nutzern geschlossen werden. Daher sollte die Privatheit von Lokationsdaten gewährleistet werden.

Der *Nutzen* von Smart-Traffic-Diensten ist oft direkt von dem Volumen und der Qualität der kommunizierten Daten abhängig. Dabei kann es unzureichend sein, Nutzern lediglich die anonyme oder pseudonyme Partizipation zu ermöglichen. Wie wiederholt in der Literatur gezeigt werden konnte, können, bei ausreichend hoher Samplingrate und -präzision, anonyme Lokationsdaten effektiv deanonymisiert werden, indem auf Korrelation und leicht zugängliches Kontextwissen zurückgegriffen wird. Außerdem werden explizit identifizierbare Daten oft benötigt, um den *Missbrauch* von Diensten zu detektieren und zu bestrafen.

Viele bestehende Arbeiten und in der Praxis eingesetzte Systeme verlassen sich auf die Existenz von dedizierten *vertrauenswürdigen Entitäten*, z.B. in der Form von vertrauenswürdigen Dienstanbietern. Diese bieten nicht nur ein attraktives Angriffsziel, sondern befinden sich oft auch in der Position für großflächige Einschnitte in

die Privatsphäre von Nutzern. Durch die Sammlung großer Datenmengen werden z.B. Korrelationsangriffe ermöglicht. In dieser Arbeit wird daher untersucht, wie, durch *Dezentralisierung*, die Abhängigkeit von einzelnen Entitäten aufgehoben werden kann. Zusätzlich zur Eliminierung zentralisierter Datensenken, ermöglicht Dezentralisierung auch die Realisierung von *Datenlokalität*: es kann erreicht werden, dass präzise Lokationsdaten nur mit Entitäten geteilt werden, die sich nachweislich in der Nähe befinden.

In der vorliegenden Dissertation wird auf drei funktionelle Bausteine eingegangen, die sowohl relevant für aufkommende Smart-Traffic-Dienste sind, als auch signifikante Herausforderungen bei der Balancierung von Privatsphärenschutz und Nutzen bieten. Diese sind: kooperative Planung, geografische Adressierung und die dezentrale Bereitstellung von Pseudonymen. Als wichtigen Unterschied zu vielen verwandten Arbeiten und praktisch etablierten Systemen, werden in dieser Arbeit Widersacher angenommen, die beliebige zentral kontrollierte Systemkomponenten infiltrieren können, um privatsphärenrelevante Daten zu sammeln. Dies soll die Gefahr widerspiegeln, die von Hacker-Angriffen, staatlichen Eingriffen und unehrlichen Dienstanbietern hervorgeht. Es wird allerdings angenommen, dass die Mehrzahl der Nutzer nie aktiv mit böswilliger Absicht zusammenarbeitet (z.B. um die Privatsphäre einzelner Nutzer zu kompromittieren). Die konkreten Beiträge in der vorliegenden Arbeit sind folgende:

- **Privatsphärengerechte kooperative Routenplanung [41].** Im Rahmen von kooperativer Routenplanung veröffentlichen Nutzer ihre geplanten Routen, um, gemeinschaftlich, Staus zu vermeiden und den Verkehrsfluss zu verbessern. Die zentrale Herausforderung, die in diesem Kontext in Angriff genommen wird, ist, Widersacher daran zu hindern, vergangene und zukünftige Nutzer-Lokationen erfassen zu können. Gleichzeitig soll das System in der Lage sein, sich gegen Missbrauch zu schützen, insbesondere gegen Nutzer, die falsche Pläne veröffentlichen. Als Lösung wird die Konstruktion *Promise Coins* (PCs) vorgestellt. PCs sind kryptographische Tokens angelehnt an David Chaums Entwürfe für elektronisches Bargeld. Eine begrenzte Anzahl von PCs wird an jeden Nutzer ausgestellt. Dabei müssen PCs ausgegeben werden, um Pläne zu veröffentlichen. Neue PCs werden ausgestellt, sobald Pläne wie versprochen erfüllt wurden. Dadurch verbleiben ehrliche Nutzer mit einem stabilen Vorrat an PCs, während lügende Nutzer schnell das Recht verlieren, an dem System teilzunehmen. Wie durch Analyse bestätigt, sind Pläne anonym und nicht einfach rekonstruierbar. Durch Leistungsmessungen der verwendeten Bausteine wird weiterhin die Praxistauglichkeit des vorgestellten Systems gezeigt.

- **Privatheitswahrender Langstrecken-Geocast [39, 40, 42, 43, 56].** Das System *OverDrive* wird vorgestellt - ein Overlay-basierter Geocast-Dienst für Smart-Traffic-Anwendungen. OverDrive ermöglicht die Adressierung von Fahrzeugen basierend auf geografischen Charakteristika. So können z.B. Nachrichten an alle Nutzer in einem bestimmten Gebiet ausgeliefert werden. Anders als in zentralisierten Ansätzen, werden die Lokationen von Nutzern nicht mit einzelnen Dienstanbietern geteilt, sondern mit einer Gruppe anderer Nutzer. Zusätzlich zur Eliminierung zentralisierter Datensenken, wird dabei auch *Datenlokalität* realisiert - nur Nutzer, die sich nachweislich in der Nähe befinden, empfangen präzise Lokationsdaten. Durch einen dedizierten Mechanismus zur Aufdeckung von gefälschten Lokationen, wird Datenlokalität auch dann sichergestellt, wenn Widersacher aktiv ihre eigenen Positionen fälschen. Zusätzlich zu einer Analyse der erreichten Privatheit, werden Simulationsstudien des vorgeschlagenen Systems durchgeführt. Die Ergebnisse demonstrieren, dass sowohl die großflächige, als auch die gezielte Überwachung von OverDrive-Nutzern nicht effektiv möglich ist, ohne physisch präsent zu sein. Bezüglich der erreichbaren Performanz, erweist sich OverDrive als praktikable Alternative zu traditionelleren, zentralisierten Ansätzen.

- **Sybil-resistente Pseudonymisierung und Pseudonymwechsel ohne vertrauenswürdige Dritte [44].** Das Ausstellen von Pseudonymen ist ein etablierter Ansatz, um die Privatheit von Nutzern zu gewährleisten, während Zugangskontrolle ermöglicht wird und Sybil-Angriffe verhindert werden. Um die Identifikation von Pseudonyminhabern durch durchgängige Beobachtung und Korrelation zu verhindern, müssen nicht-verkettbare Pseudonymwechsel ermöglicht werden. Existierende Ansätze zur Realisierung von Sybil-resistenter Pseudonymisierung und Pseudonymwechsel sind dabei entweder inhärent von vertrauenswürdigen Instanzen abhängig, oder führen zu einem signifikanten Rechenaufwand für beteiligte Nutzer und deren Geräte. In dieser Dissertation wird ein neuer Ansatz für Sybil-resistente Pseudonymisierung und Pseudonymwechsel entwickelt, welcher von der Existenz individueller Vertrauensanker unabhängig ist, ohne Nutzer unrealistisch stark zu belasten. Der vorgeschlagene Ansatz basiert auf der Verwendung von Blockchains, wie sie auch in dezentralen Währungen verwendet werden. Es wird ein genereller Ansatz vorgestellt, sowie der konkrete Entwurf *BitNym*, der auf dem unmodifizierten *Bitcoin*-Netzwerk aufbaut. Evaluationsergebnisse demonstrieren die praktische Realisierbarkeit des Ansatzes und zeigen, dass Anonymitätssets, die der gesamten Nutzerpopulation entsprechen, möglich sind.

Es wurde, bezüglich dem in ihnen erreichten Schutz der Privatsphäre, eine Analyse und Evaluation aller vorgestellten Bausteine durchgeführt. Dabei wird ein zweistufiger Ansatz verwendet. Zunächst werden die Bedrohungen *Detektion* und *Enthüllung* betrachtet. Hierbei werden die Art und die Qualität der Daten, die ein Widersacher sammeln kann, untersucht. Dies schließt die Identifizierung von möglichen Angriffen, mit deren Hilfe der Angreifer sein Wissen vermehren kann, mit ein. Basierend auf den Daten, die laut diesem Schritt zur Verfügung stehen, wird schließlich untersucht, inwieweit die *Verkettbarkeit* und *Identifizierung* von Datenobjekten möglich ist. Das Endziel des Widersachers ist dabei stets das Erhalten von identifizierbaren, mit Zeitstempeln versehenen Lokationsdaten.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
BTC	Bitcoin (unit of value)
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CIC	Coin Issuing Certificate
DHT	Distributed Hash Table
DoS	Denial-of-Service
ECDSA	Elliptic Curve Digital Signature Algorithm
EDGE	Enhanced Data Rates for GSM Evolution
ETSI	European Telecommunications Standards Institute
GPS	Global Positioning System
GPTx	Genesis Pseudonym Transaction
GSM	Global System for Mobile Communications
GUM	Geographic Unicast Message
HTTP	Hypertext Transfer Protocol
IA	Immediate Assignment
IAC	Initial Access Control
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LAN	Local Area Network
MAC	Media Access Control
NAT	Network Address Translation
mBTC	Millibitcoin (unit of value)
P2P	Peer-to-Peer
PA	Promise Authority
PAC	Promise Authority Certificate
PB	Promise Board
PC	Promise Coin
PCCH	Paging Channel

PET	Private Equality Testing
PPT	Private Proximity Testing
PT	Promise Token
PTP	Peer-Tor-Peer
PvC	Performance versus Cost
RA	Resource Authority
RAC	Resource Authority Certificate
RSA	Rivest-Shamir-Adleman cryptosystem
RSU	Road Side Unit
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SPV	Simple Payments Verification
STUN	Session Traversal Utilities for NAT
TLS	Transport Layer Security
TTP	Trusted Third Party
UDP	User Datagram Protocol
UTXO	Unspent Transaction Output

1. Introduction

In recent years, a rapid development of new information and communication technologies can be witnessed. In addition to promising significant quality of life improvements, these advancements have great potential for meeting long-standing societal challenges, like reducing environmental impacts and meeting ever increasing mobility demands. Ubiquitous communication technology and the increasing intelligence of things - physical objects - enable fundamentally new qualities of cooperation, as cooperation becomes decoupled from users' attention. In the domain of smart traffic, for example, users in smart vehicles can easily cooperate to optimize traffic flow, request and provide services and exchange sensor data.

With these new abilities, however, come new dangers: with the increasing amount of privacy-relevant data shared by users, new dimensions of privacy intrusions become possible. Without considering privacy issues early in the design of complex systems, resulting systems can become inherently susceptible to leaks of privacy-relevant data and exploits by determined attackers. Recurring media reports, e.g., about the activities of governmental intelligence agencies, reflect this as well as investigations into popular smart traffic systems like *Google Maps* and *Waze* [64].

Users are left with no choice but to accept their loss of privacy if they want to use novel services and retain their competitiveness and overall societal integration. Yet the right to privacy is widely acknowledged to be fundamental [62] and forms a central pillar of functioning democratic societies.

The realization of complex systems is significantly more difficult when privacy protection becomes a central design requirement. In many areas, there is still a lack of understanding about how complex functionality can be realized in a reliable and abuse-resistant manner without harming users' privacy in the face of strong adversaries. The focus of this thesis is on three functional building blocks that are especially challenging in this respect: cooperative planning, geographic addressing and the decentralized provision of pseudonymous identifiers.

1.1 Problem statement

Adversary model

In the scope of this thesis, adversary models are used for analyzing and evaluating the privacy characteristics of different systems and approaches. As an important distinction to many related works, this thesis assumes *strong adversaries* that can collude with entities controlling central system components, e.g., service operators and certificate authorities, for compromising user privacy. This assumption reflects the threat arising from security breaches, governmental subpoenas and dishonest service operators.

Unless noted otherwise, adversaries are assumed to be *external* to the majority of users, i.e., not colluding with them or controlling them. Adversaries can, however, be *internal* to a bounded number of users, e.g., via bribes or the compromising of user devices.

Cellular network operators are assumed to honor location privacy and never collude with adversaries to release the location data of users. Location privacy in cellular networks is a problem orthogonal to the challenges tackled in this thesis. The fact that cellular network providers can learn the locations of all of their customers is a drawback in the practical realization of cellular networks and not an inherent property. Given cellular network technologies that protect user locations from cellular network operators, the requirement of honest cellular network operators can be dropped for all designs discussed in this thesis.

Privacy in smart traffic

Privacy-preservation is the protection of *privacy-relevant data*, which encompasses all non-public data that is attributable to a person [26]. One of the most relevant classes of privacy-relevant data generated in the context of smart traffic is that of *location data* - information from which the location of a user at a given point in time can be inferred. Location data, e.g., in the form of timestamped location samples, is required for the provision of a wide variety of smart traffic services, from traffic monitoring and optimization to the delivery of messages based on geographic criteria. At the same time, timestamped location data is highly sensitive from a privacy standpoint. It can reveal habits, interests and social contacts as well as health status (by observing the frequency of hospital visits) and political engagement (by observing the presence at political events). Consequently, a central focus point of this thesis is the preservation of *location privacy* in cooperative smart traffic services, i.e., preventing potential adversaries from collecting identifiable location data. This is a significant challenge; cooperative smart traffic services often require location data to be shared continuously. As repeatedly shown in the literature, given a sufficient number of location samples, the reconstruction of trips and eventual identification of collected samples can become possible.

From a technical standpoint, the protection of privacy-relevant data is best realized by following the principle of *data minimization*. Data minimization can be defined as ensuring that no subject gets to know any data, privacy-relevant or not, unless it is absolutely necessary in a given system context [94]. Data minimization is challenging to realize without compromising the *utility* of complex services. For some services like abuse-resistant cooperative planning or the location-based addressability of mobile nodes, no compelling solutions are known that enable a satisfactory balance between privacy protection and utility.

In many existing works, key system tasks are put under the centralized control of single trusted entities. In addition to often enabling large-scale correlation attacks and privacy breaches, this approach results in centrally controlled components that are attractive targets for attacks and, therefore, costly to secure. Through *decentralization*, the need for centralized control over system-critical components can be eliminated, as well as the existence of system-wide data sinks. Additionally, the realization of *data locality* becomes possible, i.e., ensuring that precise location data is disclosed only to entities that are also provably nearby. Also, the problem of establishing trust anchors in systems with diverse (and potentially globally distributed) stakeholders can be avoided.

Scope of this thesis

In the scope of this thesis, two specific functional building blocks for upcoming smart traffic systems are explored. Both are difficult to reconcile with the requirement of location privacy under the assumption of strong adversaries. Namely, following problems are considered:

- Realizing *cooperative route planning*, where vehicles inform each other about planned routes to optimize traffic flow and avoid future congestions. A central challenge here is to ensure *abuse resistance*, i.e., that malicious participants cannot effectively manipulate traffic flow through false reports. On the other hand, the plans of users must remain private and non-identifiable, as they reveal planned future locations.

- Enabling long-distance *geocast*, i.e., the addressability of vehicles based on geographic criteria. Geocast enables location-based notifications and information queries to be forwarded to vehicles on the road, thus forming a base for a wide variety of smart traffic services. Without costly dedicated infrastructure, the locations of participating vehicles must be monitored in order to provide the service correctly. Without additional measures, this can enable large-scale privacy breaches.

In addition to these smart traffic-specific building blocks, a third, more universally relevant challenge is also tackled:

- Realizing *pseudonymization and pseudonym change* in a completely decentralized manner, i.e., without relying on any form of *trusted third party* (TTP) during setup and operation. Unlinkable pseudonyms that support changeover are required in a vast number of systems, both in smart traffic and beyond. Eliminating TTPs in this context poses a significant challenge, especially when considering *sybil attacks*, i.e., the simultaneous use of multiple fake identities.

If privacy-preservation is not a design goal, straightforward solutions exist for all of these challenges. This is also true if a weak adversary is assumed, e.g., by assuming the existence of an honest, completely incorruptible and universally trusted service provider.

Assumptions

This work is based on a number of assumptions which are realistic and easily met in today's world and the near future.

Since this thesis explores cooperative services, *users* are an entity of central importance. The term *user* is used interchangeably to describe both the driver of a smart vehicle and any on-board device on that vehicle. It is assumed that vehicles and devices operate on behalf of the driver and share its location during travel. This assumption makes the distinction between subjects, their (possibly autonomous) vehicles and their vehicles' on-board equipment irrelevant.

It is assumed that vehicles, and thus, users, are equipped with cellular networking interfaces (e.g., for 3G and 4G networks) and that cellular network coverage is ubiquitous. Additionally, the availability of short-range communication interfaces, as discussed in the context of vehicular networking [66], is assumed as well as the existence of GPS sensors. These requirements can, in essence, be met by using a commonly available smartphone, which makes the proposed solutions readily deployable in practice. As users of smart traffic services are typically associated with moving vehicles, this thesis assumes the existence of an abundant energy budget for users and does not consider the energy consumption of the presented solutions.

It is assumed that the majority of users are honest and cooperate willingly. More specifically, it is assumed that, while possibly selfish or curious in isolation, the majority of users will never actively collude with malicious intent.

1.2 Research questions and contributions

This thesis tackles two major questions:

1. How complex systems can be designed so that strong adversaries cannot learn any privacy-relevant data. More specifically, the focus is on preventing the collection of timestamped location data that is *identifiable* while preserving the systems' functionality and making sure that privacy measures do not enable significant abuse.

2. How *decentralization* can be applied to complex services, eliminating the requirement for single trusted entities. A special focus in on the implications of decentralization on trust and its interplay with privacy.

This thesis identifies several important open challenges in the context of smart traffic. The key contribution consists of developing and evaluating original solutions to these challenges. The presented solutions are readily deployable using existing infrastructure and common consumer equipment like smartphones. The specific contributions are the following:

- **Privacy-preserving cooperative route planning.** In cooperative route planning approaches, vehicles share their planned routes to cooperatively avoid congestions and optimize traffic flow. The main challenge tackled in this context is the preservation of user privacy (i.e., preventing adversaries from determining the past and future locations of individual users) while ensuring that the system is resistant to abuse, i.e., to malicious users that distribute false plans. Addressing this challenge, this thesis introduces the construction of *promise coins* (PCs) modeled after Chaum's *electronic cash* design [16]. A limited supply of PCs is issued to each user, with the spending of PCs being required for publishing plans. PCs are reimbursed to users when plans are realized as claimed, thus maintaining a stable PC supply at honest users. Malicious users, on the other hand, lose their right to participate in the system by being unable to replenish their PC supply.

- **Location privacy in long-distance geocast.** *OverDrive*, an overlay-based geocast service for smart traffic applications, is presented. OverDrive enables the addressing of vehicles based on geographic characteristics, e.g., the delivery of messages to all users within a given area. Unlike in centralized approaches, user locations are not shared with a single service provider but with a dynamic set of other users. In addition to, in this way, eliminating centralized data sinks, data minimization is realized through *data locality* - only users that are provably close receive precise location data.

- **Sybil-resistant pseudonymization and pseudonym change.** The issuing of pseudonyms is an established approach for protecting the privacy of users while limiting access and preventing sybil attacks. To prevent pseudonym deanonymization through continuous observation and correlation, frequent and unlinkable pseudonym changes must be enabled. Existing approaches for realizing sybil-resistant pseudonymization and pseudonym change are either inherently dependent on trusted third parties (TTPs) or involve significant computation overhead at end-user devices. This thesis investigates a novel, TTP-independent approach towards sybil-resistant pseudonymization and pseudonym change. The proposal is based on the use of cryptocurrency block chains as general-purpose, append-only bulletin boards. A general approach is presented as well as *BitNym*, a specific design based on the unmodified *Bitcoin* network [84].

In addition to performance evaluations, privacy analyses and evaluations of the proposed building blocks are conducted. Based on the adversary model outlined in

Section 1.1, the type and quality of data an adversary is able to collect using different attacks is analyzed. A simulation approach is used to capture the complex effects arising in realistic scenarios (e.g., the geographic distribution of users in vehicular traffic scenarios) as well as to quantify the trade-offs between privacy protection and performance.

Based on the data available to an adversary (according to analysis and simulation), it is analyzed to what extend identification is possible. In this context, it is discussed to what extend data items associated with the same user can be linked together, as this enables profiling attacks and greatly increases an adversary's knowledge gain after identifying individual data items (e.g., using additional context knowledge).

1.3 Structure of this thesis

The foundations of this thesis are introduced in Chapter 2. Amongst other things, the chapter discusses existing techniques for the preservation of location privacy as well as different approaches towards decentralization. The assumptions and adversary models underlying the remainder of the thesis are elaborated in Chapter 3. Additionally, a systematic approach for analyzing the privacy characteristics of the proposed systems is described.

Three bodies of original work are then presented. Chapter 4 describes an investigation into privacy-preserving cooperative route planning as well as the novel *promise coin* construction. Chapter 5 presents the privacy-preserving long-distance geocast system *OverDrive*. Chapter 6 addresses the challenge of realizing pseudonymization and pseudonym change without TTPs and introduces the *BitNym* system.

The thesis then concludes, in Chapter 7, with a summary of results and a discussion of future perspectives for research and application.

The thesis features three appendices. Appendix A presents a measurement-based investigation into the connectivity- and privacy-related properties of deployed cellular networks. Appendix B describes the functionality and implementation of an interactive visualization of the OverDrive system. Appendix C lists simulation parameters used for generating the evaluation results presented in Chapters 5 and 6.

2. Privacy and trust in cooperative smart traffic services

This chapter introduces foundations that are relevant for the presented thesis. Upon introducing smart traffic and the concept of cooperative services in Section 2.1, privacy-related challenges applying in this context are discussed in the subsequent Section 2.2. Common privacy threats and protection goals are introduced and specific threats related to the sharing of timestamped location data are considered. In Section 2.3, existing approaches for protecting timestamped location data are outlined and compared.

As the avoidance of centralized architectures, e.g., for eliminating the requirement of trusting singular entities, is an important theme throughout this thesis, different approaches to decentralization and their impact on trust modelling are discussed in Section 2.4.

Section 2.5 discusses existing proposals for limiting the impact of malicious users while preserving the privacy of honest ones. Notably, this includes approaches for bounding the number of identities that users are able to simultaneously maintain within a given system.

Lastly, the chapter also touches upon several related challenges. The connectivity and privacy characteristics of contemporary cellular networks are discussed as well as the practical possibilities for establishing private and secure communication channels over the Internet. Both of these challenges are discussed in Section 2.6.

2.1 Smart traffic and cooperation

Today's street traffic is still largely inefficient. Overburdened roads lead to congestions and unnecessary pollution. Accidents are common and the possibilities for social interactions with other drivers are limited. *Smart traffic*, the application of modern information and communication technology to the vehicular traffic scenario, has tremendous potential for improving such issues. Through the increasing interconnection of smart vehicles, for example, the capabilities of traffic participants to cooperate can be significantly enhanced. This section, following an outline of the capabilities of modern and near-future vehicles, discusses several examples of how improvements can be realized. Typical smart traffic applications are introduced as well as the concept of cooperative services and its application to the smart traffic context.

2.1.1 Smart cars

Modern cars are rapidly increasing in complexity, becoming equipped with a growing number of sensors and communication capabilities. On the sensor side, this includes GPS positioning sensors (for navigation), weather (e.g., temperature and rain) sensors, road condition sensors and multiple cameras (e.g., for parking assistance). Cellular communication interfaces for 3G, 4G and soon 5G networks are increasingly deployed for enabling richer traffic information and infotainment services as well as enabling maintenance-related tasks (e.g., remote diagnoses). Short-range radio interfaces for enabling ad hoc vehicular networking are being standardized and will likely be included in new vehicles in the near future [35, 66]. For receiving wide-area broadcasts, classic FM radio and the (60 bit/s) *traffic message channel* running over it can be leveraged as well.

As an important addition to these sensory and communication capabilities, modern cars are also equipped with an increasing amount of intelligence. From automatic navigation and engine management to safety-related features like automatic breaking, cars are increasingly software-controlled. The ongoing research into autonomous driving is especially interesting in this context; given a car that can navigate traffic and reach its destination by itself, the need for an actual human driver diminishes. The lack of a human driver does not, however, alleviate potential privacy concerns, as subjects are still present in the form of passengers and, potentially, car owners.

While some of the listed capabilities are not yet widely deployed on today's streets, many of them can be emulated using common smartphones. Stock smartphone features that can be leveraged include cellular network interfaces, GPS sensors, wireless LAN and Bluetooth interfaces (for short-range communication [12]) and acceleration sensors. Modern smartphones also feature substantial computing resources.

One of the main limitations of smartphone-class devices - their constrained energy budget - is not relevant in the vehicular traffic setting. Here, abundant energy resources are typically available (e.g., when considering devices located in moving cars).

2.1.2 Typical smart traffic applications

In the following, the potential behind smart traffic is motivated based on several example classes of smart traffic applications.

2.1.2.1 Safety

Increasing the *safety* of travel is a constant driving force behind vehicular innovation. Safety applications, e.g., the sending of *cooperative awareness messages* for assisting in the detection of hazards, are also one of the main motivators behind the development of short-range vehicular networking standards and protocols [66]. Local ad hoc communication is ideal for safety-related applications, as safety-critical information is usually time-critical (necessitating low communication latencies) and only of local interest. Wide-area warnings are, however, possible as well and can be used for adapting route choices (e.g., for avoiding icy roads).

2.1.2.2 Traffic optimization

In addition to improving the mobility of subjects and goods, *traffic optimization* is an important contributor to reducing the negative environmental impact of developed societies. Historically, optimizing the traffic flow has been a challenge tackled mainly by traffic authorities and city planners. With the rise of the smart traffic paradigm, however, the role of individual users increases. By reporting the locally observed traffic, users act as traffic sensors and assist in *traffic monitoring*. Additionally, through automatic *vehicular navigation* and the possibility for two-way communication with vehicles, traffic can be influenced in a more fine-granular manner.

Vehicular navigation, specifically, is one of the most widely used smart traffic applications today. Considering major deployed systems, two different evolution stages can be identified. Using *static route planning*, navigation devices determine their own current location (e.g., via GPS) and determine the shortest path to the destination based on static map material. *Dynamic route planning* is performed similarly to static navigation, but traffic updates, e.g., about congestions and construction sites, are also taken into account during route calculation. Dynamic route planning is, for the most part, the current status quo in automatic navigation and used in a wide range of navigation devices and smartphone applications.

Different possibilities exist for obtaining the current traffic state for dynamic route planning. Established approaches like police reports or infrastructure-based traffic monitoring (e.g., via inductive sensors in the road) are suboptimal in terms of cost, delay and precision. Leveraging input from vehicles on the road is significantly more promising. Such cooperative methods are discussed in Section 2.1.3.

2.1.2.3 Parking space discovery

While traffic optimization can improve the flow of traffic, shortening parking space search times through improved *parking space discovery* has the potential for reducing the overall traffic volume. Cruising for parking spaces was found to generate significant amounts of traffic [103]. In practice, payment-based reservation systems and locally deployed sensors are often used for determining free parking spaces. This information can then be shared with individual vehicles. However, payment-based parking solutions are not applicable everywhere and the deployment and maintenance of sensors can require significant investments.

2.1.2.4 Social networking

Despite large amounts of vehicular traffic, users on the road can rarely connect with each other while travelling. The popularity of *vehicular social networking* applications like *Waze*[1] as well as the communication habits of professional drivers (e.g., truck drivers) both hint at a general desire for interacting with fellow travelers. Modern communication technologies can help, e.g., by providing discovery mechanisms for communication partners with a similar location or route. In addition to facilitating social interactions within a geographically relevant scope, vehicular social networking also enables the exchange of practical traffic-related information that is cumbersome to convey in an automatic manner.

2.1.3 Cooperative services

As a general concept, *cooperative services* are services in which individual subjects (respectively devices acting on their behalf) work together for realizing the main provided functionality. In contrast to collaboration, the goals achieved through cooperation are common to all participants. Thus, cooperating users act selfishly, as they benefit from the end-result of collaboration. Cooperation can range from physical actions to the sharing of information. Many smart traffic applications discussed so far can be realized or augmented by leveraging user cooperation. In fact, most smart traffic services are inherently cooperative, as information and actions from other participants are required to achieve benefits.

In the following, common forms of user cooperation are discussed as well as their application to the smart traffic context. The discussed forms of cooperation are:

- information exchange
- planning
- one-to-one cooperation
- data distribution

2.1.3.1 Information exchange

Through *information exchange*, individuals can augment each other's view of the world. In this way, more informed decisions are enabled for all involved parties.

An application example of this pattern is traffic monitoring, respectively dynamic route planning. Here, vehicles can act as traffic sensors by transmitting their location, speed and driving direction. By collecting such *floating car data* from a large number of participants on the road, the current traffic state can be estimated. This is the approach used in, amongst others, the popular smartphone-based navigation service *Google Maps*[2].

Another example for cooperative services based on information exchange are *crowdsensing* systems [46]. Similarly to traffic monitoring, sensory input is collected from

[1] https://waze.com/

[2] https://maps.google.com/

participants. Crowdsensing can be used for anything from environmental studies (e.g., by collecting information about air quality) and the creation of fine-granular weather maps up to improving the safety of individuals in cities, by cooperatively identifying dangerous areas [21]. In this context, it should be noted that vehicular safety approaches based on short-range ad hoc networking are also centered around the idea of information exchange [66].

2.1.3.2 Planning

Cooperation is also possible in the form of cooperative *planning*, i.e., the coordination of individual actions. In a smart traffic context, this can most prominently take the form of *cooperative route planning*. Here, in addition to static map material and information about the current traffic state, the destinations and planned routes of other traffic participants are taken into account during route planning. In this way, the future traffic situation can be anticipated [23, 34].

Cooperative planning can also be applied for allocating available parking spaces and charging stations for electric cars.

2.1.3.3 One-to-one cooperation

As an alternative to more global approaches, *one-to-one cooperation* is possible as well. Individual participants can query each other for information and engage in *social interactions*.

In contrast to the information exchange concept discussed earlier, one-to-one cooperation enables the request-based, on-demand sharing of information. Additionally, the provision of services requiring more human attention is possible. This can include services that involve physical actions, e.g., a change in route as in [48].

2.1.3.4 Data distribution

Up to this point, mechanisms for distributing data to a set of recipients were implicitly assumed. In practice, such communication channels are often realized using centralized services or dedicated infrastructure. However, *data distribution* can also be realized in a cooperative manner by participating individuals.

An example for this approach is classic peer-to-peer networking (cf. Section 2.4.3). Also, many works exist that leverage data distribution via vehicular ad hoc networks, e.g., [83, 118]. As an effect of cooperative data distribution, neither revenue generation nor explicit investments are required for maintaining the availability of services. Instead, participating users contribute a fraction of their own communication, computation and storage resources.

2.2 Location privacy

As an important aspect of cooperation, users need to generate and share data that is potentially *privacy-relevant*. The term privacy-relevant encompasses all non-public data that is attributable to a natural person [26]. In the context of smart traffic, this most distinctively involves timestamped *location data*, as it is commonly required for realizing services related to moving vehicles and traffic. Considering cooperative

Threat	**Protection goal(s)**
*l*inkability	unlinkability
*i*dentifiability	anonymity, pseudonymity
*n*on-repudiation	plausible deniability
*d*etectability	undetectability, unobservability
information *d*isclosure	confidentiality
content *u*nawareness	content awareness, transparency
policy and consent *n*on-compliance	policy and consent compliance

Table 2.1 LINDDUN threats and corresponding protection goals.

traffic monitoring, for example, the transmission of timestamped location data is mandatory for reporting the locally observed traffic state.

In the following, based on a more general discussion of the concept of privacy, an overview is given of the challenges involved in protecting location data while maintaining service utility.

2.2.1 Threats and protection goals in general

Despite of its importance for maintaining individual and societal freedom and its recognition as a fundamental human right [62], the concept of “privacy” is highly vague and difficult to grasp with a precise technical definition. Different approaches to concretizing its extent have been proposed, many of which are rooted in legal definitions (see, for example, [104] and the excellent overview in [26]). In [94], Pfitzmann and Hansen propose an extensive terminology for discussing privacy-related topics in a technical context. Their work proposes precise definitions for concepts like anonymity, unlinkability and pseudonymity and is widely used in the privacy research literature.

Comprehensive analysis frameworks for identifying privacy threats have also been proposed, most notably the *LINDDUN* methodology [28, 119]. LINDDUN defines seven distinct privacy threats (and their corresponding protection goals) that can be used for reasoning about the privacy properties of a system. These are predominantly rooted in the definitions proposed by Pfitzmann and Hansen as well as in legal requirements applying to service operators. In the following, due to its systematic structure and large scope (it includes several threats not mentioned in [94], for example), the LINDDUN set of privacy threats is used as a base for concretizing the concept of privacy. The threats defined by LINDDUN are: *linkability*, *identifiability*, *non-repudiation*, *detectability*, information *disclosure*, content *unawareness* and policy and consent *non-compliance*. These threats are also listed in Table 2.1, together with their respective protection goals. Additionally, the four threats most relevant in the context of this thesis are highlighted.

For describing the LINDDUN threats, two additional concepts need to be introduced first. An *item of interest* is a generic placeholder for an information item. Typical examples can include shared sensing samples, observed actions or user identities. While items of interest can be privacy-relevant to various degrees (e.g., a passport photo versus a private party photo), a fine-granular distinction in this respect is avoided in this thesis. An *adversary* is a fictive entity used for analysis that follows a set of malicious goals. In the scope of this section, for example, these goals are to realize the threats associated with each of the discussed protection goals. The capabilities of adversaries are determined by a-priori assumptions and form the basis for security and privacy analyses and evaluations. For the remainder of this chapter, if no more specific description is given, any entity external to a subject is considered its potential adversary[3].

In the following, the threats addressed in the LINDDUN framework are discussed together with their respective protection goals. As a base, the definitions from [28] and [119] are used[4].

2.2.1.1 Linkability

The *linkability* of two or more items of interest allows an adversary to sufficiently distinguish whether they are related or not. For example, two legal documents signed by the same subject are clearly linkable between each other and to their signer. In addition to such explicit linkability via a subject identifier, links between items of interest can also be inferred via context-specific traits like timing or similarity. If, for example, a ball is found next to a broken window, a link between the ball and the act of breaking the window is reasonable. The risk from such *linking attacks*, i.e., attempts to infer links between not explicitly associated items, increases with the amount (or *richness*) of information comprising the individual items. In the scope of this chapter, the case of linking items of timestamped location data is discussed in greater detail.

The protection goal associated with linkability is *unlinkability*, i.e., ensuring that no linking between a given set of items is possible for a given adversary.

In the scope of this thesis, multiple approaches are discussed, proposed and evaluated for achieving unlinkability in different system contexts.

2.2.1.2 Identifiability

The *identifiability* of an item of interest enables an adversary to create a link between the item of interest and a subject identity. Thus, identifiability can be seen as a special form of linkability. Considering again the example of signed legal documents, a signature enables an unambiguous link between the document and a subject. In effect, correctly signed documents are clearly identifiable.

[3] However, the majority of subjects is usually assumed to not collude with malicious intent. Trust modelling is further discussed in Section 2.4.

[4] Note that LINDDUN is frequently updated. Current materials about the methodology can be found at: `https://distrinet.cs.kuleuven.be/software/linddun`

Identifiability is arguably the most important threat discussed here. Without identifiability, the privacy-relevance of data can be disputed as no attribution to a subject is possible (for the considered adversary model). Consequently, the identifiability of sensitive information can lead to significant real-world harm for the concerned users, ranging from unwanted personalized advertisements to financial harm (e.g., an increase in insurance fees due to identifiable information about driving behavior or health status) and actual physical harm (e.g., due to the repression of subjects based on political, sexual or religious orientation).

Identifiability is tightly related to the protection goal of *anonymity*. A subject is said to be *anonymous* if an adversary cannot distinguish it within a set of subjects, the *anonymity set* [94]. Likewise, an item of interest can be said to be anonymous or *anonymized* if it cannot be linked to a subject within a set of subjects. The size of the anonymity set is a common metric for evaluating the strength of the achieved anonymity, i.e., the *anonymity level*. The term *k-anonymity* is also often used for describing the same metric, with *k* denoting the size of the anonymity set.

A related notion which is frequently used in the scope of this thesis is that of *pseudonymity*. An item of interest associated with a subject is pseudonymous or *pseudonymized* if the adversary can link it only to a *pseudonym*, an identifier other than the subject's actual identity. In many systems, the use of some sort of identifiers is unavoidable, e.g., when the interaction with a subject involves several rounds of communication. A central practical difference between pseudonymity and anonymity is the fact that pseudonymity implies the linkability of all items of interest associated with the same pseudonym. Consequently, if each pseudonym in a given system is associated to only one item of interest each, the term anonymity can be used as well.

It should be noted that, since it enables linkability, pseudonymity can be difficult to maintain. A plethora of works exist that deanonymize datasets based on linkability within them [50, 68, 85, 86, 109, 123]. A straightforward approach is using auxiliary *context knowledge* that helps with the identification of a single item of interest. For example, a pseudonymous user might accidentally use his real identity in one interaction. If only one single item of interest can be identified, all items that are linkable to it become deanonymized as well. For limiting the risk from identification, it is, thus, desirable to minimize linkability, e.g., through the frequent change of pseudonyms.

Achieving anonymity and effective pseudonymity in different smart traffic systems is a major area tackled in this thesis and one of its main motivations.

2.2.1.3 Non-repudiation

The *non-repudiation* threat applies when an adversary has irrefutable evidence concerning the occurrence or non-occurrence of an event or action. Non-repudiation can be viewed as an extreme case of linkability in which an adversary can create a link with absolute certainty and prove its existence to an independent third party, the so called *judge*. For example, in a legal context, the signature on a document makes the actions of reading and accepting it non-repudiable for the signer. An adversary in possession of the signed document can use it as proof in front of an actual judge, triggering potential legal consequences for the subject. The opposite to non-repudiation, and the respective protection goal, is *plausible deniability*.

Plausible deniability is especially relevant in scenarios where the punishments resulting from the judge's decision are potentially not justified within a larger context. For example, the judge might operate within the legal framework of an oppressive government that punishes the expression of critical opinions (a classic motivation for off-the-record communication). Voting (e.g., in parliamentary elections) is another relevant scenario. Here, plausible deniability is mandatory for ensuring the freedom of choice of voters and preventing vote buying.

In the scope of this thesis, no scenarios are considered in which, based on the considered adversary models, the modelling of an external judge entity is relevant. Thus, non-repudiation and plausible deniability is not considered in detail. However, the limiting of abuse, e.g., by punishing malicious users, is an important theme in this thesis. Approaches are investigated that enable such regulation without requiring the explicit (non-repudiable) accountability of subjects.

2.2.1.4 Detectability

The *detectability* of an item of interest refers to the adversary's ability to sufficiently distinguish whether it exists or not. Consequently, *undetectability*, the inability to make such a distinction, is the main associated protection goal. Consider, as an example, the passing of paper messages in a classroom setting. Ideally, the teacher (the adversary in this case) is unable to detect that such messages exist and are being exchanged.

The concept of *unobservability* can also be used in the context of detectability. It describes a stronger form of undetectability: an unobservable item of interest is undetectable by all entities but the ones directly associated with it, and the subjects associated with it are, additionally, anonymous to each other. Recalling the last example, consider the leaving of an anonymous paper message at a hidden location. Only the author of the message and the person finding it know of the message's existence and content. Neither of them knows the identity of the other.

Even if no details about an item of interest are known to an adversary, its existence can still represent a privacy-relevant piece of data. Communication *metadata* is a relevant example in this context: the fact that a subject is communicating with another might already disclose information about its identity and the relationship between the two. As an additional argument for undetectability and unobservability, the two concepts also imply that the content of items of interest (e.g., the content of messages) is protected as well.

Detectability plays an important role in the scope of this thesis. It is mainly prevented through ensuring that only limited amounts of data are shared with potential adversaries, e.g., through considering decentralized approaches for system design.

2.2.1.5 Information disclosure

The information *disclosure* threat describes the exposing of the content of an item of interest to an entity unauthorized to see it, i.e., an adversary. It corresponds to the protection goal of *confidentiality* which is also relevant for security. Examples for information disclosure are the reading of intercepted communications. Consequently, encryption is commonly used for preventing such information leaks.

Compared to undetectability and unobservability, confidentiality protects only the content of the item of interest. The adversary is still aware of the item of interest's existence and can potentially link it to subjects or other items of interest.

In the scope of this thesis, two cases are considered in the context of confidentiality. First, the case in which information must be transmitted over a potentially insecure medium. In this case, encryption can usually be used for protecting the content against adversaries that are external to all participants in the communication. The second case is that items of interest need to be shared with potential adversaries for realizing services. This is a common case in the context of cooperative services, as communication (e.g., involving privacy-relevant data) is usually required for enabling cooperation.

2.2.1.6 Content unawareness

Content *unawareness* is a broad concept focusing on the interaction of humans with complex information and communication systems. Wuyts and Joosen define it as "not understanding the consequences of sharing personal information in the past, present of future" [119]. Without content *awareness*, subjects do not understand the implications of sharing privacy-relevant data or how it is generated and collected. An important specific threat associated with this lack of understanding is that too much privacy-relevant data is leaked to potential adversaries. Even if this information is of no use in the present, there is no reliable way for it to be "forgotten" in the future, so that harms can materialize at a later date. Common approaches for preventing content unawareness include guaranteeing high degrees of *transparency* about the collection and processing of data. The human side of things, i.e., education, informed consent and suitable user interface design, are arguably the most important aspects in this context.

The challenge of protecting against content unawareness is relevant mainly at the point where human users interact with technology, e.g., during the decision whether a given system should be used at all or not. Consequently, it exceeds the scope of this thesis and is not considered further.

2.2.1.7 Policy and consent non-compliance

Policy and consent *non-compliance* means that one of the entities processing potentially privacy-sensitive data does not act in the manner advertised in its privacy policy, the consent signed by users or the active data protection legislation. In essence, it can be viewed as the threat that the employed trust model is wrong in some aspect, i.e., that some trusted entity turns into an adversary. Steps can be taken within organizations and legislative areas to minimize this risk, e.g., in the form of frequent audits by independent controllers.

However, such processes are out of the scope of this thesis. Instead, the focus is on determining the impact of individual entities becoming malicious, as well as on designing systems in such a way that individual malicious entities are limited in the amount of harm they can inflict. Policy and consent non-compliance is, thus, not considered as a threat but instead used (to varying degrees) as an underlying assumption during the analysis and evaluation of systems in respect to the remaining privacy threats.

2.2.2 Location samples and localization

In the scope of this thesis, a *location sample* is a unit of timestamped location data, i.e., a tuple of a location and time value.

Locations can be represented in several different ways. The most commonly used in the context of smart traffic is via geographic coordinates, e.g., pairs of latitude and longitude ("49.01256, 8.40792"). This is also the representation that is implicitly assumed in the remainder of this thesis. Alternative forms of representation include context-specific statements ("at the SCC building", "20 meters north from here") and hierarchical models ("Germany/Karlsruhe/KIT/Building 20.20.").

Localization is the process of determining a given entity's location. It can be performed in an *implicit* manner by external entities that are aware of their own location and are able to observe the entity that is to be localized. For example, cellular network operators know the location of base stations under their control and can observe the radio communication of users within their network. If a base station is within the communication range of a user, the operator can determine the user's location on a cell-level granularity. By evaluating the signal strength of the user's signal at neighboring base stations, an even more precise localization is possible.

In order for an entity to learn its own location, a different, *explicit* localization approach is more commonly used. Here, the entity actively collects information that it can use for inferring its own location. The *global positioning system* (GPS) is the most prominent and widely used example from this category. Here, entities calculate their location based on beacons from multiple satellites in the Earth's orbit. Note that this is a passive form of localization, in the sense that entities can determine their position without leaking any information (and most importantly their current location) to third parties. It places entities in control over the granularity and sampling rate of location samples containing their own location. In contrast, if implicit localization is used or possible, only the sampling rate can be influenced in some cases. For example, silent periods can be introduced during which radio interfaces are switched off.

In practice, GPS-based localization can be augmented by introducing additional localization information. Notable from a privacy standpoint are cloud-based approaches. Here, raw GPS sensor readings are transmitted to centrally operated servers, often in addition to other information like the set of visible cellular network base stations and wireless LAN hotspots. The service operator then estimates the entities' locations and sends them back to them. In effect, localization speedups and an increased accuracy can be achieved. Such techniques are used by default, for example, on Android smartphones [64]. From a privacy standpoint, however, the usage of this approach is not recommended, as the localization provider learns the locations of users with significant precision.

2.2.3 Privacy-relevance of identifiable location samples

Identifiable location samples (i.e., location samples that can be linked to used identities) are highly privacy-relevant and can enable wide-ranging harms to users if leaked to an adversary. Consider the location samples generated by a regular citizen on any given day. Typically, they include the location of their home and locations where they spend their free time. Through inference, significant amounts of additional information can be extracted from such data. Information about visited shops or recreational areas can help create detailed user profiles for targeted advertisement (that might not always help subjects make good decisions). Hospital visits may hint towards sensitive health conditions and may, thus, be interesting to know for insurance companies. Co-location with other users can allow the reconstruction of social ties. Co-location with political demonstrations can reveal opinions and assist oppressive governments with the prosecution of dissidents.

Based only on these few examples for concrete potential harms, it can be concluded that identifiable location samples are highly sensitive from a privacy standpoint. Systems relying on the sharing of location samples must be designed in such a way that potential adversaries cannot collect identifiable location samples.

2.2.4 Identification attacks

A non-obvious, yet relevant, approach for attaining identifiable location samples is by using inference techniques and additional context knowledge on otherwise unidentifiable data. Several such *identification attacks* have been proposed, focusing mostly on the case that data on taken *trips* is available, i.e., sets of linked location samples.

The reconstruction of trips is trivial if pseudonyms are used that are changed only rarely: all location samples shared using a given pseudonym have obviously been shared by the same user. However, given a high enough sampling rate (i.e., short intervals between shared samples) and a high accuracy of the shared locations, effective *linking attacks* on location samples are possible even if pseudonyms are changed regularly [117].

Several approaches have been proposed for inferring identities based on trips, i.e., groups of linkable location samples. In [68], for example, Krumm proposes to identify home locations first (using clustering methods and time-of-day heuristics), from which identities can be inferred using easily accessible context knowledge. Within the considered evaluation scenario, the correct home locations could be determined with a mean error of around 60 meters. Concerning the inference of identities from home locations, Golle and Partridge [50] found that identification is possible with a significant probability if the work location of a subject can also be inferred from the present trip. For example, they found that knowledge about an unknown subject's home and work locations are enough to (on average) uniquely identify it within the U.S. working population. The level of anonymity was found to increase with the decreasing precision of available home and work locations. However, a granularity on the level of counties is required for achieving significant anonymity sets. In conclusion, the authors suggest that home and work personas need to be kept unlinkable.

In addition to home and work locations, another form of context knowledge that can aid identification are physical observations, e.g., in the form of timestamped photos or videos. By correlating such observations with pseudonymously reported locations, the corresponding pseudonyms (and all past and future location updates shared using them) can be linked to the respective users. In [109], for example, timestamped photos were used for linking pseudonymized taxi locations with the identities of celebrity passengers. Since the location samples reported by a given taxi were linkable to each other, the destinations of the passenger's trips could be inferred.

The identification of individual location samples can also be possible by analyzing communication metadata. For example, employed IP addresses might be used for inferring identities. Challenges and solutions related to the protection of communication metadata are discussed in Section 2.6.

2.3 Data minimization

A straightforward approach for preserving privacy is by not interacting with other entities at all. Upcoming smart traffic systems, however, depend on interactions and promise significant benefits for both individuals and society, so that their realization is desirable. This outlines a central tension field: balancing the protection of privacy-relevant data with the *utility* of the resulting systems. The principle of *data minimization* is an established approach for addressing this tension [94]. Data minimization can be defined as ensuring that no subject gets to know any data, privacy-relevant or not, unless it is absolutely necessary in a given system context.

In this section, following a motivation of relevant protection goals, related work on data minimization in smart traffic is presented and discussed. Note that the listed works are only a small relevant subset of the much larger field of location privacy and privacy in location-based systems. A more thorough overview of the field, that is not focused on the smart traffic scenario, is given, for example, by Wernke et al. in [115]. The related questions of reducing trust requirements through decentralization and preventing abuse are discussed in the separate Sections 2.4 and 2.5.

2.3.1 Protection goals

For motivating the choice of related works presented in the following, the general protection goals from Section 2.2.1 are now discussed in the context of location privacy, i.e., with location samples as the main item of interest. This discussion forms the base for the analysis and evaluation approach used in the main part of this thesis.

For all scenarios and approaches discussed in the scope of this thesis, the main considered privacy-related threat is the leaking of identifiable location samples to potential adversaries.

If location samples are undetectable by an adversary (e.g., by not generating or sharing them in the first place) or if their confidentiality is maintained (e.g., the adversary only learns them in an encrypted form), no further steps can be taken towards obtaining identifiable location samples. Ensuring the *undetectability and confidentiality*[5] of location samples is, thus, the first line of defense considered here. However, this is difficult to achieve in the smart traffic context, as location samples must often be shared for realizing services.

If location samples become disclosed to a potential adversary, he can now try to identify them using additional information. In the simplest case, location samples can have been shared using an identifiable pseudonym or an identifiable communication address. However, the adversary can also use context knowledge and statistical methods for performing the identification. As a second line of protection, this direct identifiability of location samples must be prevented, i.e., the *anonymity* or *pseudonymity* of location samples must be established and preserved.

If the adversary can learn several location samples that he cannot identify directly, his next possibility is to attempt to discover links between them. The availability of reconstructed trips of linked location samples enables an additional range of possible identification attacks. Additionally, should one location sample be identifiable directly, any samples linkable to it become identifiable as well. Thus, the *unlinkability* of location samples must be ensured.

If deployed protection measures fail and identifiable location samples become available to the adversary, a last line of defense can consist in reducing the privacy-relevance of the shared samples. For example, samples concerning especially sensitive information (e.g., hospital visits) can be omitted or their precision reduced. In the scope of this thesis, such schemes are not considered and all location samples are treated equally for analysis purposes, i.e., independently of their perceived sensitivity. However, approaches for reducing the precision of location samples are discussed in the context of protecting against identification and linkability.

Note that the non-repudiation, unawareness and non-compliance threats are also not considered further here (cf. Section 2.2.1 and Table 2.1). They do not map well to the considered scenarios and their consideration exceeds the scope of the thesis. Note also that adversaries considered in the following are mainly internal, in the sense that they collude with entities directly participating in the realization of the considered service. The protection against external adversaries can usually be re-

[5] Note that the two protection goals are deliberately joined into one here. The threat that location samples are detectable by an adversary but not disclosed to him does not map well to the smart traffic scenario.

alized in a straightforward manner by leveraging communication encryption and cryptographic authentication. It is, thus, not considered in detail.

2.3.2 Undetectability and confidentiality

The undetectability and confidentiality of location samples is achieved if a potential adversary cannot learn about their existence and content. Two basics approaches for achieving this exist:

1. Reducing the volume of shared location samples.
2. Reducing the possibilities for adversaries to intercept shared location samples.

Concerning the first approach, the basic reasoning is that if less data is shared, more data remains undetectable and confidential. Clearly, simply reducing the data volume will deteriorate the utility of most services. Trade-offs in this respect are highly application- and scenario-specific. In [59], for example, Hoh et al. describe an approach for the privacy preserving collection of location samples for traffic monitoring purposes. Location updates are communicated only every couple of hundred meters, when passing a previously placed virtual trip line. While a significant improvement to the continuous collection of user locations (as practiced in many commercial systems [64]), the available information is still sufficient for assessing the current traffic state with high precision.

Reducing the opportunities during which potential adversaries can intercept shared location samples, on the other hand, is mainly a question of system architecture. If the assumed adversary model considers that operators of centralized systems can become malicious, more decentralized architectures must be considered. By removing centralized data sinks, linking and identification attacks become more challenging; adversaries might be unable to collect relevant samples even if they have been shared by users. Different approaches to decentralization are discussed in Section 2.4.

2.3.3 Anonymity and pseudonymity

Here, approaches for preventing the identifiability of individual location samples are considered, namely anonymity, pseudonymization and obfuscation. Preventing linkability between multiple location samples, which can also enable identification, is discussed in the next section.

2.3.3.1 Anonymity

Anonymity is given if no identifiers of any sort are associated with shared location samples or, if there are such identifiers, no two location samples are associated with the same identifier. An important aspect in this context is the protection of communication addresses. The protection of communication addresses is less challenging when communication is local and radio-based, as the available broadcast domain allows participants to choose and change their own communication addresses at will.

In the same context, communication can be skipped altogether by deploying dedicated infrastructure, e.g., inductive sensors for measuring traffic flow. However, in many scenarios, neither the sole reliance on short-range communication nor the costly deployment of dedicated infrastructure is feasible. Different approaches for enabling non-local anonymous communication are, thus, discussed in Section 2.6.2 of this chapter.

While anonymity is highly desirable from a privacy standpoint, the usage of pseudonymous identifiers is often unavoidable. For example, it is not clear how access control (e.g., for preventing sybil attacks) can be realized without any form of identifiers. Also, if the addressability of participants is required, e.g., for receiving information queries, some form of communication address (which can already be considered a pseudonym) is needed. Lastly, it should be noted again that pseudonymity can be improved into anonymity by ensuring that each pseudonym is used only once.

2.3.3.2 Pseudonymization

Pseudonymization, i.e., the protection of identities through the use of pseudonyms, is a widely used concept in smart traffic and vehicular networking [61, 92, 93]. Pseudonyms used in these contexts are typically composed of communication addresses and cryptographic credentials needed for authenticating and securing communication channels. In order for pseudonyms to be unlinkable to subjects, all identifiers comprising them (i.e., communication addresses and cryptographic credentials) must be unidentifiable as well. Likewise, in order for a pseudonym change to be effective, all identifiers comprising the pseudonym must be changed simultaneously.

Pseudonymization schemes are especially helpful for assisting in the management of abuse. For example, access control mechanisms can be implemented that bound the number of pseudonyms per registered user, thus preventing sybil attacks. Approaches like this, that limit the extent of abuse while maintaining pseudonymity, are discussed in Section 2.5.

2.3.3.3 Obfuscation

The goal of *obfuscation* is to reduce the precision of shared location samples so as to make an identification using additional context knowledge more difficult[6]. The effectiveness of obfuscation against identification attacks has been shown, e.g., in [68] and [50]. However, obfuscation also decreases the utility of location samples [123].

A variety of approaches for realizing location obfuscation have been proposed in the literature [115]. Example techniques include adding artificial noise to reported location measurements and sharing multiple fake dummy locations in addition to the correct ones. Some approaches take additional care to guarantee *k-anonymity*, by

[6] Obfuscation can also be used for reducing the sensitivity of location samples, by making it more difficult to determine a subject's exact location ("hospital or supermarket?"). This property of obfuscation is, for the most part, ignored in the scope of this thesis; all identifiable location samples are considered equally worthy of protection (cf. Section 2.3.1).

ensuring that at least *k* users report the same obfuscated location at a given time. The coordination of participants within this anonymity set is often realized by relying on one or more trusted anonymization providers. Alternatively, the use of local ad hoc networks between participants has been proposed for eliminating the need for such trusted entities [20]. However, since the effective range of ad hoc networks is limited [53], the expected size of obfuscation regions with this approach is small, reducing the benefit of obfuscation.

A more general challenge in the context of location obfuscation is that of *map matching* [115]. Here, context knowledge about the road map defining the movement of participants is assumed. In this way, unlikely locations can be discarded when regarding obfuscated location data. The attack is further aggravated if the speed and movement direction of participants can be inferred as well, which is often hard to avoid given continuous location updates. Other approaches for attacking location obfuscation include forming intersections between multiple shared locations and using additional knowledge about location probability distributions [115].

2.3.4 Unlinkability

Identification attacks are often based on the availability of reconstructed trips, i.e., groups of linkable location samples (cf. Section 2.2.4). For avoiding this attack vector, the *unlinkability* of location samples must be improved. Challenges in this context, that are characteristic to the smart traffic scenario, include that a continuous sampling is often necessary (e.g., for traffic monitoring) and that pseudonyms are often used that cannot be changed for every shared sample (e.g., due to performance constraints).

Concerning the challenge of continuous sampling, it was found that a decrease in sampling rate has significant negative effects on tracking efficiency. According to [117], for anonymous location samples, linking is effectively prevented by using sharing intervals of as little as 16 s. This corresponds to the results by Hoh et al. in [59], where location samples are shared in spatial intervals between 100 m and 500 m. As an orthogonal contributing factor, the use of noisier (i.e., more strongly obfuscated) location data was also found to reduce linking performance in [117]. How far the sampling precision and rate can be decreased in practice without significant cuts in system utility is highly scenario-specific and must be evaluated for each specific case.

If complete anonymity is infeasible, i.e., pseudonyms must be reused for multiple location samples, additional care must be taken concerning the timing of pseudonym changes. More specifically, pseudonym changes must be synchronized with other nearby pseudonym holders so that no unambiguous linking (based on correlating location samples before and after the change) between old and new pseudonyms is possible. In other words, the goal is that a *mixing* of pseudonyms takes place. Mixing can happen in an on-demand, peer-to-peer manner, by leveraging short-range radio communication [71]. Alternatively, the changing of pseudonyms at predefined spots, so called *mix zones* [7], has been proposed. Typically, mix zones are areas in which no communication takes place. After traversing a mix zone, participants start using new pseudonyms. By choosing areas with high degrees of traffic and a low relative speed between participants as mix zones, a high degree of confusion is possible without any additional communication between peers. Example proposals for the placement of mix zones include street crossings [91] and parking lots [74]. Note that independently of the concrete realization, the effects of mixing are amplified by chaining multiple mixing steps, e.g., by passing several mix zones.

2.4 Trust and decentralization

In this section, possibilities for more fine-grained trust modelling are discussed based on different approaches to system architecture and decentralization.

This thesis uses "decentralization" in the sense of distributing functionality across multiple components controlled by non-colluding entities. Therefore, and in contrast to some related works, systems in which key system components are physically distributed (e.g., over multiple machines and data centers) but controlled by a single entity (the service operator) are considered centralized here.

Furthermore, this thesis treats the concepts "trust model" and "adversary model" as mutually inverse. While the adversary model describes which entities can be under adversarial control and what these entities can do, the trust model states assumptions about which entities are trusted to not collude with adversaries, either in general or concerning some set of actions within their capabilities.

In the following, the trust assumptions behind cooperative services are first recapitulated. Following that, different approaches to system architecture are introduced and discussed, namely:

- centralized systems
- peer-to-peer systems
- ad hoc networks
- block chain networks
- mix networks

During all discussions, the main focus is on the implications of the considered approaches for trust modelling.

2.4.1 Cooperative services

Description

Cooperative services are based on the idea that participating users are actively involved in realizing the provided functionality (cf. Section 2.1.3). For example, users can share sensory data for enabling the monitoring of traffic or coordinate plans for optimizing traffic flow. Cooperating in such a way is usually in the interest of all participating users as they benefit from the service provided in the end, e.g., in the form of reaching their destinations quicker.

Trust assumptions

The main assumption behind cooperative services is that, while individual users can be selfish and even malicious, the majority of users will not actively collude against the common interest.

The impact of different percentages of adversary-controlled users within the user population is specific to the individual considered service. For illustration, a simple sensing service can be considered where all participating users contribute one sensory value (e.g., about the average speed on a given road segment) and the median of all values is taken as the final sensing result. If all regular (i.e., non-adversarial) users R always contribute the same correct value c and all adversary-controlled users A always contribute the wrong value w, the end result will be c for $|R| > |A|$, $\frac{c+w}{2}$ for $|R| = |A|$ and w for $|R| < |A|$. Thus, in this simple example, as long as the adversary controls less than half of the user population, he cannot influence the service functionality in any meaningful way.

Note that when considering anonymity and pseudonymity in the context of cooperative services, it must additionally be assumed that the majority of users will not actively collude against specific individual users as well. If users cooperate with an adversary for identifying a specific victim within a system, they cannot be counted towards the anonymity sets for that victim.

Practical challenges

A major challenge in the context of cooperative services is the protection against sybil attacks. If sybil attacks are possible, the percentage of adversary-controlled users can become arbitrarily high. While not immediately causing a higher identifiability of users[7], this can be detrimental to service functionality.

2.4.2 Centralized systems

Description

In the scope of this thesis, systems are considered centralized if one or more crucial[8] system components are under the control of one or few entities (e.g., the same organization). This is a more trust-centric approach to defining centralization and does not have to correspond to physical centralization. For example, a classic client/server system, a cloud-based service distributed over multiple data centers and a nation-wide infrastructure of interconnected base stations managed by one operator are all considered centralized systems here. Many popular smart traffic systems are centralized according to this definition, for example Google Maps [64].

An example centralized system is depicted in Figure 2.1. In the example, all data that is relevant for achieving the system's functionality is collected and stored at a central server.

[7] During an undetected sybil attack, anonymity sets might be perceived as larger than they are in reality. However, anonymity sets will not shrink as the direct cause of a sybil attack - sybil attacks do not directly influence the number of participating regular users.

[8] In the sense that core system functions depend on their availability and correct operation.

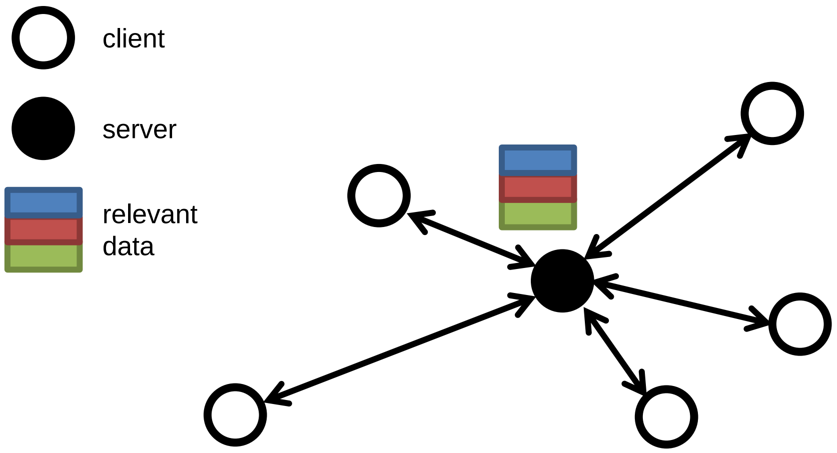


Figure 2.1 A centralized system based on a client/server architecture.

Trust assumptions

Entities controlling crucial components in a centralized system, i.e., service operators, must be trusted for being able to make statements about overall system utility and privacy aspects. This is a clear disadvantage, considering the potential vulnerability of individual service operators to coordinated attacks, bribes, legal subpoenas and abuse within their own organizational structures. In some cases, centralized systems can be designed in such a way that some privacy protection goals are met even under the assumption of a malicious service operator. However, the availability and functionality of a service are inherently dependent on the service operator's integrity. Also, the undetectability and confidentiality of data that is critical for providing a service's functionality (e.g., location samples when considering a traffic monitoring system) is usually unachievable in centralized systems without the introduction of additional communication channels.

Practical challenges

Practical disadvantages of centralized systems include the difficulty of financing the continuous maintenance of central system components [97]. As a result of this difficulty, centralized systems are often deployed by for-profit companies that might not have a sufficient interest in designing their services in a privacy-conscious manner.

2.4.3 Peer-to-peer systems

Description

Peer-to-peer systems are, in essence, the application of the cooperative services paradigm to system architecture. All functionality provided in a peer-to-peer system is realized directly by individual participating entities, i.e., *peers*. Peer-to-peer systems enable decentralization; peers can be autonomous and no centralized control is necessary.

Distributed hash tables (DHTs) like the widely used *Kademlia* [78] are a classic example for peer-to-peer systems. As their name suggests, DHTs realize key-value storages where the values are stored on different peers. DHT protocols define mechanisms for maintaining an *overlay network* structure layered on top of an existing communication substrate like the Internet. In this context, peers are usually also referred to as overlay *nodes*. Rules are defined for, amongst other things, forming and maintaining *neighbor* relationships with other peers. The resulting overlay topology is typically optimized for allowing the efficient routing based on keys and node identifiers. Overlay networks have also been proposed in the context of smart traffic. In [101], for example, a decentralized traffic monitoring system - *PeerTIS* - is realized with the help of a scenario-tailored DHT.

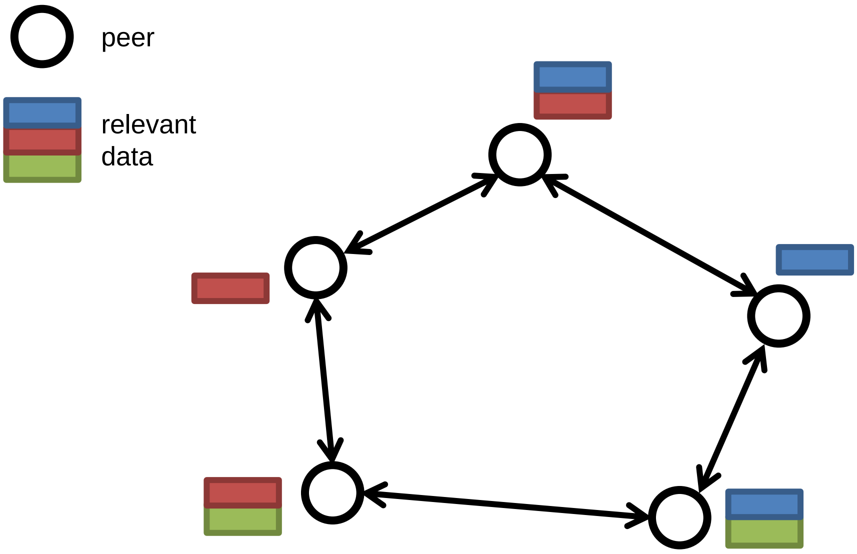


Figure 2.2 A peer-to-peer system realizing a DHT.

An example peer-to-peer system realizing a DHT is depicted in Figure 2.2. Data stored in the DHT is replicated across multiple peers.

Trust assumptions

Similarly to cooperative services, the trust assumption underlying DHT-like peer-to-peer systems is that the majority of peers does not collude with malicious intent.

Concrete harms are, again, specific to the individual systems. A simple example system will be considered for illustration. Consider a DHT in which a key lookup involves the querying of k distinct peers. With p_R denoting the probability that a given random peer does not collude with the adversary (i.e., p_R is the ratio of regular peers), the probability p_t that an adversary will be able to tamper with the lookup is $p_t = 1 - {p_R}^k$. With this probability, the adversary can either block the lookup or change its result.

Practical challenges

As in cooperative services in general, DHTs and similar peer-to-peer systems are heavily vulnerable to sybil attacks[9]. While sybil attacks are a general challenge in

[9] Approaches for preventing sybil attacks and lessening their potential impact are discussed in Section 2.5.1.

the context of peer-to-peer systems, some approaches discussed in the following sections, most notably the block chain paradigm, effectively manage this threat as a part of their design.

Most existing peer-to-peer systems are not primarily concerned with the meeting of privacy protection goals. The PeerTIS system [101] mentioned earlier, for example, enables arbitrary peers to collect precise location samples.

2.4.4 Ad hoc networks

Ad hoc networks are, in essence, peer-to-peer systems based on short-range radio communication between peers. *Vehicular ad hoc networks* are the application of this approach to the scenario of vehicular traffic. Vehicular ad hoc networks have been widely discussed in the research community. A comprehensive overview of the field is given, for example, in [35] and [66].

Vehicular ad hoc networks are not at the focus of this thesis. Reasons include their limited geographic scope. The maximum transmission range of common vehicular radio technologies is limited. This is even more the case in densely populated areas with a large number of physical obstructions, e.g., due to buildings. Additionally, it was found that even using multi-hop routing, the scalability of mobile ad hoc networks remains limited [53].

2.4.5 Block chain networks

Description

Block chain networks are a form of peer-to-peer systems in which a single data structure is maintained and synchronized across all peers - the so called *block chain*. First proposed in the context of *Bitcoin* [84, 110], the block chain paradigm is especially interesting in the context of eliminating the need for singular trust anchors while maintaining robustness against sybil attacks.

A cooperatively maintained block chain realizes a distributed append-only bulletin board with unified global state, as well as a timestamping service. Its high resilience to sybil attacks stems from the fact that the level of influence of individual peers is directly tied to the possession of limited resources like, e.g., computing power. In a Bitcoin-type network, for example, an adversary must control more computing resources than the remainder of the network in order to cause significant disturbances. In addition to the original application of realizing a decentralized payment system, the block chain paradigm has been adapted for realizing naming systems[10], adding incentives to sensing services [89], timestamping cryptographic commitments [24], securing username registration [45], bootstrapping trust without a TTP [54] and realizing decentralized anonymous credentials [47], to name just a few examples.

An example block chain network is depicted in Figure 2.3. In addition to feature-complete peers, also referred to as *full nodes*, the figure features a *simple payments verification* (SPV) client that doesn't store the complete block chain. In popular block

[10] `https://namecoin.info`

chain networks like Bitcoin, full nodes face significant communication and storage requirements[11]. Addressing this challenge, SPV clients have been proposed as early as in the original Bitcoin paper [84]. SPV clients need to store only metadata and can query full nodes for additional data on demand.

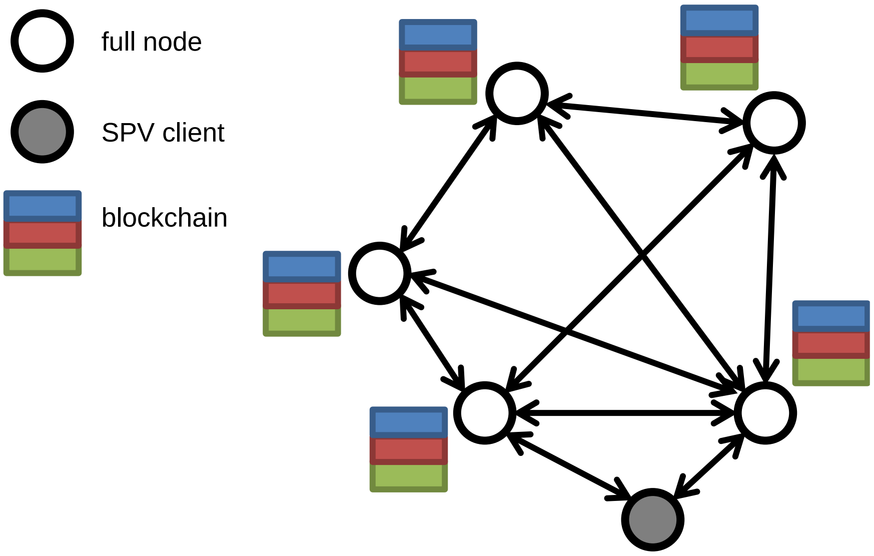


Figure 2.3 A block chain network.

An in-depth description of Bitcoin and similar block chain-based systems exceeds the scope of this chapter. In Chapter 6, a system is proposed that builds upon the block chain paradigm and can be implemented on top of the Bitcoin network. Relevant technical details related to Bitcoin are introduced there.

Trust assumptions

In contrast to DHT-like peer-to-peer systems, the main trust assumption underlying block chain networks is not that the majority of peers do not collude maliciously, but that all entities colluding with malicious intent do not own more than 50% of the relevant resources in the network. So, for the population of regular peers R, the population of adversary-controlled peers A, and with $\text{res}(x)$ denoting the relevant resources held by the peer x, a block chain network is considered secure as long as:

$$\sum_{r \in R} \text{res}(r) \geq \sum_{a \in A} \text{res}(a)$$

As long as this assumption holds, adversaries can neither alter the current state of the block chain nor prevent the inclusion of new entries. They can, however, delay the inclusion of new entries by a factor proportional to the amount of resources held by them (up to a factor of 2). Independently of the amount of resources an adversary controls, he can never add entries to the block chain that are considered invalid according to the used block chain protocol. For example, in Bitcoin, the stealing of

[11] At the time of writing, the Bitcoin block chain (which must be stored by all full nodes and downloaded upon joining the network for the first time) had a size of just slightly under 50 GB.

currency units from a specific Bitcoin address is impossible without knowledge of a matching private key. However, an adversary controlling more than half of the resources in the network might be able to alter past inclusions in the block chain. In the context of Bitcoin, this can be used for performing *double-spending attacks*, i.e., reverting payments that were previously marked as finalized. Such history rewriting attacks are, however, highly time-intensive for entries past a certain age, as well as difficult to hide. An easier attack for adversaries controlling the majority of resources in the network is preventing the inclusion of certain new entries in the block chain, effectively censoring affected users.

Concerning the security of SPV clients, transaction data received from full nodes is cryptographically verified so that only few deception possibilities remain. Malicious full nodes can withhold data from connected SPV clients or attempt to present an alternative chain of entries. Due to the inherent logic behind the block chain paradigm and the SPV approach, the latter attack vector quickly becomes prohibitively expensive. Additionally, both threats can be managed by connecting to multiple full nodes and comparing the data received from them.

All information stored on the block chain is public. Consequently, value transfers in block chain-based payment systems are realized through a publicly viewable ledger. While value transfers are pseudonymous, transactions are linkable to each other by design and the flow of currency units between pseudonymous addresses is openly visible. A variety of approaches have been proposed for improving the privacy properties of block chain networks in this respect and enabling transaction unlinkability. For example, the usage of accountable external mixing providers [9] and new transaction types involving zero-knowledge proofs [6, 79] have been proposed. Several solutions based on the cooperation of users (e.g., for forming joint mix transactions) have been proposed as well [8, 77, 100, 124].

Practical challenges

Popular block chain networks like Bitcoin are inherently energy-inefficient. Proofs of controlled computing power, realized through the solving of cryptographic puzzles, are continuously required for adding new block chain entries. On a global scale, this leads to adverse environmental impacts [5]. More energy-efficient alternatives are being developed. A notable example is the *proof-of-stake* approach used in the *PPCoin* cryptocurrency [67]. Here, the level of influence of individual peers is tied to the portion of the currency supply they control. Without an active exchange of currency units, this is largely based on the time they have already spent participating in the system.

2.4.6 Mix networks

Description

Mix networks are networks of non-colluding entities that are most commonly used in the context of anonymous communication. The main idea is that messages are sent through a chain of randomly chosen nodes from the mix network, so called *mixes*. Mixes act as proxies and forward messages to subsequent mixes or the actual destination. By using multiple layers of encryption, *decryption mix nets* can be realized in which every mix is aware only of its predecessor and successor in the chain and only the communication endpoints are aware of message contents. In this way, as long as at least one of the mixes does not collude with the remaining ones, the anonymity of the communication is effectively preserved against all participating entities but the message source. If sufficient amounts of traffic are present, i.e., if multiple users are sending messages through the mix network at the same time, a deanonymization of communication processes also becomes difficult for external adversaries that can observe all or parts of the traffic in the network. In the context of this thesis, the most relevant instantiation of the mix network paradigm is the anonymous communication system *Tor* [31]. Tor offers low-latency two-way anonymous communication through a network of independently operated mixing servers, so called *relays*. The characteristics and service-related properties of the Tor network are further discussed in Section 2.6.2.

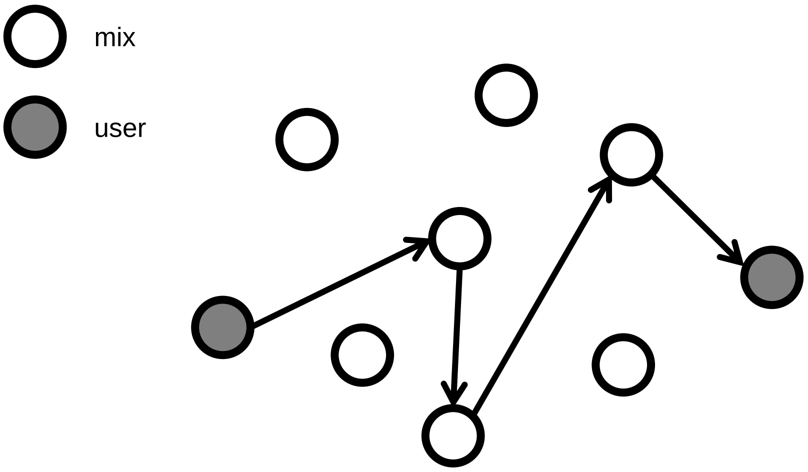


Figure 2.4 Communication through a mix network.

Figure 2.4 depicts an example mix network with two users. One of the users is communicating with the other via three randomly chosen mixes.

Trust assumptions

Since mixes are usually chosen randomly by the user when communicating through a mix network, it can never be guaranteed with certainty that an effective anonymization will be realized. However, setting aside any possible linkability based on external traffic observations and timing, an effective anonymization is possible as long as at least one of the randomly chosen mixes does not collude with the remaining ones. Thus, if p_A denotes the probability that a randomly chosen mix is controlled by the adversary and k is the used mix chain length (the default used by Tor is $k = 3$), the probability that the anonymization was successful becomes $1 - {p_A}^k$. Since this probability is heavily influenced by the number of mixes an adversary can control (assuming an uniformly random selection process, p_A is the ratio of adversary-controlled mixes in the network), effective protection mechanisms against sybil attacks are needed for maintaining the anonymization utility of the mix network. Also, note that attacking the functionality of the service, e.g., by not forwarding a message to next mix, is still possible even if only one mix in the chain is controlled by the adversary. However, such attacks are easily spotted and the responsible mixes can be excluded from future chains.

Practical challenges

A different approach for deanonymizing communications through a mix network is by observing the flow of traffic between mixes. Especially in low-latency networks, this approach can be highly effective [82]. The attack can be made more difficult for an adversary by increasing communication latencies as well as the number of simultaneously active users, the anonymity set. Naturally, however, active users must be non-colluding with the adversary in order to be counted towards anonymity sets.

2.5 Limiting abuse

The impact of abuse, i.e., attacks on the correct functioning of services, can be significant in the context of smart traffic. For example, traffic monitoring can be influenced by reporting false data, ultimately leading to an alteration of vehicular traffic [64]. This section introduces and discusses different approaches for managing abuse. Two different aspects are considered:

1. *Preventing sybil attacks*, as they amplify the significance of many types of attacks.

2. *Reacting to abuse*, e.g., by implementing punishment mechanisms for malicious behavior.

2.5.1 Preventing sybil attacks

In a *sybil attack* [33], a single adversary creates a large number of pseudonymous identities (sybils) and uses them simultaneously. In this way, he gains the potential to heavily disturb cooperative services. Sybil populations can not only reduce the utility of a system, e.g., by deliberately contributing disproportionate amounts of falsified data, but also weaken privacy protection mechanisms, e.g., by crowding out regular users from anonymity sets. The possibility for sybil attacks undermines the main trust assumption behind cooperative services and many approaches to decentralized system design - that the majority of participating identities are non-colluding.

A variety of defense mechanisms against sybil attacks have been proposed in the literature, mostly in the context of peer-to-peer networking [111]. Some require the existence of a *trusted third party* (TTP), a single entity trusted by all participants, while others can be realized in a decentralized manner. In the following, existing approaches are discussed based on this classification, starting with approaches relying on a TTP.

2.5.1.1 Approaches based on the existence of a TTP

Using a TTP like a certificate authority or pseudonym issuer is a straightforward solution to the sybil challenge. In essence, the TTP issues pseudonyms to users and keeps track of the number of identities issued per user. TTP-based solutions are widely used. Even systems aimed at high degrees of decentralization (e.g., the decentralized online social network *Safebook* [25] and the peer-to-peer signaling protocol *RELOAD* [63]) include TTPs in their design for bounding the number of identities per user.

In order to ensure that only the TTP can issue valid identities, pseudonyms must contain some sort of cryptographic proof like a cryptographic signature from the TTP. However, regular cryptographic primitives like asymmetric signatures enable the TTP to learn the pseudonyms issued to individual users and identify all pseudonymized actions and communication. A common solution for avoiding this identifiability by the pseudonym issuer is the usage of *blind signature* schemes [16]. When requesting a blind signature on some data, a user first applies a *blinding* operation on the data. The signing authority calculates a signature on the blinded data without knowledge of the original data. The signature on the blinded data can be *unblinded* by the user, yielding a valid signature on the original data. After this process, the signing authority is unable to link the act of signing (and, hence, the identity of the user) with the resulting signature used for authentication. Blind signatures (including a possible implementation) are further discussed in Chapter 4 of this thesis, where they are used for enabling privacy-preserving and abuse-resistant cooperative planning.

A simple practical realization of pseudonym issuing using blind signatures is described in [60]. As an alternative to the usage of blind signatures, pseudonymization schemes have also been proposed based on the use of zero-knowledge proofs [14]. In [47], Garman et al. propose an approach for realizing decentralized anonymous credentials using zero-knowledge proofs and block chain networks. While it likely supports anonymizing credentials determined without a TTP, the approach inherently requires a TTP for initially setting up system parameters.

In systems where users are active for extended periods of time, the changing of pseudonyms needs to be supported as well for reducing the linkability between actions and shared samples. The issuing of pseudonym pools to participants (as proposed, e.g., in [36]) as well as the use of self-certifying sybil-free e-token dispensers [13] have been proposed in this context for allowing pseudonym changes without introducing a possibility for creating sybil identities.

The central drawback of all mechanisms discussed in this section is that, while TTPs must not necessarily be trusted for guaranteeing the non-identifiability of pseudonyms, they still retain the ability to mount sybil attacks. This situation is problematic when considering strong adversaries that can compromise individual system components and cryptographic keys (e.g., by colluding with a service operator). Furthermore, the TTP becomes inherently required for the correct functioning of the system, making it a single point of failure and attractive attack target. Securing TTPs against both compromise and attacks on their availability is costly (as is reflected, e.g., by the fees charged by certificate authorities in practice). Additionally, appointing TTPs is non-trivial in systems of global scope, as all participants need to agree on a common trust anchor.

2.5.1.2 Decentralized approaches

Decentralized approaches for preventing sybil attacks are often based on the assumption that adversaries cannot control an arbitrarily larger amount of a specific scarce resource than regular participants [33]. Consequently, a large number of works describe mechanisms that enable participants to *prove* their possession of a given resource. Commonly used resources include:

- *Communication addresses.* In [30], for example, new participants are required to register their IP address in a DHT which can later be queried for identifying sybils.
- *Computing power.* So called *proof-of-work* schemes are usually based on the solving of cryptographic puzzles that require a certain amount of computing time [4, 10].
- *Human attention. CAPTCHAs* [113] are commonly used for ensuring that a human subject was involved in an operation. These are human-solvable challenges for which no reliable automated solution is known.

For effectively limiting sybil attacks, resource proofs must be requested regularly from each participant. This requirement leads to a central drawback of proof-of-resource schemes - the involved challenges are either too difficult, thus deterring honest users (e.g., when considering mobile devices with limited resources or the user experience from solving countless CAPTCHAs), or too easy, thus offering insufficient protection against determined adversaries (e.g., IP address blocks and computing power can be rented from cloud providers, crowdsourcing marketplaces can be used for solving CAPTCHAs on a large scale) [69].

A related approach, *proof-of-burn*, is based on the idea that a non-replenishable resource needs to be provably destroyed, e.g., when a new identity is established. In contrast to proof-of-work, a proof-of-burn can be completed instantly and adversaries are unable to leverage economy of scale effects: every proof "costs" the same for everybody. The approach was proposed in the context of Bitcoin [84] for establishing trustworthy internet pseudonyms by provably destroying a certain amount of Bitcoin [54].

Protection mechanisms based on the social graph between participating users have been proposed for peer-to-peer networks [27, 121, 122]. Such schemes are based on the assumption that sybil node populations do not exhibit the same interconnection patterns found in regular social graphs [112]. It can be argued that, compared to controlling communication addresses, computing resources or human attention, the forming and maintenance of social links with human users is significantly more difficult for an adversary and does not scale as well. In many social graph-based approaches, the social relationships between users must be made public. From a privacy standpoint, this can be highly problematic [86]. Peer-to-peer systems like X-Vine [81] and Whanau [70] follow a different approach and integrate social graph-based sybil protection directly into their overlay network structure.

In some systems, a small group of entities is responsible for detecting and preventing sybil attacks. In the Tor network, for example, 5-10 independent *directory authorities* constantly monitor the network and, based on voting, blacklist nodes suspected of malicious behavior or of being part of a sybil population. This approach replaces a singular TTP with a set of third parties, the majority of which need to be trusted to not collude with malicious intent.

Effective and easy to deploy approaches exist for preventing sybil attacks in systems based on short-range communication. In addition to the difficulty of multiplexing a radio interface across multiple identities, different proposals exist for identifying whether two claimed identities reachable over short-range radio share the same position (making them likely members of a sybil population). To name a few examples, techniques from this class have been proposed using radio-based position verification [49] and statistic signal strength distribution analysis [120].

2.5.2 Reacting to abuse

In the scope of this section, it is assumed that some possibility for abuse exists and that an adversary is willing to exploit it. General possibilities for the *detection of abuse* are considered first, followed by approaches for privacy-conscious *blacklisting and reputation management*.

2.5.2.1 Detection

In order for users that abuse the system to be identified and, for example, punished, their malicious behavior must first be detected. Meeting this precondition is, again, a scenario-specific challenge. Here, a few noteworthy examples relevant to smart traffic are discussed.

A general approach for identifying malicious users is by comparing their actions with those of regular users. A simple example for this approach can be constructed for cooperative sensing systems where multiple users contribute sensor values concerning the same object (e.g., for determining the road quality at some location). Such a system can be abused by contributing false sensor values. However, if the cooperative services assumption holds and the majority of contributing users do not collude with the adversary, false values are detectable through their distance to the median. The sources of such contributions can be marked - with high probability, they are either malicious or their sensing equipment is malfunctioning.

When disturbing smart traffic systems, malicious entities must often also falsify their own location. Through such *location spoofing*, an adversary can, e.g., contribute false traffic reports about some area (to influence traffic monitoring results) without needing to be physically present [64]. Solutions for the protection against location spoofing exist that require additional infrastructure support [15] or spot checks by local trusted entities [96]. In Chapter 5, an alternative approach is investigated that is based on a reduction of the location verification challenge to the challenge of verifying an entity's proximity.

2.5.2.2 Blacklisting and reputation management

If anonymity or pseudonymity is given, it becomes highly challenging to punish users upon detected misbehavior. For example, malicious users can simply switch to a new pseudonymous identity. In this context, several anonymous blacklisting systems have been proposed that allow service operators to serve anonymous users while retaining the possibility to blacklist them [57]. As a downside, these approaches are primarily focused on centralized setups in which users authenticate to one or more service providers. Additionally, the existence of a TTP is usually assumed.

Reputation systems have been proposed for allowing a more fine-granular distinction between cooperative and malicious nodes, e.g., in the context of peer-to-peer systems [65]. Several works have also attempted to reconcile reputation systems with the requirement of participant anonymity, resulting in the proposal of several anonymous reputation systems. Androulaki et al., for example, propose an approach centered around the idea of *repcoins* [1] (based on blind signatures and David Chaum's original electronic cash idea [16]). The *IncogniSense* framework [22] enables reputation transfers between pseudonymous identities that preserve the unlinkability between pseudonyms. The approach is also based on the usage of blind signatures. Both IncogniSense and the repcoins approach, however, rely on the existence of a TTP for securing reputation scores.

2.6 Related challenges

In the following, two challenges are discussed that are related to the main topics of this thesis. The connectivity and privacy characteristics of contemporary cellular networks are discussed first. Following that, practical solutions for securing both the content and the metadata of non-local communication are considered.

2.6.1 Connectivity and privacy in cellular networks

For scenarios in which long-distance connectivity is required for highly mobile entities (such as moving vehicles), no practical alternatives to the use of cellular networks are known. However, the reliance on cellular network connectivity introduces a number of new challenges and open questions.

For one, in currently deployed cellular network architectures, service operators can both identify and locate all active clients. Consequently, they need to be trusted for maintaining their user's location privacy in all services relying on cellular network connectivity. This is an assumption shared by virtually all works from the smart traffic domain that rely on cellular network services. While greatly exceeding the scope of this thesis, designing cellular networks in a privacy-conscious and trustless manner does not represent a technical impossibility. Data minimization techniques like pseudonymization can be applied within cellular networks just as within other systems. Concrete proposals in such a direction have already started to emerge in the research literature. With *ZipPhone* [107], for example, users connect to the network anonymously by sharing a pool of virtual SIM cards with other participants.

In addition to these general privacy concerns, a number of more deployment-specific characteristics are also interesting in the context of this thesis, namely:

1. Whether cellular networks are ubiquitously available for smart traffic users.
2. Whether the establishment of peer-to-peer connections to other cellular network users is possible.
3. Whether IP address changes can be initiated by users (for ensuring the unlinkability of pseudonym changes).
4. Whether IP addresses used in cellular networks are identifiable by external adversaries.

It can be argued that 1 is easily true for cities and major roads in the developed world. For determining the validity of 2, 3 and 4, the connectivity properties of all cellular networks available in Germany were evaluated. The setup and detailed results of this evaluation are discussed in Appendix A. In the following, a summary of the results is presented:

2. The establishment of peer-to-peer connections is not always possible. Restrictive middleboxes and filtering policies were deployed in three of the four considered networks. For clients of two of the four considered networks, peer-to-peer connections were completely infeasible.
3. IP address changes can be enforced by users.
4. New IP addresses still belong to the subnets of the same cellular network operator. This increases the linkability between pseudonym changes and might eventually lead to an identification of users.

2.6.2 Private and secure communication

This section considers the establishment of communication channels that are suitable for the use in cooperative services and decentralized systems. For maintaining privacy protection and robustness in cooperative services, several privacy and security-related protection goals should be fulfilled. From a privacy standpoint, the *anonymity*, *undetectability* and *confidentiality* of communication is especially relevant (see Section 2.2.1 for a detailed explanation of these protection goals). For limiting various possibilities for system compromise and abuse, the security goals of *integrity* and *authenticity* should additionally be realizable. Integrity is given if the content of the communication cannot be altered by adversaries without being noticed. Authenticity is given if impersonation attacks are prevented and the (possibly pseudonymous) identity of the communication partners can be verified. Note that this does not necessarily imply that the communication endpoints are identifiable in the sense that links to subjects can be inferred (which would conflict with the anonymity goal). If pseudonyms are used, for example, authenticity is also given if the communication partner is provably the rightful holder of the claimed pseudonym.

In the following, possible approaches for realizing these protection goals in both local and non-local (e.g., IP-based) communication are discussed, with an additional focus on the anonymization service Tor.

2.6.2.1 Local communication

For adversaries that are not locally present, all of these protection goals are straightforwardly met by communication using short-range radio. If radio is used and adversaries are within receiving range, the undetectability of communication cannot be ensured. The anonymity of communication is only possible as long as the adversary cannot localize the transmitting stations and link them to subjects via, e.g., physical observations. The remaining protection goals can be met using state of the art security mechanisms from the vehicular networking domain [61]. These commonly rely on asymmetric cryptography for bootstrapping an authenticated channel, using cryptographic certificates included in pseudonyms.

2.6.2.2 Non-local communication

Concerning the security of non-local communication (e.g., over the Internet), various secure communication protocols that offer confidentiality, integrity and authenticity have been proposed in the research literature and technical standards. The *transport layer security* (TLS) protocol [29] is arguably one of the best studied and most widely used of these proposals. Again, the authenticity of communication partners can be verified using asymmetric cryptography and the presentation of cryptographic certificates. Upon bootstrapping a communication channel, symmetric cryptography is used, for performance reasons, for ensuring confidentiality, integrity and authenticity.

For protecting the metadata of IP-based communication (achieving anonymity and ideally also undetectability), several approaches exist that are of practical relevance. Proxy servers and virtual private networks are popular amongst end users. However, they rely on the assumption that the chosen operators are honest and do not collude with potential adversaries. By chaining multiple services, the trust requirements can be reduced. In essence, mix networks (cf. Section 2.4.6) realize just this - a chaining of multiple potentially malicious anonymization providers so that the likelihood that all entities in a chain are colluding is reduced. The Tor network [31] is an especially notable instantiation of the mix network approach that is discussed separately in the following. Other systems based on mix networks, e.g., *HORNET* [18], are not deepened upon due to their lower availability and user base (and hence achievable anonymity sets) at the time of writing.

2.6.2.3 Tor an Peer-Tor-Peer

Tor [31] is an infrastructure-based mix network, i.e., anonymization is provided by a network of dedicated relays (often high-bandwidth servers) that are reachable from the public Internet (i.e., are not firewalled or behind NAT). Originally focused on enabling anonymous web surfing, Tor is also low-latency, in contrast to high-latency mix networks where messages are delayed to achieve a higher resilience to timing-based correlation attacks. At the time of writing (May 2016), the Tor network was comprised of more than 7000 relays distributed over 1273 different autonomous systems and 82 countries[12]. The likelihood that all relays in a chain are malicious and colluding is, therefore, lower than in any other practically deployed mix network. The Tor network is currently estimated to serve more than 1 700 000 concurrent users per day[13]. This number is, to the best of the author's knowledge, higher than for any other anonymous communication service deployed today. Note that the number of users of an anonymization service directly limits the achievable anonymity level of its users (i.e., the size of achievable anonymity sets).

In addition to sender anonymity, Tor also supports the registration of *hidden services* providing receiver anonymity. Using hidden services, users can become reachable via the Tor network without disclosing their true IP address. Cryptographically generated ".onion"-addresses are used for addressing Tor hidden services. For registering hidden services and accepting connections, only outbound connections must be made by clients. Thus, using hidden services, peers are able to connect to each other independently of any restrictive firewalls or NAT middleboxes, as long as outbound connections to Tor relays are possible. As was discussed in Section 2.6.1, this is highly relevant in currently deployed cellular networks where the establishment of direct communication links between users is often impossible.

In the scope of this thesis, an abstraction library for Tor and Tor hidden services was developed. Named *Peer-Tor-Peer* (PTP)[14], the library is based on the Java language and aims at enabling the rapid development of Android applications that require the establishment of peer-to-peer connections. In addition to enabling peer-to-peer connectivity even in the presence of restrictive middleboxes, PTP (through the use of Tor and Tor hidden services) also realizes the pseudonymity, undetectability, confidentiality, integrity and authenticity of resulting communication channels. In the context of this thesis, PTP can be used for building practically usable implementations of OverDrive (cf. Chapter 5) and the mixing protocol used in BitNym (cf. Chapter 6). While not explicitly evaluated in scenarios with high mobility, PTP is expected to provide continuous connectivity as long as cellular network connectivity remains present and forced IP address changes are rare[15].

[12] `https://compass.torproject.org`

[13] `https://metrics.torproject.org/`

[14] `https://github.com/kit-tm/PTP`

[15] IP address changes necessitate the reregistration of hidden service identifiers, leading to delays during which the reachability of PTP peers is impaired.

3. Assumptions, adversary model and evaluation approach

A number of *assumptions* underlying the works presented in this thesis are stated in Section 3.1 of this chapter. A general *adversary model* is defined in Section 3.2. The model defines the combination of influence, goals and knowledge against which all solutions proposed in this thesis strive to protect users. A structured *analysis and evaluation approach* used in the remainder of this thesis is introduced in Section 3.3.

3.1 Assumptions

This thesis is based on a number of general assumptions that are introduced in the following. Whenever works presented in subsequent chapters make use of additional, more specific assumptions, these are presented in those respective chapters.

3.1.1 Users

In the scope of this thesis, extensive use of the term *user* is made. For simplicity, it is used in a very broad manner so as to include both the driver of a smart vehicle and any intelligence within the vehicle (e.g., in the form of on-board devices like navigation systems) acting in the interest of that subject[1]. For the sake of simplicity, one subject is assumed per vehicle that controls the vehicle and its movement[2].

In the following, several assumptions are made concerning the capabilities of users. While formulated with current and near-future smart cars in mind, the required

[1] The fact that subjects can fulfill multiple roles in the context of smart traffic, e.g., can also be passive passengers, is abstracted away.

[2] Considering the near-future possibility of widely available autonomic vehicles, not necessarily by actually driving.

assumptions can also be met by commonly available smartphones (if only vehicles with more conservative capabilities are available). Following assumptions are made:

- Cellular networking interfaces are available and cellular network coverage is ubiquitous. Sufficient bandwidth capacities are present to service large numbers of users independently of their geographic distribution.
- Short-range communication interfaces, e.g., dedicated inter-vehicle communication interfaces [66] or simply Bluetooth [12], are available[3].
- Users are always able to determine their own location, e.g., using GPS. Additionally:
  - No privacy threats are associated with localization. For example, no enhanced localization services involving the disclosure of location information to a cloud service (as investigated in [64]) are used.
  - The localization precision is sufficient for all use-cases considered in this thesis.
- Users have an unlimited energy budget[4].
- The majority of users are honest and cooperate willingly. While possibly selfish or curious in isolation, the majority of users will never actively collude with malicious intent[5].

3.1.2 Privacy in cellular networks

Cellular network operators are assumed to honor the location privacy of their users and never collude with adversaries to release collected location data. With currently deployed cellular network architectures, this assumption is required in any system relying on cellular network connectivity. This includes systems proposed in related works (e.g., [59]) as well as commercially deployed systems like Google Maps.

Location privacy in cellular networks is a problem orthogonal to the challenges tackled in this thesis. However, it does not represent a technical impossibility and research in that direction is available [107]. Given cellular network technologies (or similar long-range wireless communication approaches) that effectively prevent operators from obtaining identifiable location samples, the assumption of honest cellular network operators is no longer necessary.

[3] This assumption is required only for the abuse-resistant cooperative route planning system proposed in Chapter 4.

[4] Users of smart traffic services are typically associated with moving vehicles, e.g., cars. For this reason, this thesis assumes the existence of an abundant energy budget and does not specifically consider the energy consumption of the presented solutions.

[5] This assumption is central to the cooperative services concept (cf. Section 2.4.1). It also implies that the majority of users will not actively collude to identify specific individual users or their actions.

3.1.3 Existing building blocks

A number of existing cryptographic building blocks and privacy-enhancing technologies are used in this thesis. On the cryptography side, this includes symmetric encryption (e.g., via AES), digital signatures and asymmetric encryption (e.g., via RSA and elliptic curve cryptography), secure communication protocols like TLS [29] and blind signature schemes like RSA blind signatures [17]. The security of all of these building blocks is assumed. Note that specific implementations of cryptographic primitives are freely exchangeable in practice as long the provided functionality is preserved. On the user side, it is assumed that users, unless controlled by an adversary, are able to generate secure cryptographic keys and maintain their secrecy if necessary.

In addition to existing cryptographic primitives, several systems deployed in practice are assumed to be secure within their individual trust model. In addition to the Bitcoin network, which is of interest mainly in Chapter 6, this includes the anonymous communication services Tor and PTP (cf. Section 2.6.2). Specifically, it is assumed that Tor effectively anonymizes communication for any adversary considered in this thesis. Likewise, it is assumed that PTP successfully leverages functionality provided by Tor for enabling authenticated and both content- and metadata-protected connections between pseudonymous peers. These assumptions imply that no timing attacks based on global scale communication eavesdropping will be considered and that the majority of Tor relays are assumed to be non-colluding. Connections to the Tor network, and, consequently, interactions using PTP, are assumed to be possible for all entities with Internet access (e.g., via cellular networks).

3.2 General adversary model

An adversary model is defined in the following for enabling the analysis and evaluation of privacy characteristics. The main goal of the adversary is the collection of identifiable location samples from users (i.e., of timestamped locations than can be linked to subject identities). In general, the presented model applies to all systems discussed in this thesis. Scenario-specific concretizations are made where necessary.

The adversary's area of influence, goals and assumed inference capabilities are defined first. Following that, concretizations of the adversary model for the different scenarios investigated in this thesis are listed.

3.2.1 Capabilities

It is assumed that the adversary can infiltrate individual system entities (e.g., users, service operators) for achieving his goals. Entities controlled by the adversary are not required to adhere to any predefined rules or specifications, e.g., can ignore protocol steps and distribute false information.

As an important distinction to many related works, if single entities (e.g., service operators) control indispensable system components (as is the case in centralized systems), these entities are explicitly assumed to collude with the adversary[6]. This

[6] Cellular network operators are an exception here (cf. Section 3.1.2).

assumption reflects the threat arising from security breaches, governmental subpoenas and dishonest service operators.

Concerning the infiltration of the user population, it is assumed that the majority of users does not collude with the adversary. Weakening this assumption would question the entire cooperative services concept. A bounded number of users can, however, be infiltrated by the adversary, e.g., via bribes or security compromise. The exact percentage of the user population that is under adversary control is varied as an evaluation parameter.

Compared to the Dolev-Yao model [32], the adversary assumed in this thesis is not globally omnipresent[7]. While he can intercept individual messages, he is not able to deanonymize anonymous low-latency communication (e.g., communication over Tor) using timing-based attacks. Note that anonymous communication technologies like Tor and PTP (cf. Section 2.6.2) as well as secure communication protocols like TLS [29] are used throughout this thesis. Under the assumption that these building blocks are not vulnerable to the adversary and provide their services correctly, the possibility for malicious interception, injection or altering of messages does not pose a significant threat and is therefore ignored.

The assumed adversary is not able to deploy additional infrastructure that is not already required by the considered system. For example, an adversary deploying dedicated vehicle tracking equipment (e.g., license plate scanners) with the explicit goal of tracking users is not covered by the adversary model described here.

3.2.2 Adversary goals

The primary goal of the adversary is the collection of identifiable location samples from users. Depending on the adversary's priorities, two flavors of the goal are possible. In the simplest case, the adversary tries to collect as much samples as possible from as many users as possible. Alternatively, he targets one specific user and attempts to track his movement. If nothing else is specified, the former behavior is assumed.

The adversary also pursues the realization of other privacy threats that can lead up to the obtaining of identifiable location samples. In the scope of this thesis, this most notably includes the identification, or breaking, of pseudonyms.

Apart from these adversarial goals, following constraint is also assumed for the adversary: if he colludes with entities of central importance to a system, e.g., with a service operator that performs key tasks, it becomes an additional goal for the adversary to maintain the utility of the system. This goal models the fact that he might want to escape detection and maintain the secrecy of his surveillance, ensuring that users won't stop using the compromised service. In the case that the adversary's identity is identical to that of the service operator, maintaining the system might also simply be more important for him than compromising the privacy of users. As a direct result of the described constraint, the adversary model developed here is

[7] It should be noted that adversaries in the Dolev-Yao model are not able to infiltrate participating entities, which is a major characteristic of the adversary outlined here.

unsuited for evaluating the abuse resistance of a system. Separate, scenario-tailored adversary models for this goal are introduced where necessary.

3.2.3 Context knowledge and identification attacks

Several identification attacks on location samples have been proposed in the literature, based on correlation, timing and additional context knowledge (cf. Section 2.2.4). The evaluation of well-known attacks is not a focus of this thesis. It is assumed that attacks from the literature are possible for the adversary if he has access to a comparable set of data. If additional context knowledge is required for a considered attack (e.g., the home and work locations of users as in [49]), the adversary is assumed to have access to it.

3.2.4 Concretizations for different systems

Concretizations to the presented adversary model are made for each of the specific scenarios tackled in this thesis. The more important of these adaptations are listed in the following.

Concerning the *promise coin* approach to privacy-preserving cooperative route planning, a centralized system architecture is given. Consequently, according to the defined adversary model, the service operator with all of its infrastructure is assumed to be under the full control of the adversary. The adversary has an interest in maintaining the utility of the system, i.e., ensuring an improved traffic flow, but also wants to collect as many identifiable location samples from as many users as possible (effectively tracking their movement in this way). An additional, separate adversary is modelled in this context that does not collude with the service operator and strives towards disturbing the system and abusing it for influencing traffic.

OverDrive realizes a decentralized, peer-to-peer architecture. The adversary is assumed to control a portion of the overlay network. Nodes controlled by the adversary do not have to correspond to actual users. However, the use of a sybil-resistant pseudonymization scheme (as developed in Chapter 6) is assumed, so that the number of nodes under the adversary's control is bounded. During the evaluation, it is assumed that the adversary can control up to 1% of the node population. If sybil attacks are not possible and a large user population is assumed, infiltrating 1% of the node population represents a significant challenge. Nodes can't choose their physical location arbitrarily, but nodes controlled by the adversary can lie about their own location. Goals of the adversary in this scenario are the collection of location samples from all participating nodes, i.e., establishing a view similar to that of an operator in a comparable centralized system, as well as the targeted tracking of individual users.

BitNym is based on the usage of block chain networks as distributed append-only bulletin boards. The adversary is assumed to be unable to attack the underlying block chain network, so that its security properties are preserved. In the case that the Bitcoin network is used, for example, it is assumed that no entity can control more than 50% of the computing resources in the network (cf. Section 2.4.5). However, the adversary can deanonymize regular value transfers via Bitcoin, i.e., identify the pseudonymous addresses used in them. The focus of the adversary in the context of BitNym is on breaking pseudonyms, as it enables the identification of pseudonymously shared location samples.

3.3 Evaluation approach

In the following, it is described how, based on the previously discussed assumptions and adversary model, the privacy characteristics of the proposed systems are evaluated. The proposed approach is inspired by *LINDDUN* [28] and similar approaches to privacy threat modeling [26]. Like in related approaches, privacy threats and corresponding protection goals are first identified, followed by a structured search for possible attacks. The feasibility of individual attacks is then discussed and, where applicable, evaluated using simulations.

In contrast to more practice-oriented approaches (like LINDDUN), the approach used here follows a narrower scope in terms of the considered system model and protection goals. Implementation-specific details are abstracted away to a large extent. Only the four most relevant privacy threats for this thesis - identifiability, linkability, detectability and information disclosure - are considered (cf. Section 2.3.1). This reduction of scope enables a more exhaustive consideration of possible attack vectors. Also, threats are not seen in isolation, but as contributors to the main adversary goal of obtaining identifiable location samples. In this way, a better comparability of evaluation results can be achieved.

Note that more security-related attack goals, like abusing a given system or creating sybil identities, are not covered by the approach outlined here. The approaches used for evaluating these aspects are more specific and are discussed when considering the concrete contexts.

The employed approach consists of several steps that are grouped based on the privacy threats most prominently addressed through them. The two groups of steps are *detection and disclosure* and *linking and identification*.

3.3.1 Detection and disclosure

Here, the amount, content and quality of the data that the adversary is able to gather is determined. Following steps can be identified:

1. *Identifying knowledge from passive observation.* Based on the considered system and the entities controlled by the adversary, the data collectable by him under normal operation is identified. This includes all messages exchanged with compromised entities, their content and metadata, including source identifiers (if available) and timing data.

2. *Identifying possible actions.* Here, all actions are enumerated that the adversary is able to perform within the system model and that exceed the actions performed during regular operation. In other words, actions are identified that

might be used for constructing active attacks. This includes malicious behavior like lying or not following protocol specifications. Actions that are not part of the adversary model (e.g., physical surveillance) are not considered.

3. *Evaluating possibilities for increasing knowledge.* For each combination of identified actions, it is investigated to what extent it can be used as an active attack for increasing the adversary's knowledge. Additional constraints, like the aim to preserve a service's utility if colluding with the service operator, are taken into account.

Considering the complex scenarios addressed in this thesis - involving, e.g., vehicular traffic and overlay networks - a purely analytical approach is often insufficient for evaluating detection and disclosure threats. Thus, on several occasions, simulations are complementarily used to determine the characteristics and extent of collected data as well as the effects of attacks.

3.3.2 Linking and identification

Here, the extent is determined to which the adversary can extract identifiable location samples from the data available to him (according to the preceding evaluation steps). Following steps are relevant in this context:

1. *Direct identification.* It is investigated to what extent location samples known to the adversary can be identified directly, i.e., by using explicit links to other data items.

2. *Linkability.* For each combination of data types known to the adversary, it is determined whether links between individual data items of these types can be established. Linking might be possible based on spatio-temporal similarities between location samples.

3. *Identification attacks based on linked data.* The feasibility of identification attacks based on clusters of linkable data items (trips, in the case of linkable location samples) is investigated. Correlation- and context-based inference attacks can be considered in this context.

Linking and identification attacks from the literature can often be reused if the data available to the adversary can be matched to the setup in the original publications. In the scope of this thesis, such mappings are performed whenever possible to avoid the replication of existing results. If no mapping is possible, the possibilities for linking and identification attacks based on the available data are discussed analytically.

4. Privacy-preserving cooperative route planning

This chapter deals with one of the most popular and relevant smart traffic applications today - improving overall traffic flow. The focus is on *cooperative route planning*, a concept for optimizing vehicular routing on a global scale by gathering data about planned routes. As in other smart traffic applications, the benefits of such a system come at the cost of an increased privacy risk for participating users. Planned routes include both the current and the planned future locations of drivers and passengers - all highly privacy-relevant pieces of information. With only an indirect benefit from publishing their intended route, users might choose against participating in cooperative route planning if they face the danger of privacy loss. Lower adoption rates, in turn, decrease the utility of cooperative route planning systems.

It is, therefore, desirable to allow users to participate in cooperative route planning anonymously. However, without additional precautions, granting users anonymity can enable wide-scale abuse and thus greatly deteriorate service utility. How to resolve this tension is not obvious and poses a significant challenge in the context of cooperative route planning.

Addressing this challenge, a system is proposed here that realizes cooperative route planning in a privacy-preserving manner while offering protection against users that repeatedly publish false plans. According to the author's knowledge, the work presented here is the first that addresses this combination of requirements.

Cooperative route planning and the challenges addressed in this chapter are described in greater detail in Section 4.1. Additionally, the adversary model for this chapter is concretized. Following a discussion of related work (in Section 4.2), a system is proposed that enables users to publish travel plans in an anonymous fashion.

More specifically, users can publish *promises* - intents to pass at specific waypoints at approximate times. While the unlinkability of promises to individual users is ensured, lying users are quickly excluded from participation.

A high-level overview of the general scheme and the underlying system architecture is given in Section 4.3. In Section 4.4, the design is concretized and a detailed description of the involved entities, objects and mechanisms is provided. A privacy analysis of the proposal is performed in Section 4.5, following the approach from Chapter 3. The privacy analysis is followed by an analysis of the system's security against different forms of abuse (attacks on the functionality of the system), presented in Section 4.6. In Section 4.7, possible performance bottlenecks of the solution are evaluated to prove its practicability. The chapter concludes in Section 4.8 with a summary of results.

This chapter is largely based on [41].

4.1 Cooperative route planning

The distribution of vehicular traffic today is still largely inefficient. Overburdened roads lead to congestion, traffic accidents and increased pollution due to stop and go. Adaptive route planning based on traffic monitoring is a widely accepted measure for improving present inefficiencies. Vehicles receiving traffic updates can adapt their routes accordingly, thus reaching their destinations quicker and improving the overall traffic flow. Most currently deployed systems restrict themselves to estimations about the current traffic situation. Predictions about the future, if at all, are made only based on the current state and historic data. The growing interconnection of vehicles and infrastructure, however, enables more advanced forms of cooperation in vehicular route planning. Information about planned routes can be shared by vehicles on the road (respectively their navigation devices), leading to more precise traffic predictions. Precise predictions, in turn, enable the selection of faster routes and result in an improved traffic flow [23, 34]. In the scope of this thesis, the approach of sharing planned routes and integrating the plans of others into route planning decisions is referred to as *cooperative route planning*.

4.1.1 Example use case and potential benefits

The cooperative route planning concept can be motivated using a typical use-case. Consider an ideal route planning system and the following scenario: Before a user starts his trip, he enters his destination into his navigation device[1]. His navigation device receives updates about the current traffic situation as well as precise predictions about future developments. In this way, it can propose a route, a start time and times and locations for possible coffee breaks, so that the user wastes as little time and fuel as possible. As a side-effect, by ensuring a fast and uncongested journey for

[1] Automated route planning is even more interesting when considering self-driving cars which, unlike human drivers, cannot function without it. Also, they always adhere to the route proposed by their route planning component.

the user, his navigation device also contributes towards improving the global traffic flow for all traffic participants.

The effectiveness of the navigation device's planning largely depends on the quality of the data it receives. For the current traffic state, good estimates can be obtained by gathering data from road side infrastructure or speed measurements from vehicular fleets. For calculating the future traffic state, on the other hand, input from all other vehicles on the road is desirable for deriving an accurate forecast. If all vehicles on the road publish their intended route and consider the published routes of others, significant improvements in overall traffic flow can be achieved [23, 34].

In its pure form, cooperative route planning has a strong altruistic element - by publishing his plans, a user contributes to the common good. Additional incentives can easily be introduced into such a system as well. For example, participating users can receive benefits, like road toll discounts or the right to use a reserved lane, if they publish their plans beforehand.

Returning to the previous example, a navigation device implementing cooperative route planning can announce that the user will be passing a certain, frequently congested road segment at some approximate time in the future. On a trip to Switzerland, for example, it can announce that they will be passing the Gotthard tunnel between 13:15 and 13:30 on the same day. When they indeed arrive at 13:23, the user is able to use a reserved lane and pass the tunnel immediately.

4.1.2 Privacy and security challenges

Independently of any benefits, insufficient trust in the privacy provided by a cooperative route planning system can hamper the deployment of such a system. Users may weigh their own privacy higher than the common good (in the form of better global traffic flow) or the incentives a service operator may offer. Additionally, as was already discussed in Chapter 2, the centralized collection of large amounts of privacy-relevant data might also be undesirable on a societal level. As in most smart traffic scenarios, a central challenge is the protection of location data. Both current and intended future locations should be shared in such a way that they are not easily identifiable. Additionally, to prevent profiling and identification based on context knowledge, the reconstruction of user trips should be hampered as well.

Somewhat in opposition to these requirements to user privacy, several security aspects must also be considered to guarantee the effectiveness of cooperative route planning. It must be ensured that the plans reported by users reflect their actual intentions and that the number of plans a user can publish is bounded. Without precautions, malicious users can influence traffic flow by distributing large numbers of fake plans (a form of sybil attack). In addition to controlling the registration to cooperative route planning systems, mechanisms are needed for monitoring that plans have actually been fulfilled. The latter is also a precondition to the implementation of incentives (e.g., the right to use a reserved lane if participating honestly).

4.2 Related work

Cooperative, anticipatory route planning has been widely discussed in the context of intelligent transportation systems [23, 34]. Existing works focus on improving route planning decisions based on collected planning data from users. Proposed systems are evaluated using traffic simulations. Evaluation results demonstrate that significant improvements in average travel times are possible, especially in scenarios with a high traffic density. While these contributions are valuable for advancing the cooperative route planning concept, no works are known that consider the issue of protecting user privacy or minimizing the systems' vulnerability to abuse.

In addition to academic works on the subject, the practical deployment of cooperative route planning approaches is gaining traction as well. Examples include the smartphone app *NUNAV*[2]. As in the discussed academic proposals, the destinations of all users are collected in a centralized fashion. Route planning algorithms are continuously applied on the data by the service operator and resulting route recommendations are communicated back to users. Since the actual route planning is performed by the service operator, he needs to be aware of the current location and the destinations of all users.

As was already discussed in Section 2.3, there is a plethora of works on location privacy and privacy for collaborative sensing that focus on achieving different privacy protection goals while performing operations on a user's current location. While there are many works dealing with privacy in systems reporting a user's current state, very few works are known that deal with intention privacy.

In [19], Chim et al. propose a scheme for making power usage reservations in a smart grid context using blindly signed anonymous credentials. While their scenario and approach are similar to the ones in this chapter, their scheme has important drawbacks. For example, it only guarantees that users will consume the reserved amount of energy per day and not whether they will consume it at the specific time frames they claim. Additionally, the electricity provider in [19] is assumed to know the total daily power consumption of a user. In a traffic scenario, this would correspond to the traffic authority or a private company knowing the distance traveled by each participating vehicle, which is undesirable from a privacy standpoint and difficult to realize. The idea of preventing abuse using single-spend, renewable tokens has also appeared in the context of privacy-preserving subscription services [106]. However, the question of realizing anonymous promises and reservations has not been raised in these works.

Privacy-preserving reputation systems have been proposed for preventing abuse in systems with pseudonymous participants (see also Section 2.5.2). Androulaki et al., for example, propose an approach centered around the idea of *repcoins* [1], which, similar to the approach introduced in this chapter, are based on David Chaum's original electronic cash idea [16]. However, their scheme does not consider the promise scenario, i.e., is lacking a mechanism for publishing promises in such a way that rep-

[2] `http://www.graphmasters.net/`

utation can be gained on promise fulfillment without enabling an adversary to link the promise back to its author. The *IncogniSense* framework [22] enables reputation transfers between pseudonymous identities that preserve the unlinkability between pseudonyms. Layering a cooperative planning system on top of IncogniSense, the same pseudonym could be used for publishing and fulfilling individual promises, with the pseudonym's reputation score incrementing upon fulfillment. In order to avoid linkability between promises however, a different pseudonym will have to be used per promise. This requirement is difficult to reconcile with IncogniSense's periodic pseudonym switching mechanism, resulting in a more complex and computationally demanding solution than the approach proposed here.

Another related research area is privacy-preserving incentive systems. In [72], Li et al. propose two schemes for privacy-preserving incentives in mobile sensing scenarios. One of their schemes is based on a trusted third party, the other uses a chain of blindly signed tokens for ensuring the payment of users and protecting against cheating (in this context, committing data for the same sensing task multiple times). While related to the problems tackled here, the problem of publishing promises and verifying promise fulfillment in a privacy-preserving manner is not considered.

To the best of the author's knowledge, the presented work is the first to tackle the problem of publishing planned routes in a vehicular route planning context while protecting the past and future locations of users and effectively limiting abuse.

4.3 High-level overview of solution

An abstract overview of the proposed solution is given here. The *promise coin* (PC) construction is introduced as well as its application to the problem of realizing abuse-resistant cooperative route planning without compromising the location privacy of users.

4.3.1 Promise coins

A central challenge when considering stronger privacy protection in cooperative route planning systems is to ensure that malicious users cannot disturb the functionality of the system. A single malicious user might, for example, repeatedly publish made-up routes, making individual road segments appear overburdened and thus greatly altering the flow of traffic. Given a simple, registration-free system with full user anonymity (i.e., users are anonymous during all interactions with the system), it is impossible to exclude such users even if their misbehavior can be identified. Therefore, the promises of other participants can't be fully trusted, which leads to a degraded system utility. On the other hand, anonymity and the unlinkability of user actions are highly desirable from a privacy standpoint.

To answer this tension, a scheme is proposed based on the novel concept of *promise coins* (PCs) - cryptographic constructs related to Chaum's electronic cash [16]. In electronic cash, financial authorities issue digital bills to users using *blind signatures*, i.e., without being able to link these bills to the receiving users later on.

A pool of PCs is issued, using blind signatures, to each user upon (identifiable) authentication at a central authority. In the following, this central authority is referred to as the *promise authority* (PA). PCs are used to make anonymously published *promises* about a user's planed route trustworthy. From a high-level view, a user publishes a promise by *paying* the PA one PC for it. In exchange, he receives a *promise token* from the PA. Once the user proves that he has fulfilled his promise, or if he revokes the promise in a timely manner, he can redeem the promise token and *receive a new PC* from the system. Depending on the deployment scenario, users might additionally receive benefits when redeeming a promise token, like a discount on the road toll for the current road segment. Promises can also be treated as reservations. For example, only users with valid promise tokens might be allowed to pass a given tunnel or bridge.

Honest users will never run out of PCs and will never need to perform the identity-coupled PC issuing protocol with the PA again. Uncooperative users, on the other hand, will run out of PCs upon repeated misconduct and can be denied new PCs by the PA, effectively banning them from the system.

An unspent PC is a guarantee for the PA (and other users) that the anonymous user owning the PC is trustworthy enough for publishing at least this one promise. In contrast to regular currencies, the semantic of a PC is therefore not "money" but trust, and fulfilling promises regenerates trust.

Using blind signatures for the issuing of PCs, it is ensured that they are completely unlikable to specific users or to other PCs of a user. The properties of blind signatures are described in Section 4.4.6, together with an example implementation based on the RSA cryptosystem.

4.3.2 Cooperative route planning with promise coins

When calculating routes, users take traffic updates from the *promise board* (PB) into account. The PB is the publicly accessible database of all currently relevant promises. Traffic updates from the PB include traffic forecasts based on the published promises of other participants. Once a route has been found, the user (respectively its navigation device) identifies parts of its route for which promises can be made. Specifically, it identifies *resources* (road segments, tunnels, etc.) along its planned route, for which a *resource authority* (RA) has been deployed[3]. For each of these resources, it then formulates a *promise* containing a resource identifier and a time frame (e.g., a 15-minute interval) during which it expects the resource to be passed. Using an anonymous communication channel, this promise is transmitted to the PA together with a valid, previously unspent PC. After validation of both the promise and the PC, the PA commits the promise to the PB and returns a *promise token* (PT) to the user. Once the user reaches the resource region, he can show the promise token to the RA. If he arrived at the region within the promised time frame, the RA will cash in the promise token and issue a new PC to the user.

[3] Promises can also be made for resources for which no RA has been deployed. However, abuse cannot be effectively prevented for such promises, rendering them less trustworthy.

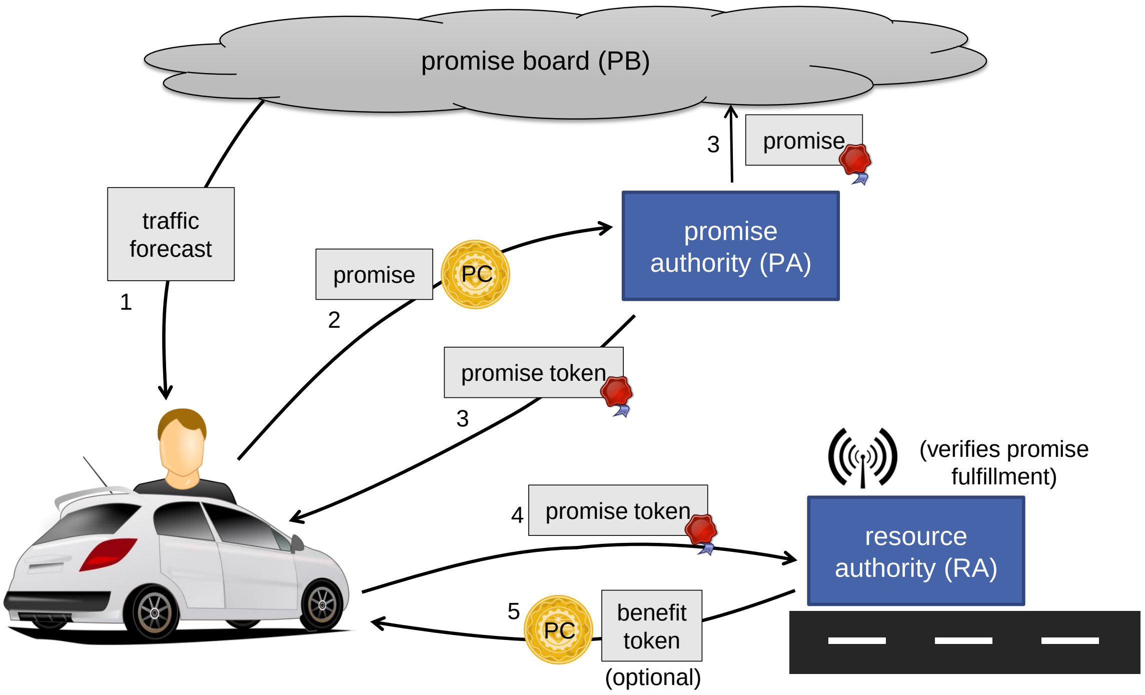


Figure 4.1 Cooperative route planning with promise coins - overview.

Figure 4.1 depicts an overview of these steps. The diagram depicts the distribution of traffic forecasts (step 1), the publication of a promise (step 2), the commitment of a promise to the PB and the issuing of a promise token (step 3), the promise token redemption request at the responsible RA (step 4) and finally the issuing of a new PC upon the successful redemption of a promise token (step 5). In the last step, the optional granting of a benefit is shown as well, in the form of issuing a *benefit token*. The diagram doesn't depict the identity-bound protocol for requesting new PCs from the PA. For honest users (and barring any hardware failures), this step is performed only once at the beginning of their participation in the system.

As in the remainder of this thesis, while the term *user* is frequently used when describing and analyzing approaches, the actual work in a practical implementation will be performed by navigation devices or self-driving cars. The details of the proposed system can easily be hidden from human drivers and passengers, making the resulting system appear as a regular route planning application.

4.4 Design

Following the high-level overview given in Section 4.3, a detailed description of the proposal is now presented. A number of underlying assumptions are recapitulated first. The participating entities and employed data objects are then introduced. Following that, the mechanisms comprising the system are described. An overview of the involved cryptographic certificates and data objects can be found in Table 4.1.

4.4.1 Underlying assumptions

Several assumptions underlie the presented design that are also used for the analysis of the proposal. These reflect the general assumptions at the core of this thesis, as discussed in Chapter 3.

Two types of adversaries are considered:

1. An adversary attempting to compromise the privacy of users, modelled in accordance to the general adversary model discussed in Section 3.2. All entities controlled by the service operator are assumed to be also controlled by the adversary. The adversary has an interest in maintaining service utility. He performs no actions that would negatively impact traffic flow or the functionality of the system.

2. An adversary attempting to abuse the system for disrupting or manipulating traffic flow. His capabilities are restricted to match these of one or more regular users.

Concerning the required existing building blocks, blind signatures play a central role in the developed system. It is assumed that they are secure against all considered adversaries. Additionally, an anonymous and secure (i.e., confidentiality- and integrity-preserving) communication channel is required. Here, the *Tor* [31] network and *TLS* [29] are used as available building blocks. It is assumed that, in combination, they are secure against impersonation and eavesdropping attacks by parties not participating in the communication. Likewise, it is assumed that none of the considered adversaries can break the metadata protection provided by Tor.

Vehicles (and, consequently, users) are assumed to be equipped with both cellular (e.g., 3G or 4G) and short-range radio (e.g., IEEE 802.11p) communication interfaces.

All involved entities are assumed to share a globally synchronized clock for correctly evaluating promise fulfillment and promise revocation requests. The required clock precision depends on the chosen time frame granularity. For time frames of several minutes and more, errors in the range of a few seconds are acceptable.

4.4.2 Entities

This section introduces all entities relevant in the context of the proposed system, namely:

- the promise authority (PA)
- resource authorities (RAs)
- the promise board (PB)

4.4.2.1 Promise authority (PA)

The *promise authority* (PA) is the trust anchor in the proposed system. It maintains the publicly known *promise authority certificate* (PAC). The PAC is used for establishing trusted communication channels with the PA, signing promise tokens and

Object	Authentication	Function
PA certificate (PAC)	external, trust anchor	secure connections to PA, trust anchor
coin issuing certificate (CIC)	signed with PAC	blind signing of PCs
promise coin (PC)	signed with CIC	authenticating promises
promise	by spending a PC, later signed with PAC	publishing future intent
promise token (PT)	signed with PAC	redeemed for a new PC upon promise fulfillment
RA certificate (RAC)	signed with PAC	generating (public part) and cashing in PTs
user ID	external	requesting first batch of PCs

Table 4.1 Overview of used certificates and data objects.

validating actions concerning the state of the PB (e.g., the revocation of promises). The PA additionally maintains the *coin issuing certificate* (CIC). The CIC is only used for blindly signing PCs. A PC is only valid if it includes a signature by a currently valid CIC[4].

The PA can be maintained by a public traffic authority or a private company providing the cooperative route planning service. In the scope of this chapter, for simplicity, the focus is on a setup featuring one PA. In practice, the deployment of multiple PAs might be interesting for achieving higher fault tolerance and a decentralization of control.

4.4.2.2 Resource authorities (RAs)

A *resource authority* (RA) is an entity responsible for a specific resource, e.g., a road segment, bridge or tunnel. In order for a user to be able to make promises about a resource, a RA needs to be deployed for that resource. RAs are equipped with a *resource authority certificate* (RAC) signed by the PA. RACs can be used for authentication and their public part is used by the PA for generating promise tokens. All RAs, their resources and their RACs are publicly known. Information about RAs can be distributed, for example, by the PB.

The main function of a RA is the verification of the fulfillment of promises. Thus, a RA needs to be both reachable for users passing its resource and be able to verify that users are really passing it. For satisfying both of these requirements, RAs are envisioned to be equipped with short-range radio interfaces as widely discussed

[4] In a practical deployment, CICs will likely need to be rotated after certain periods to maintain the scalability of PC double-spending detection (cf. Section 4.4.4.3). A CIC rotation can happen seamlessly to users, with the PA offering to exchange PCs signed with the old CIC for new ones.

in the vehicular networking community [66]. Radio-equipped *road side units* (RSUs) are a central element of various vehicular networking architectures and are expected to find major deployment, e.g., for relaying safety messages and time-critical sensor readings. RAs should be implemented in such a way that they have access to one or multiple RSUs scattered around the respective resource managed by them. Other existing infrastructure like toll stations can be leveraged by RAs as well. However, it is assumed that RAs have no access to dedicated vehicle identification equipment like license plate scanners. While the anonymous publication of routes is still possible with RAs equipped in this way, the routes vehicles have already taken become difficult to anonymize.

Upon verifying that a promise has been fulfilled, RAs cash in the supplied promise token and hand out a new PC to the user. This step, again, involves only the RA and the user. In general, none of the actions performed by a RA requires any communication with the PA, PB or other RAs. Thus, as an important detail, RAs can safely be realized in an "offline" manner without any form of long-range (i.e., Internet) connectivity.

Concerning the number and deployment density of RAs, there are two interesting extreme cases: the *dense deployment* case, where all major roads are split into multiple short segments and mapped to individual resources, and the *sparse deployment* case, where RAs are only set up for common bottlenecks like tunnels, bridges or busy crossings. While the dense deployment case leads to a higher granularity of promises and, thus, potentially better traffic predictions, it is also tied to a larger initial investment for setting up RAs.

4.4.2.3 Promise board

The *promise board* (PB) is a publicly accessible database for promises. The main function of the PB is to keep track of published promises and to make this information public to all interested parties. Information from the PB can be used for making route planning decisions, by checking how much demand for a resource (e.g., a road segment) is expected in the future. Promises and promise revocations are pushed to the PB by the PA after they have been validated.

The PB can be realized as a centralized service, maintained by the service operator. In principle, it can also be realized in a decentralized manner using a distributed data structure like a DHT. *PeerTIS* [101], for example, is a DHT optimized for storing traffic-related data that could be extended for supporting time-dependent traffic estimations based on published promises. Since no privacy-related benefits are expected from a decentralization of the PB, this possibility is ignored in the following.

4.4.3 Objects

This section defines object types that are at the center of the proposed system, namely:

- promises
- promise coins (PCs)
- promise tokens

4.4.3.1 Promise

In the context of this chapter, a *promise* is a public statement of intent from an anonymous user to make use of a specific resource within a specific time frame. So, with *resourceID* denoting some sort of resource identifier and *timeframe* denoting a time frame, a promise P has the form:

$$P := (\text{resourceID}, \text{timeframe})$$

In a cooperative route planning context, a promise can have a semantic like "I will be passing the Gotthard tunnel going south, between 11:00 and 11:30 today.". Depending on the context and the type of the resource, a promise can also be treated as a reservation, i.e., only vehicles that made a reservation might be allowed to pass the Gotthard tunnel at high-traffic times. In a vehicular traffic context, resources represent short road segments with a length of up to several kilometers. For sharing whole routes, users publish multiple promises. Given a sufficiently high RA deployment density, a user can publish promises covering its entire planned route.

4.4.3.2 Promise coins

A *promise coin* (PC) is a cryptographic construct consisting of a random *coin ID* and a signature by the PA using the CIC. A PC X can be formally defined as:

$$\text{PC}_X := (\text{coinID}_X, \text{sig}_{\text{CIC}}(\text{coinID}_X))$$

PCs do not include a value field as they all have the same semantic: each PC can be used for publishing exactly one promise. This property offers the benefit that the blind signature protocol can be completed in one round and the user needs to generate only one coin candidate. Since all PCs have the same "value", the PA can safely sign any blinded coin ID it receives.

4.4.3.3 Promise token

Promise tokens (PTs) are generated upon the publication of a promise. They enable users to receive new PCs and benefits upon promise fulfillment. Promise tokens are composed of a promise, an encrypted blinded CIC signature for a new PC and a PAC signature. The blinded PC signature is encrypted in such a way that both the RA and PA can decrypt it in the case of promise fulfillment or promise revocation. We denote this encryption as $\text{enc}_{\text{RAC,PAC}}()$. In its simplest implementation, $\text{enc}_{\text{RAC,PAC}}()$ can be realized by concatenating the results of $\text{enc}_{\text{RAC}}()$ and $\text{enc}_{\text{PAC}}()$. The promise token $\text{PT}_{P,Y}$ for a promise P and a new PC candidate Y can now be formally defined as (using a temporary value a):

$$a := (P, \text{enc}_{\text{RAC,PAC}}(\text{sig}_{\text{CIC}}(\text{blind}(\text{coinID}_Y))))$$

$$\text{PT}_{P,Y} := (a, \text{sig}_{\text{PAC}}(a))$$

The mechanisms involved in constructing a promise token are introduced in greater detail in the remainder of this section. An example implementation of the blind() function is described in Section 4.4.6.

4.4.4 Mechanisms

In the following, the core mechanisms comprising the proposed system are introduced. These are:

- the establishment of communication channels
- the initial generation of promise coins
- the publishing of promises
- the fulfillment of promises and the redemption of promise tokens
- the revocation of promises
- the dissemination of traffic information

4.4.4.1 Establishment of communication channels

In order to avoid linkability of user actions based on communication metadata like communication addresses, an anonymous communication channel must be established for non-local communication (i.e., all communication over the cellular network). Examples for communication that needs to be protected in this way include the publishing and revoking of promises at the PA and, depending on the implementation, the querying of the PB for traffic forecast data. The use of Tor is proposed. Tor is an openly accessible anonymization service. It aims at offering low communication latencies and has the additional benefit that it establishes *circuits*, i.e., stateful paths through the underlying mix network that provide bidirectional communication channels. In the scope of this chapter, this enables responses to be sent via the same channel as the corresponding request. For further improving unlinkability, each circuit to the PA is used for only one message exchange linkable to a PC, and discarded afterwards. Thus, a user establishes a dedicated communication channel for each promise publication and revocation.

Non-local communication channels must additionally be secure, in the sense that the confidentiality, authenticity and integrity of communication is preserved. Since all non-user entities are equipped with verifiable cryptographic certificates and no direct user to user communication is required, the establishment of a secure connection over a Tor circuit is straightforward using, for example, TLS [29][5]. The steps a user needs to take for forming a non-local anonymous and secure communication channel are then the following:

[5] Alternatively, the PA could register a *hidden service* within the Tor network or use the PTP library (cf. Section 2.6.2.3). Such an approach realizes end-to-end confidentiality, authenticity and integrity. However, it also implies additional overhead for both the PA and the Tor network while its main benefits - network-independent reachability and receiver pseudonymity - are irrelevant in the considered context.

1. Establishing a Tor circuit with the destination.
2. Establishing a secure connection over the Tor circuit, e.g., using TLS.

For local communication, i.e., communication with RAs over short-range radio, linkability based on communication metadata can easily be avoided; users can freely choose (and change) any communication addresses they might be required to use for themselves (e.g., MAC addresses). Furthermore, user authentication, which might threaten anonymity, is not required in the proposed system. Concerning the authenticity of the RA as well as the confidentiality and integrity of short-range communication, neither are critical for the security of the proposed system. Messages can be exchanged immediately without a preceding setup phase.

4.4.4.2 Initial coin generation

During initial coin generation, a user receives a batch of PCs upon identification at the PA. The size of this batch is a system parameter that should be chosen based on the specific deployment conditions. For users that are honest and can therefore maintain a mostly stable supply of PCs, initial coin generation needs to be performed very rarely (e.g., when their coin supply dwindles due to occasional route planning errors or hardware failures). It must be performed at least once, namely when the user starts using the system for the first time. The mechanism is composed of the following steps:

1. For each new PC X that the user wants to request from the PA, he generates a new random coin ID.

$$\text{coinID}_X := \text{random}()$$

2. The coin ID is blinded and sent to the PA, together with a proof of the user's identity. Such a proof can easily be realized by, e.g., equipping navigation devices with certificates tied to the user's identity.

$$\text{PC request} := (\text{userID}, \text{blind}(\text{coinID}_X))$$

3. If the PA finds the user to be eligible for another PC (i.e., if he is in principle eligible to participate and hasn't received too many PCs already), it answers with a CIC signature on the blinded coin ID.

$$\text{PC reply} := \text{sig}_{\text{CIC}}(\text{blind}(\text{coinID}_X))$$

4. Using this signature and information about the applied blinding, the user can construct a signature for the original unblinded coin ID and thus assemble a new PC.

$$\text{PC}_X := (\text{coinID}_X, \text{unblind}(\text{sig}_{\text{CIC}}(\text{blind}(\text{coinID}_X))))$$

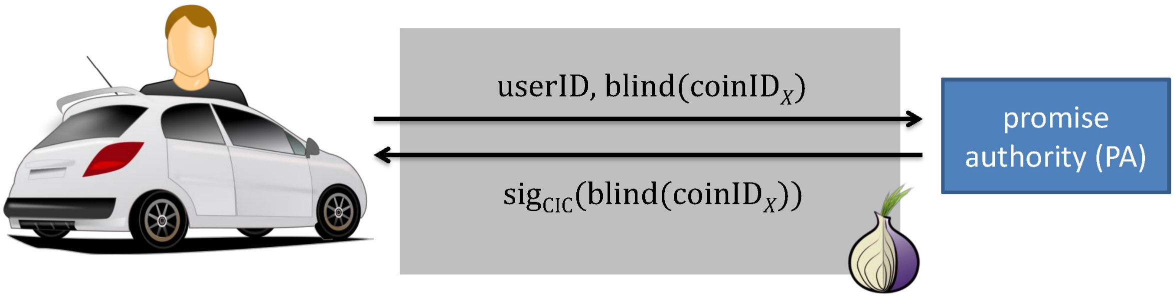

userID, blind(coinID$_X$)
sig$_{\mathrm{CIC}}$(blind(coinID$_X$))
promise authority (PA)

Figure 4.2 Initial issuing of one coin.

Parts of this mechanism are also depicted in Figure 4.2. A straightforward, RSA-based implementation of the $\mathrm{blind}()$, $\mathrm{unblind}()$ and $\mathrm{sig}_{\mathrm{CIC}}()$ functions is described in Section 4.4.6. Due to the blinding function used by the user, the PA never learns the original coin ID of the PC it is signing. Thus, it cannot link the resulting PC back to the user. The PA keeps track of the number of PCs issued to individual users upon user authentication. In this way, cheating users can be identified and denied new PCs.

4.4.4.3 Publishing a promise

As one of the main features of the proposed system, users can publish promises about their future intents. The process of publishing a promise P is composed of an exchange between the user and the PA. It can be subdivided into the following steps:

1. The user establishes an anonymous secure communication channel with the PA.

2. The user generates a new random coin ID. This will be the base for the PC he will get back upon promise fulfillment.

$$\mathrm{coinID}_Y := \mathrm{random}()$$

3. The user sends one of his PCs, his promise P, and the blinded new coin ID to the PA over the anonymous channel. He also stores the unblinded value of coinID_Y so that he can construct PC_Y at a later time.

$$\text{promise request} := (P, \mathrm{PC}_X, \mathrm{blind}(\mathrm{coinID}_Y))$$

4. The PA checks the validity of the used PC. Specifically, it verifies the validity of the supplied PC's signature and checks that no identical PC (i.e., with the same coin ID and signed with the same CIC) has been spent before. For the latter check, i.e., the prevention of double-spending, a data structure containing the coin IDs of previously spent coins must be consulted.

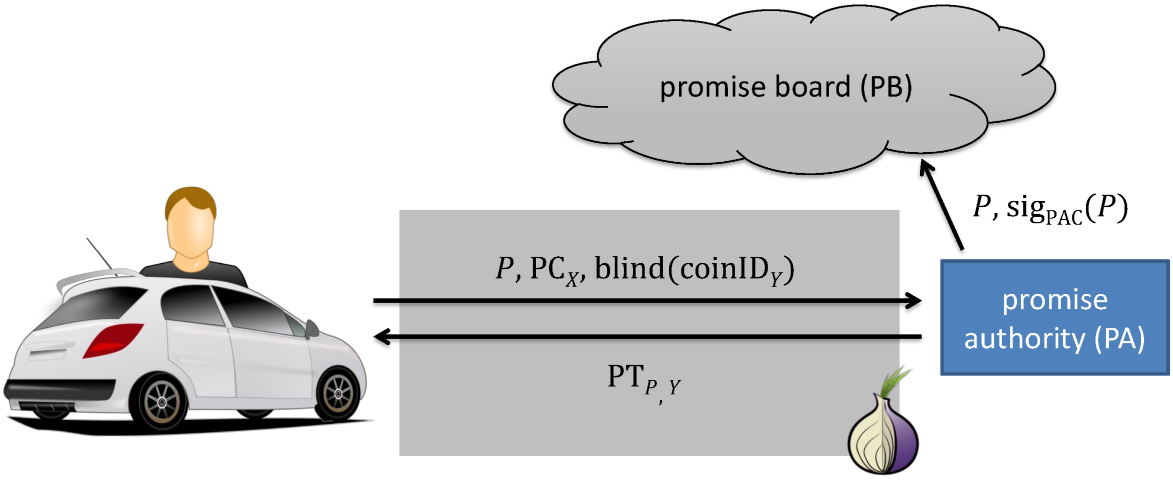

promise board (PB)
P, $\text{sig}_{\text{PAC}}(P)$
P, PC_X, $\text{blind}(\text{coinID}_Y)$
$\text{PT}_{P,Y}$
promise authority (PA)

Figure 4.3 Publishing a promise.

5. If the PA considers both the promise and the PC valid, it signs the promise and sends it to the PB. It also marks the PC as spent by storing its coin ID in the data structure consulted in the previous step.

6. The PA furthermore proceeds to generate a promise token for P and the new PC Y. It signs $\text{blind}(\text{coinID}_Y)$ and encrypts it in such a way that both the RA responsible for the resource in P and the PA itself can decrypt it. This is easily done using the public keys from the respective RAC and the PAC by using an asymmetric encryption scheme. Let a be the intermediate result of these steps.

$$a := (P, \text{enc}_{\text{RAC,PAC}}(\text{sig}_{\text{CIC}}(\text{blind}(\text{coinID}_Y))))$$

   Using $\text{enc}_{\text{RAC,PAC}}()$ instead of only $\text{enc}_{\text{RAC}}()$ is necessary for being able to redeem the token in the case of a revocation of P or a failure of the RA. The PA finishes the construction of a promise token for P and Y by signing a:

$$\text{PT}_{P,Y} := (a, \text{sig}_{\text{PAC}}(a))$$

7. The resulting promise token is sent back to the user via the same communication channel initiated by the user.

$$\text{promise acknowledgement} := \text{PT}_{P,Y}$$

The communication exchanges involved in publishing a promise are also depicted in Figure 4.3. In order to prevent the double-spending of PCs, the PA must store all spent PCs (more specifically, their IDs). This requirement can lead to scalability issues as the set of spent PCs can grow indefinitely. Thus, in a practical deployment, the CIC should be switched periodically. PCs signed with older CICs can be declared invalid so that only the spent PCs signed with the last few CICs need to be

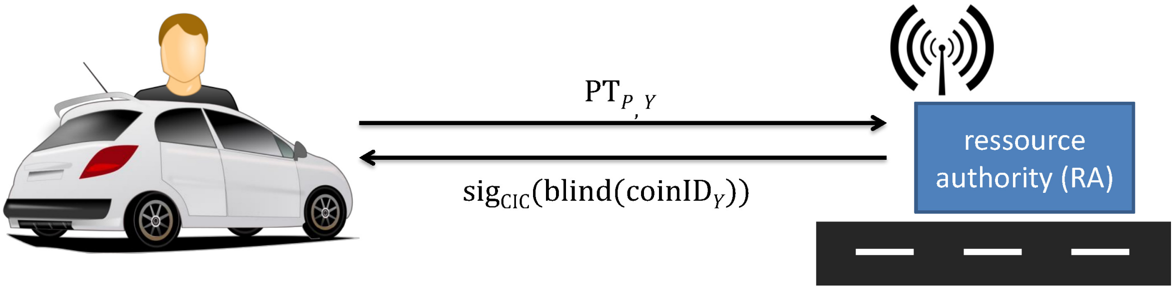

$\mathrm{PT}_{P,Y}$
$\mathrm{sig}_{\mathrm{CIC}}(\mathrm{blind}(\mathrm{coinID}_Y))$
ressource authority (RA)

Figure 4.4 Promise fulfillment and coin reissuing.

stored. The PA can offer an exchange service for PCs whose CIC has become invalid recently. The rotation of CICs can also be beneficial from a security standpoint [17].

4.4.4.4 Promise fulfillment and promise token redemption

RAs are responsible for verifying the fulfillment of promises and cashing in promise tokens. For a user that has published a promise P and has arrived at the resource mentioned in P[6] within the time frame mentioned in P (in other words, is fulfilling P), the specific steps are the following:

1. The user transmits, via short-range radio, the promise token $\mathrm{PT}_{P,Y}$ to the RA. This is the promise token he received from the PA upon the publication of P.

$$\text{fulfillment request} := \mathrm{PT}_{P,Y}$$

2. The RA verifies the PAC signature of the promise token. If it is valid, it extracts P from the promise token and determines if the user has fulfilled it. Most importantly, it verifies whether P concerns its own managed resource, whether the user is indeed within the limits of this resource and whether the time commitment noted in P has been held.

3. Upon validating that P was indeed fulfilled, the RA decrypts the blinded PC signature found in $\mathrm{PT}_{P,Y}$ and sends it back to the user.

$$\text{fulfillment acknowledgement} := \mathrm{sig}_{\mathrm{CIC}}(\mathrm{blind}(\mathrm{coinID}_Y))$$

4. Since the user knows both the original unblinded coin ID and the employed blinding function, the receipt of the blinded CIC signature is equivalent to receiving a new PC.

$$\mathrm{PC}_Y := (\mathrm{coinID}_Y, \mathrm{unblind}(\mathrm{sig}_{\mathrm{CIC}}(\mathrm{blind}(\mathrm{coinID}_Y))))$$

Figure 4.4 gives a graphical overview of the involved communication steps. Given a good placement of the RA's communication infrastructure (e.g., if it is distributed

[6] The arrival at a specific resource can be detected by users using GPS or short-range radio beacons from the RA.

over several RSUs covering the resource area), the proof that a user is indeed within the limits of a resource is implicitly given when communicating over short-range radio. For combating sybil attacks where a single vehicle attempts to redeem multiple promise tokens simultaneously, techniques based on radio-based position verification [49] or statistic signal strength distribution analysis [120] can additionally be deployed at RAs. In essence, such techniques work by identifying distinct radio stations based on signal strength, propagation delays or additional data sources like cameras. They also prevent *wormhole attacks* in which an adversary places a stationary transceiver in the resource area for redeeming promise tokens from a distance.

4.4.4.5 Promise revocation

Promise revocations are necessary if a user changes his route or finds out that he cannot reach the resource mentioned in his promise in time. The revocation mechanism is analogous to the promise fulfillment mechanism, with the main exception that it is performed between user and PA and that not promise fulfillment is checked, but whether the revocation is early enough to be considered valid. Upon a successful revocation, the PA decrypts the PC signature contained in the supplied promise token and sends an update to the PB.

In order to avoid the exploitation of the revocation mechanism by malicious users, e.g., the deliberate publication of false promises and their later revocation, revocations can additionally be penalized by the PA. Since PCs are atomic, fine-granular penalties can only be realized probabilistically. Depending on the application scenario, the PA might choose, with a probability p, to not issue a new PC after a promise revocation. The probability p is then the revocation penalty. On a side note, since RAs do not necessarily synchronize with the PA or PB, it is also possible for a user to both revoke a promise and later fulfill it at the respective RA. However, this leads to no benefit for the user. The impact of such a behavior on traffic prediction is comparable to the impact of a traffic participant unequipped with cooperative route planning capabilities. Traffic predictions will remain consistent, as RAs are only responsible for verifying the fulfillment of promises.

4.4.5 System management and policing

The proposed system offers multiple possibilities for adapting to specific scenarios and reacting to changes. For one, this includes adapting the number and deployment density of RAs. Additionally, a service operator can determine the number of PCs that users receive initially and the number of new PCs they are allowed to request before being blacklisted. These parameters can be made dependent of, e.g., the maximum expected route lengths in the scenario (with a safety margin) and experience-based likelihood estimations for the unintended non-fulfillment of promises (e.g., due to unforeseeable events or routing miscalculations).

The PA can also enforce different policies by varying the awards a user receives for fulfilling a promise. For example, revocations can have a chance of not regenerating a PC, i.e., the PA might supply a fake coin ID signature with a given probability p. Likewise, fulfilling promises faithfully might have a chance of spawning extra PCs for the user. The applicability and impact of different management and policing approaches is highly scenario-dependent. An in-depth discussion and evaluation of such issues exceeds the scope of this chapter.

4.4.6 Blind signatures and RSA blind signatures

In the following, an example realization of some of the cryptographic functions used so far is presented. Specifically, the functions required for the blind signing of PCs are defined, namely blind(), $\text{sig}_{\text{CIC}}()$ and unblind(). RSA-based signatures are used, as well as RSA blind signatures as introduced in [17].

The primitives *blind*() and *unblind*() form the cornerstone of the blind signature concept. Together with the coin signing function $sig_{CIC}()$ (that generates signatures with the *coin issuing certificate* (CIC)), they satisfy the following equation (for an arbitrary message m):

$$\text{unblind}(\text{sig}_{\text{CIC}}(\text{blind}(m))) = \text{sig}_{\text{CIC}}(m)$$

In the following, it is assumed that the employed CIC is based on a RSA key with d and e being the private and public parts of this key and N being its public modulus. With a random r relatively prime to N, the blinding of m can then be realized as:

$$\text{blind}(m) := mr^e \pmod N$$

In order for the unblinding step to work, $sig_{CIC}()$ must be implemented without any modifications (e.g., padding or hashing) on the message m after it has been transmitted to the signer (in our case, the PA). Thus, we use the plain RSA signature scheme for $sig_{CIC}()$:

$$\text{sig}_{\text{CIC}}(m) := m^d \pmod N$$

$$\text{sig}_{\text{CIC}}(\text{blind}(m)) := (\text{blind}(m))^d = m^d r^{ed} \pmod N$$

For unblind() we apply the inverse of r to the signature, thus receiving a valid signature on m:

$$\text{unblind}(\text{sig}_{\text{CIC}}(\text{blind}(m))) := m^d r^{ed} r^{-1} = m^d \pmod N$$

There are known dangers to using RSA blind signatures. If the key used for blind signing is used for encryption, an attacker can trick the signer to decrypt arbitrary bits of encrypted data. Thus, a dedicated key is used for signing PCs - the private part of the CIC. A second danger lies in the commutativity of unpadded RSA signatures as required for the RSA blind signature scheme. It can be exploited to generate more than one valid message-signature pair during the blind signing process. For example, a user can easily use multiple blinding factors on the same message. Unblinding separately with each blinding factor then yields multiple distinct messages with valid signatures. Thus, the use of the hash and sign paradigm is recommended in implementations using RSA blind signatures: blinding and signing not the PC itself, but a cryptographic hash of it.

RSA-based blind signatures were chosen here primarily due to their simplicity. More sophisticated (and complex) blind signature schemes, e.g., such based on bilinear maps [90], can be considered as well for practical deployments.

4.5 Privacy analysis

In this section, based on the approach described in Chapter 3, potential privacy threats relevant to the proposed system are considered. A system model is first formulated for establishing a suitable level of abstraction. Following that, the assumed adversary model is defined. Lastly, a comprehensive privacy analysis of the proposed system is performed. Based on the privacy threats considered through them, the applied analysis steps are separated into two groups:

1. Detection and disclosure.
2. Linking and identification.

4.5.1 System model

An implementation of the proposed system is assumed that features one dedicated PA, one PB and multiple RAs associated with different resources. Various deployment densities of RAs are considered. For simplicity, the PA, PB and RAs are all assumed to be run by the same organizational entity, e.g., a national traffic authority or a private company.

In addition to these centrally controlled entities, it is assumed that the majority of traffic participants is using the proposed system and that the majority of users are honest. While it is not necessary that all vehicles on the road contribute by publishing promises (with knowledge about the usage ratio, interpolations can be made), a critical mass of users is desirable for attaining the full benefits of cooperative route planning[7].

Further assumptions include that:

- All employed communication channels are anonymous. The existence of communication metadata (e.g., MAC addresses that are used for only one message exchange) is, therefore, abstracted away.
- CIC rotations are rare[8]. Therefore, the CIC used for signing a PC has no significant implications for the linkability between PCs.

[7] Note that an in-depth exploration of the relationship between user penetration rate, RA density and system utility exceeds the scope of this chapter.

[8] As an example, given a user population of 10 million, an average number of daily PC uses per user of 100, a storage requirement per spent PC of 10 B and a maximum storage capacity for spent PCs of 1 TB, the PA can go for 100 days without changing the currently active CIC.

4.5.2 Adversary model

A strong adversary is assumed that controls the PA, the PB and all RAs. This corresponds to a service operator that is either malicious himself or is heavily colluding with a malicious entity. As a constraint, it is assumed that the adversary cannot deploy additional infrastructure elements, e.g., in the form of dedicated tracking equipment like license plate scanners.

The main goal of the adversary is to obtain identifiable location samples from users - concerning both past and planned future locations.

4.5.3 Detection and disclosure

Starting with the actual analysis, the amount and structure of the data that the adversary is able to gather is determined here. Following things are considered:

1. The knowledge from *passive observation* that the adversary can acquire during normal operation.
2. All *actions* that are possible for the adversary within the system model.
3. The adversary's possibilities for *increasing knowledge* based on the actions possible for him.

Following that, the resulting types of data items available to the adversary are fused and filtered, arriving at more refined data item types for the subsequent analysis steps.

By applying these analysis steps, relevant types of data obtainable by the adversary can be determined. A concrete estimation of the volume and content of the collected data is impossible using only analysis. These characteristics depend on deployment-specific system parameters (e.g., the RA deployment density) and properties related to actual vehicular traffic. Traffic-related properties include traffic volume, the likelihood of different trips (respectively origin-destination pairs) and the resulting distribution of vehicles in the road network. Traffic simulations are an established approach for capturing such properties and arriving at a more concrete assessment of systems involving vehicular traffic. However, the implementation of a realistic simulation study of the proposed system exceeds the scope of this chapter. Instead, an analytical discussion is presented.

4.5.3.1 Knowledge from passive observation

The adversary controls the PA, the PB and all RAs. Thus, the knowledge available to him during normal operation corresponds to the data collectable by these entities. For every data item he can collect, the adversary can also store the time at which the data item was collected. For received messages, this corresponds to the time at which the message was received. The following can be collected:

- By the PA:

  - All PC requests (cf. Section 4.4.4.2):

$$(\text{PC request}, \text{time}) = (\text{userID}, \text{blind}(\text{coinID}), \text{time})$$

  - All promise requests (cf. Section 4.4.4.3):

$$(\text{promise request}, \text{time}) = (\text{promise}, \text{PC}, \text{blind}(\text{coinID}), \text{time})$$

  - All revocation requests and the times at which they were sent (revocation requests contain, in essence, the same information as promise fulfillment requests; cf. Section 4.4.4.5):

$$(\text{fulfillment request}, \text{time}) = (\text{PT}, \text{time})$$

- By the PB: depending on the implementation, users might need to send queries to the PB for receiving traffic data. Information about a user's position and planned route might be deduced from such a query. However, query privacy is considered to be out of scope here and is not considered further. Multiple promising solutions to the problem of query privacy in location-based systems have been discussed in the literature[9]. A simple solution is to omit the querying step altogether and have the PB proactively broadcast aggregated information to users, similar to the traffic message channel widely used in navigation systems today.

- By RAs:

  - All promise fulfillment requests (cf. Section 4.4.4.4):

$$(\text{fulfillment request}, \text{time}) = (\text{PT}, \text{time})$$

  - Depending on the capabilities of RAs, they might also be able to collect precise location samples from users within the regions of their respective resources. This ability might be needed for detecting sybil attacks on the promise fulfillment step. As an extreme case, it is assumed in the following that short trips (sets of linkable location samples) within resource regions can be learned. These trips can likely be linked to promise fulfillment requests. In the end, data items of the following form can potentially be collected by RAs:

$$(\text{PT}, \text{time}, \text{trip within resource region})$$

    Note that the quality of the data collectible by the adversary can vary greatly depending on the range of RAs, the employed tracking mechanisms and the communication behavior of users within resource regions.

[9] See, e.g., [115] for a good overview on the topic.

Depending on implementation (e.g., offline versus online RAs), it is also not immediately clear whether RAs will be able to store and transmit collected data to the adversary. In practice, RAs might even be controlled by separate and mutually non-colluding organizational entities (e.g., local authorities or traffic infrastructure owners), making it more difficult for the adversary to obtain all data generated at RAs.

4.5.3.2 Possible actions

In addition to all actions performed during regular operation, and excepting any actions not covered by the adversary model (e.g., deploying additional surveillance infrastructure or physically altering the traffic flow), the adversary is left with delaying, altering or omitting messages sent by the PA, PB and RAs. These messages are:

- PC replies sent from the PA during the issuing of new PCs (cf. Section 4.4.4.2)
- traffic updates from the PB
- promise acknowledgements from the PA (cf. Section 4.4.4.3)
- fulfillment acknowledgements from RAs (cf. Section 4.4.4.3)
- revocation acknowledgements from the PA (cf. Section 4.4.4.5)

The delaying, altering or omitting of most of these messages by the PA can easily be detected by users. The only exception are traffic updates which can be altered without a straightforward possibility for detection. However, the altering of traffic updates for performing some sort of privacy attack will likely have a negative impact on overall traffic flow. According to the assumed adversary model, a worsening of traffic flow is undesirable for the adversary. Thus, the altering of traffic updates for soliciting additional data from users is not considered further.

Concerning the delaying, altering and omitting of the remaining messages and the delaying and omitting of traffic updates, such actions will, due to their easy detectability, lead to a lowering of the service operator's credibility. A loss in credibility will, in turn, likely diminish user numbers and, in this way, decrease the overall effectiveness of the system. This effect is, again, undesirable for the adversary according to the assumed adversary model. Additionally, privacy-conscious users detecting non-standard behavior from a system component might immediately cease interacting with the system for fear of disclosing too much private information. In this way, potential active attacks are thwarted easily.

4.5.3.3 Increasing knowledge

As was argued in the previous section, no actions exceeding regular operation are worthwhile for the adversary within the considered system and adversary models. Consequently, no possibilities for increasing the adversary's knowledge through additional actions are considered.

4.5.3.4 Data fusion and reduction

The goal of this section is to arrive at a more concise representation of the data item types collectible by the adversary and in this way assist the subsequent analysis. The explicit linkability between different data item types is investigated with the goal of fusing multiple data item types into new, richer ones. Additionally, fields that are arguably anonymous, i.e., do not allow any form of linkability or identifiability, are identified and filtered out from representations.

Let $t_{\text{PC request}}$, t_{promise}, $t_{\text{revocation}}$, $t_{\text{fulfillment}}$ denote the times at which, respectively, PC requests, promise requests, revocation requests and fulfillment requests have been received by the adversary. Let trip_X denote a trip within an area identified by X. Then, following types of data items are available to the adversary according to the preceding analysis:

1. $(\text{userID}, \text{blind}(\text{coinID}), t_{\text{PC request}})$

2. $(\text{promise}, \text{PC}, \text{blind}(\text{coinID}), t_{\text{promise}})$

3. $(\text{PT}, t_{\text{revocation}})$

4. $(\text{PT}, t_{\text{fulfillment}}, \text{trip}_{\text{resourceID}})$

It is now discussed how these data item types can be reduced and merged while preserving the correctness of analysis.

Given the use of a secure blind signature scheme, no linkage between a blinded coin ID and a PC is possible. Furthermore, recall that coin IDs are random and not reused. In effect, the coin IDs used in instances of 1 and 2 will always be different. It is therefore concluded that no linkage based on blinded coin IDs is possible and any $\text{blind}(\text{coinID})$-entries can be ignored in the following.

PCs, while used only once each, might offer a limited form of linkability based on the CIC with which they are signed. In practice, CICs might need to be rotated periodically, so that it cannot be guaranteed that all PCs in use are sharing the same CIC. However, according to the assumed system model, CIC changes will happen only rarely. Additionally, users have the possibility to (anonymously) exchange PCs signed with an older CIC for new ones. The threat that PCs can be linked based on the CIC used for signing them is, thus, ignored in the following. PCs are considered completely unlinkable and, therefore, excluded from future discussion.

Collected promise revocation requests (item 3) can be safely excluded from consideration as well. They do not offer, in qualitative terms, any information gain compared to collected promise fulfillment requests (item 4).

Concerning promise tokens, they are explicitly linkable to previously published promises and, thus, enable a linking between items of type 2 and 4. Beyond this and apart from the content of promises (resource identifiers in combination with a time frame), promise tokens do not contain any additional information that may be of interest to the adversary. The remaining contents of a promise token are signatures which originate from the PA and, thus, the adversary himself.

Based on these considerations, the types of data items available to the adversary that are relevant for further discussions can be reduced to the following two:

1. $(\text{userID}, t_{\text{PC request}})$
2. $(\text{resourceID}, \text{timeframe}, t_{\text{promise}}, t_{\text{fulfillment}}, \text{trip}_{\text{resourceID}})$

Without additional knowledge and without resorting to correlation-based linking attacks, no linkage between these two data item types is possible.

4.5.4 Linking and identification

Here, it is discussed to what extend the adversary can extract identifiable location samples from the data available to him according to the preceding steps. Recall that a location sample was defined as a tuple of a location and a time, i.e.:

$$\text{location sample} := (\text{location}, \text{time})$$

The analysis is structured around discussing following approaches possible for the adversary:

1. Direct identification.
2. Linkability.
3. Identification attacks based on linked data.

4.5.4.1 Direct identification

The direct identification of a data item is possible if one of its fields or a combination of its fields can be linked to a user identity.

Recalling the results from before, data items of the following types are available to the adversary:

1. $(\text{userID}, t_{\text{PC request}})$
2. $(\text{resourceID}, \text{timeframe}, t_{\text{promise}}, t_{\text{fulfillment}}, \text{trip}_{\text{resourceID}})$

User identities are included only in 1. Data items of this type do not include location samples. Furthermore, they are generated during the initial obtaining of PCs, which, for honest users, will need to be performed only very rarely. One piece of information that the adversary gains from 1 is that a given user is participating in the system. By collecting all generated data items of this type, he can learn the identities of all users of the system. Such knowledge can make the obtaining of context knowledge easier and also helps with the reduction of anonymity sets, as non-registered traffic participants do not need to be considered.

Actual location samples are included only in 2, in the form of resources (with varying granularity) and potentially also more fine-granular trips within resource regions. While a data item of this type is not immediately identifiable, an identification might be possible with additional context knowledge. An exact estimation of this danger is not possible without detailed quantitative and qualitative information about the data included in 2. Estimates can be obtained using traffic simulations and a complete implementation of cooperative route planning using PCs.

4.5.4.2 Linkability

As was already discussed, $(\text{userID}, t_{\text{PC request}})$-tuples do not contain location samples. Furthermore, they are usually generated only very rarely. They are, thus, not expected to contribute to an increased linkability between data items containing location samples and are ignored in the following.

Instead, the focus is on discussing the linkability of data items of the following form:

$$(\text{resourceID}, \text{timeframe}, t_{\text{promise}}, t_{\text{fulfillment}}, \text{trip}_{\text{resourceID}})$$

If the adversary can link such items together, he can reconstruct long-distance trips taken by users. As was shown in the literature (cf. Section 2.2.4), such trips can be highly identifiable if a minimum amount of context knowledge (e.g., about the homes and workplaces of users) is available. Thus, the forming of such trips is a serious threat.

It can be argued that the resourceID, timeframe and $t_{\text{fulfilment}}$ fields offer an insignificant information gain compared to the trip within the resource potentially recorded at the respective RA[10]. Therefore, the following discussion is focused on linkability based on recorded short trips and the times of promise publications t_{promise}.

The question of linkability based on short trips can be mapped to the question of tracking feasibility in existing scenarios from the literature. In [59], for example, Hoh et al. propose a system in which location samples are shared only when passing preconfigured virtual trip lines. The resulting scenario is very similar to the one considered here, where short-range radio communication happens only within resources that might not cover the complete road network without gaps. Based on the positive results from [59], it is expected that the reconstruction of long-distance trips using only the available RA-recorded trips will yield poor results for the adversary. It should be noted, however, that in a very dense deployment scenario, i.e., if the complete road network is covered by RAs without gaps, the reconstruction of long-distance trips can become easier. Areas in which no communication with RAs occurs perform a similar function as mix zones (cf. Section 2.3.4) - they increase the number of potential hypotheses for the adversary and, therefore, his degree of linking uncertainty. Without such areas, no mixing is possible.

[10] It is implicitly assumed that every user chooses from the same selection of non-overlapping time frames (e.g., evenly spaced intervals from a common start time), so that no linkability based on the choice of time frame boundaries is possible.

Additional correlation attacks might be possible by also using the collected t_{promise}-values. Groups of promises arriving in close succession can hint at a user currently calculating his route and can be linked by the adversary. As a simple protection mechanism, navigation devices can introduce jitter between promise publications and decouple the order in which promises are published from the order in which they will be fulfilled.

An open question, however, is the combination of linking based on t_{promise} with linking using $\text{trip}_{\text{resourceID}}$. It is unclear whether such an approach can significantly improve the tracking success.

In general, the difficulty and impact of linking collected data items greatly depends on the practical deployment conditions of the proposed system. Given a sparse deployment of RAs and a large user population, it is expected that trips will be difficult to reconstruct. Additionally, a variety of additional privacy improvements are possible. Depending on the scenario, jitter, revocations or promise omissions may be used more or less extensively for reducing linkability and the reconstructibility of trips at the expense of traffic prediction performance. In a practical deployment (and as a change to the assumptions used here), RAs can also be realized in such a way that they are controlled by different organizational entities. Such a setup would result in an increased undetectability of data shared with RAs.

4.5.4.3 Identification attacks based on linked data.

The main value of reconstructed trips for the adversary is that they potentially enable more powerful identification attacks than individual samples. Such identification attacks are commonly based on context knowledge, e.g., about the home and work location of users [50].

As an inherent property of cooperative route planning, the origins of trips (often homes or workplaces) are unlikely to be included in promises, as users are already there when starting to publish their plans. Additionally, in most application scenarios, users are not obliged to publish all of their plans all of the time and might choose to selectively omit publishing parts of their individual routes that might be helpful for identification. For example, the very last parts of trips might be omitted. In scenarios with a sparse RA deployment, it is furthermore very likely that only high-traffic areas will be covered by RAs. Such areas rarely correspond to final destinations and implicitly offer high anonymity due to the large number of passing users.

4.5.5 Conclusion

An adversary colluding with the service operator is able to collect all location samples shared in the system. However, location samples from users are not easily identifiable. It is, furthermore, expected that long-distance trips taken by users, which can be used in identification efforts, are not easily reconstructible by the service operator. Additional investigations, e.g., using traffic simulations, are required to confirm this expectation.

A number of recommendations for practical deployments were stated that can further hinder adversaries from obtaining long-distance trips and identifiable location samples. These include randomizing the times at which promises are published and selectively omitting some promises (e.g., promises related to trip destinations, as trip destination are often used in correlation attacks). Distributing the control over RAs across different organizations can be considered as well, for making it more difficult for an adversary to obtain all location samples shared in the system.

4.6 Abuse resistance analysis

This section investigates possible attacks on the functionality of the proposed system. The system model from Section 4.5.1 is reused. However, a different adversary model is used, as it was previously assumed that an adversary colluding with the service operator has no interest in disturbing the service's utility.

Upon describing the assumed adversary model, different classes of attacks are introduced and their effectiveness discussed. These classes are:

- simple lying
- sybil promises
- RA hijacking
- denial-of-service

4.6.1 Adversary model

The adversary model used here is based on attackers that want to either disturb the system or exploit it for their own benefit (e.g., use it to redirect traffic). Due to these goals, it is assumed that the organization maintaining the PB, PA and the RAs has no interest in colluding with such adversaries. Specifically, it is assumed, for the presented analysis, that an adversary can control only user entities. A strong attacker might, however, be able to control multiple user entities, e.g., by registering under multiple fake or stolen identities. An especially strong attacker might additionally have gained control over one or more RAs, respectively have compromised their RACs. Compromising the PA, respectively capturing the PAC or a currently valid CIC, is assumed to be impossible for any adversary considered here. Such a compromise is significantly more difficult to achieve in practice than compromising a (potentially physically isolated) RA.

4.6.2 Simple lying

A group of malicious users (respectively a strong attacker controlling multiple user identities) might want to influence traffic by publishing false promises. For example, they might collude to publish multiple identical promises concerning a specific resource, thus making it appear overburdened and causing other traffic participants to avoid it. The influence a group of malicious users can exert on the global PB state is bounded by the number of PCs available to them. It is expected that even if they can place a number of false promises before their PC pool is depleted, this will not influence routing decisions significantly. Users attempting such schemes will also run out of PCs quickly, being unable to redeem the promise tokens they received. They might request new PCs, but a non-compromised PA will blacklist them at some point, thus effectively banning them from the system and denying them the possibility to cause more mischief.

The threat stemming from large groups of user identities under malicious control is one of the main reasons for reissuing PCs only upon the verification of promise fulfilment. Simpler approaches, e.g., with PCs valid for only one day and automatically reissued afterwards, would allow such groups to disturb traffic on a continual basis.

4.6.3 Sybil promises

In the *sybil promises* attack, an attacker publishes multiple identical promises, i.e., for the same resource and time frame. In contrast to regular lying, he is honest about the promise and just cheating by anonymously publishing it multiple times, thus reserving more resources than he needs. The benefit for the attacker is that other traffic participants will perceive the resources along the attacker's route as more crowded, thus potentially avoiding them and granting the attacker a road with less traffic. The effectiveness of sybil promises in comparison to regular lying depends on the possibility of realizing a sybil attack on the promise fulfillment mechanism. In other words, the possibility to trick a RA into mistaking a vehicle in its resource area for multiple vehicles. If this is feasible for a malicious user, he might be able to fulfill multiple identical promises simultaneously, effectively avoiding any loss of PCs in the process. Reliable techniques for identifying and counteracting sybil attacks in radio-based communication systems exist and have been evaluated for vehicular networking scenarios. See, for example, [49] for techniques based on radio-based position verification and [120] for an approach based on statistic signal strength distribution analysis. With such sybil protection mechanisms in place at RAs, the simultaneous fulfillment of multiple identical promises becomes infeasible.

Position-verification and similar countermeasures can also be used against wormhole attacks, where a user pretends to be at a given resource by using a proxy device physically placed within the resource region. Specifically, if a radio source doesn't exhibit the same mobility signature as other vehicles on the road, it can be blocked from attempting to fulfill promises.

4.6.4 RA hijacking

If an attacker has gained control over a RA (e.g., by physically compromising it) and has learned its RAC, the implications are more severe. The attacker can then publish multiple identical promises for the corresponding resource, thus emulating multiple cars planning to drive through this exact spot. Since he has access to the RAC, the attacker can redeem promise tokens immediately, allowing him to publish an infinite amount of promises for this resource. In the end, he can make it appear to the PB and other traffic participants that the resource will be hopelessly overloaded, thus heavily influencing routing decisions. For this reason, care must be taken to protect RAs and RACs. For example, tamper-proof hardware can be used. In addition to that, anomaly detection mechanisms should be deployed to identify and blacklist malfunctioning RAs and compromised RACs quickly.

4.6.5 Denial-of-service

A malicious group of users might try to attack the availability of the whole system by mounting a *denial-of-service* (DoS) attack against the PA or PB. Preventing such attacks is comparable to preventing DoS attacks in many existing information and communication systems, e.g., popular websites or cloud-based navigation services. This challenge is, therefore, considered to be out of the scope of this chapter. An attacker might also mount a DoS attack on a RA, e.g., by damaging it physically or by jamming its radio interface. As a consequence, users publishing promises concerning the resource managed by that RA will not be able to get a new PC upon promise fulfillment. Malfunctioning RAs can be blacklisted by the PA. For promises published before the RA was blacklisted, the PA can reimburse affected users by regenerating PCs for the blacklisted RAs without verifying the promises' fulfillment.

Requests validated with a PC, e.g., promise requests, cause a slightly higher overhead at the PA. However, DoS attacks using such messages are infeasible given the limited amount of PCs per user. An alternative is to use promise revocations. Such DoS attacks can also be defeated easily by requiring a certain time to have passed between promise publication and promise revocation. Section 4.7 gives an overview of the performance bottlenecks of the proposed system and gives an idea on the amount of promises that need to be generated by malicious users in order for a DoS attack on the PA or the PB to be effective.

4.7 Performance evaluation

This section describes the performance evaluation of key building blocks of the proposed system. Note that a performance evaluation of a complete cooperative route planning system, assessing also the impact of the approach on actual traffic flow, is beyond the scope of this chapter.

Only the two major expected performance bottlenecks for the proposed system are considered here:

1. The *computation overhead* introduced by the extensive use of cryptographic primitives.

2. The additional latency introduced by the use of different *anonymous secure communication channels* for each promise publication and revocation.

The overhead of mechanisms common to other systems (e.g., the distribution of traffic state via the PB) is not discussed here, as these mechanisms are already well studied and either irreplaceable for realizing cooperative route planning or negligible in terms of their impact on performance. Similarly, the communication overhead of the proposed system is also not evaluated. The small (easily below 1 kB), infrequent messages used in the system are unlikely to cause neither an overburdening of the short-range radio link to RAs nor significant costs when communicating over cellular network links with the PA.

Concerning the communication latencies of interactions between users and RAs and the associated requirements on region length and RA reachability, no explicit evaluation was conducted either. Assuming message sizes of below 1 kB, less than 2 kB must be transferred in total for cashing in a promise token at a RA. With a data rate of 16 kbit/s (realistically achievable using, for example, the IEEE 802.11p standard for short-range communication [52]), this implies that the whole operation (request and response) can be completed in less than a second. In an urban environment with a maximum speed of 20 m/s (72 km/h), it is therefore enough for RAs to be able to communicate with users in a 20 m stretch of road. Analogously, on a highway setting with speeds up to 50 m/s (180 km/h), 50 m will likely be enough.

4.7.1 Computation overhead

Key steps from the mechanisms described in Section 4.4.4 were implemented that rely heavily on the use on cryptographic primitives. Specifically, the following operations were implemented:

- generating and blinding a coin ID (user)
- signing a blinded coin ID (PA)
- unblinding a blinded coin ID signature (user)
- validating a coin ID signature (user/PA)
- signing a blinded coin ID and generating a promise token with it (PA)
- validating a promise token signature (user/RA/PA)
- cashing in a promise token, decrypting the coin ID signature contained in it (RA)

The implementation is based on RSA and RSA blind signatures as described in Section 4.4.6. Key sizes of 1024 as well as 2048 bit were evaluated. All user operations were evaluated on a low-end Android smartphone (ARM-based CPU at 600 MHz) to model the use of a device with low computational resources. The remaining operations were evaluated on a notebook computer (Intel i7 CPU at 2.60 GHz per core, the evaluation was single-threaded). All operations were repeated 100 times with different cryptographic keys and their average execution time was logged. The power consumption at the user side was not evaluated. User side operations will likely be performed inside a moving vehicle, where the energy consumption of a smartphone-class device can be neglected.

Operation	Time (ms)	
	1024-bit RSA	2048-bit RSA
signing blinded coinID (PA)	0.83	5.57
validating promise coin (PA)	0.1	0.14
generating promise token (PA)	2.03	11.31
validating promise token (RA/PA)	0.03	0.18
cashing in promise token (RA/PA)	0.94	5.49
generating blinded coinID (user)	2.15	5.59
unblinding coinID signature (user)	6.22	23.13
validating promise coin (user)	1.27	4.56
validating promise token (user)	1.38	4.58

Table 4.2 Computation times with implementation based on RSA.

The results of the measurements are shown in Table 4.2. It can be seen that for 1024-bit RSA, all user operations require less than 7 ms and for 2048-bit RSA, less than 24 ms to complete on the chosen hardware. Operations that need to be performed by RAs require less than 1 ms for 1024-bit RSA and less than 6 ms for 2048-bit RSA. For the PA, the most expensive operation is the generation of a promise token. The generation of a promise token involves the computation of two cryptographic signatures - one for the new PC and one for the promise token itself. Hence, and because in the RSA cryptosystem signing is more expensive than signature verification, the average computation time here reaches 2.03 ms for 1024-bit RSA and 11.31 ms for 2048-bit RSA. Even for 2048-bit RSA, a PA will still be able to process more than 88 promise requests per second and CPU core. For perspective, given a population of 1 million simultaneously active users, an average deployment density of one RA per 10 km and an average vehicular movement speed of 100 km/h (so that an average of 10 promises will be made per hour), an average of about 2800 promises are likely to arrive at the PA per second. Thus, with the presented implementation and key lengths of 2048 bit, computational resources equivalent to 32 CPU cores (one high-end server) will need to be deployed by the service operator. This requirement can be reduced by several orders of magnitude by employing specialized cryptographic hardware. Also, more efficient blind signature schemes exist (e.g., [90]), whose evaluation exceeds the scope of this chapter.

4.7.2 Anonymous secure communication channel latency

The unlinkability of user actions to communication metadata like IP addresses is an important prerequisite for the privacy assurances offered by the proposed system. Additionally, communication channels should be secure from end to end, to prevent attackers from disturbing the functionality of the system. These requirements can potentially lead to a significant increase in communication latency, both for setting

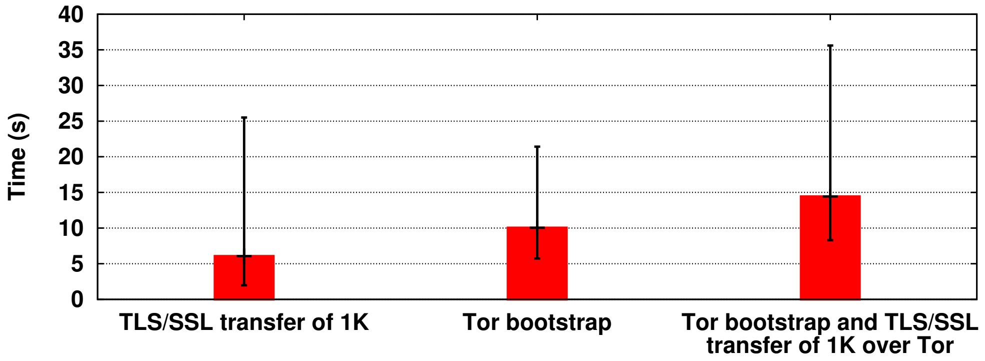


Figure 4.5 Secure and anonymous communication channel latency.

up a communication channel and for sending data over it. For deriving an estimate for the scale of these latencies, the approximate communication latency of user-PA interactions via the anonymous secure communication channel described in Section 4.4.4.1 was evaluated. An evaluation scenario was constructed to measure the time required for establishing an anonymous secure communication channel over a cellular network link and transferring 1 kilobyte of data over this channel. The 1 kilobyte data transfer was chosen for deriving an estimate for the latency involved in a common user-PA operation like the publishing of a promise. HTTPS (HTTP on top of TLS) was used to download a 1 kilobyte file to model the latency of setting up a secure TLS connection and transferring 1 kilobyte of data over it. The following was measured:

- the time for performing this file transfer without Tor
- the time for bootstrapping a Tor circuit
- the compound time for bootstrapping a Tor circuit and performing the file transfer over the Tor connection.

The measurements were performed on 7 different days in February 2014 during different times of day. In total, each measurement was repeated 70 times (10 measurements on each measurement day). All measurements were performed using a cellular network interface communicating via the EDGE standard (3G).

The measurement results are presented in Figure 4.5. The figure depicts the median of all measured times together with the respective minimum and maximum times measured during evaluation. According to the measurements, establishing an anonymous and secure channel using an EDGE-based cellular network link and performing an operation like a promise publication over it takes between 10 and 37 seconds, with a median duration of 15 seconds. Compared to the baseline latency in the evaluation scenario - the data transfer over a secure but not anonymous communication channel (2-26 seconds, median of 6 seconds) - this is a median increase of only about 9 seconds. In any case, all measured latency values are, by a large margin, acceptable for a cooperative route planning application. Route planning is typically done well in advance and operates on a timescale of tens of minutes instead of on a timescale of seconds.

4.8 Conclusion

This chapter introduces a system for sharing planned routes in a cooperative route planning context that enables the anonymous publication of plans while offering protection against abuse. According to the author's knowledge, this is the first work to consider the problem of privacy and abuse-prevention in cooperative route planning systems where participants publish information about their planned routes. Plans are published anonymously as a series of promises concerning segments of the planned route. Abuse resistance is realized by requiring the use of blindly signed *promise coins* for making each promise. New promise coins are issued upon promise fulfillment. Thus, honest users retain their right to participate in the system while malicious users get banned quickly. Promise coins are always signed blindly and thus not identifiable or linkable. Consequently, and as confirmed by analysis, promises are anonymous as well and not easily linked to trips. Through performance measurements of the involved cryptographic operations and latency measurements of the employed anonymous communication channel, it was furthermore shown that the proposed privacy-preserving cooperative route-planning system is not limited by significant performance bottlenecks and, therefore, practically feasible.

5. Privacy-preserving long-distance geocast

The availability of Internet access in vehicles offers a variety of new opportunities to assist road users. Examples include the exchange of current traffic information or the localization of free parking spaces. Long-distance communication (i.e., communication over more than a few hundred meters) is also important for vehicular cloud applications like *Pics-on-Wheels* [48], where vehicles act as service providers to which location-based service requests need to be propagated. All of these applications can be realized using an efficient *geocast* service, and in some cases even depend on one. Geocast is a communication paradigm enabling the addressing of users based on their geographic location. The realization of a cost-efficient long-distance geocast service that effectively protects the location privacy of users would allow for a variety of smart traffic applications to be realized in a privacy-preserving manner.

If geocast, respectively services depending on geocast functionality, are realized in a centralized manner, it is impossible to hide (potentially privacy-relevant) location data from service operators without heavily deteriorating service utility. Therefore, this chapter explores *decentralized* solutions to realizing geocast, focusing on overlay-based approaches [11, 51, 56, 95] and their privacy characteristics.

The rough idea behind overlay-based geocast services is the creation of a logical *overlay network* on top of cellular communication technologies and the *Internet Protocol* (IP). In the overlay network, nodes propagate their location to other participating nodes and use this information for choosing overlay neighbors and forwarding messages. Thus, neither a central entity nor additional (potentially expensive) infrastructure support is necessary. With *OverDrive* [56], this approach was specifically adapted to smart traffic scenarios. The evaluation of OverDrive showed [56], from a performance standpoint, its viability as an alternative to more traditional, centralized approaches. However, an in-depth consideration of the privacy characteristics of OverDrive and similar systems is lacking. Geocast in general is discussed in Section 5.1, focusing on applications and possible realizations. OverDrive is then introduced in detail in Section 5.2.

In Section 5.3, a privacy analysis of OverDrive is conducted. As a result of the analysis, active attacks are proposed that are specific to the geocast overlay scenario. One considered attack goal is the establishment of a precise *global view* over all users and their positions, thus eliminating the privacy benefit of decentralization. Another is the *individual surveillance* of a single user, by leveraging characteristics of the overlay maintenance logic and neighbor selection.

In Sections 5.4.1 and 5.4.2, respectively, techniques for addressing discovered vulnerabilities and improving the location privacy offered by OverDrive are proposed. Through distance-based location obfuscation and mechanisms for detecting location spoofing attempts, *data locality* is realized - precise location data is shared only with participants in the physical vicinity. Both the discovered attacks and the proposed enhancements were integrated with the existing OverDrive implementation and evaluation environment [56] based on the overlay simulation framework *OverSim* [3]. Details concerning the implementation are found in Section 5.5.

The impact of the discovered attacks was evaluated, showing that they are relevant to the original OverDrive design and effectively prevented by the protection mechanisms developed in this thesis. Simulation results demonstrate that the proposed extensions render both the large-scale surveillance and the targeted tracking of OverDrive users infeasible for adversaries controlling hundreds of overlay nodes. Still, OverDrive continues to achieve high delivery ratios in mobile environments while communication overhead is kept low. The evaluation setup and results are described in Section 5.6. In Section 5.7, the chapter concludes with a short summary of the contributions and results.

This chapter is based on [40, 42, 43] as well as, to a lesser extent, [56].

5.1 Long-distance geocast

Geocast (originally proposed in [88]) is a communication abstraction based on the addressing of points or regions in geographic space. In contrast to, for example, communication based on the *Internet Protocol* (IP), entities reachable via geocast receive messages solely based on their current location, so that no explicit addresses or unique identifiers are necessary. A *geocast service* is defined here as a communication system that allows each participant to send messages to other participants based on their location. This includes both forwarding a message to a *single participant* and *flooding a message to all participants* in a given area, for requesting or distributing information. A simple application example for such a service in a vehicular traffic scenario is the sending of an information query to all vehicles driving on a certain road.

Throughout this chapter, the focus is exclusively on long-distance geocast services. In contrast to geocast systems based on multi-hop ad hoc networks between vehicles (that have been shown not to scale to significant distances [53]), long-distance geocast places no restriction on the geographic distance between message origin and destination.

In the following, application scenarios for long-distance geocast in the context of smart traffic are discussed. Following that, an overview of existing approaches for realizing geocast is given.

5.1.1 Application scenarios

Geocast enables the realization of various cooperative services for smart traffic, i.e., services that are based on users cooperating on the road (cf. Section 2.1). *Traffic monitoring*, where users exchange traffic information to improve overall traffic flow, is a popular example. Some additional example applications are touched upon in the following, namely *parking space discovery*, *social interactions* and *vehicular clouds*.

5.1.1.1 Traffic monitoring

Chapter 4 of this thesis discusses *cooperative route planning*, i.e., how traffic participants can coordinate plans for anticipating future traffic. *Traffic monitoring* is a simpler approach where users exchange *current* traffic information for adapting to congestions and other traffic events. Using current, fine-granular information about traffic, faster routes can be found and the overall traffic flow can be improved. Traffic monitoring is a valuable complement to cooperative route planning, in that it enables individual plans, and thus the global traffic flow, to be quickly adapted to changing traffic conditions.

A large number of traffic monitoring systems has been proposed that leverage traffic participants as sources of current traffic information. The predominant approach, if the deployment of dedicated infrastructure is not an option, is based on a centralized setup and the continuous collection of self-reported location and speed data from participants. This is also the approach used by deployed commercial systems like *Google Maps* and *Waze*.

Geocast-based traffic monitoring follows a different approach - traffic data is shared in an on-demand manner directly between users. For example, a user of a geocast-based traffic monitoring system can send an information query to a road segment lying ahead of him. Vehicles residing in the target area will receive the query and can answer the requester directly, informing him about their own local view on traffic. In addition to requests, traffic warnings and messages of general interest can be delivered to participants in a given area, e.g., a section of road that was passed previously.

5.1.1.2 Parking space discovery

Tightly related to optimizing the flow of traffic is the challenge of reducing the amount of unnecessary traffic. Cruising for parking spaces was found to be a significant contributor to the latter, accounting for a non-negligible percentage of total city traffic [103][1]. In addition to saving the time of individuals, shortening parking space search times would thus also reduce traffic in general, benefiting the environment and improving quality of life. The parking space discovery problem might become even more relevant with the growing popularity of electric vehicles that need to be charged for extended periods of time, thus blocking possibly scarce charging stations.

A geocast-based parking space discovery system can support both the distribution of notifications about freed up spots (initiated, e.g., by a vehicle leaving a spot) and the delivery of search requests to vehicles and infrastructure elements in a given destination area. Recipients of such messages can answer if they are aware (via sensors or human input) of any free spots. In this way, no investments in dedicated infrastructure are necessary for deploying an effective parking space discovery system.

5.1.1.3 Social interactions

Geocast-based traffic monitoring and parking space discovery can easily be extended to support *social interactions* as well. Drivers can form complex queries that are answered by actual humans in the target region. Examples can include "Have you noted deer on the road?" and even "Watch out, there is a crazy driver coming that way!". The popular demand for such social interaction on the road is evident, considering popular social navigation systems like Waze and the CB radio-based communication habits of truck drivers and other professionals.

5.1.1.4 Vehicular clouds

Vehicular clouds [116] are a recent paradigm in the context of vehicular interconnection. Here, vehicles act as service providers, offering services based on available sensors as well as computational and storage resources. Together, participating vehicles form a cloud as in conventional cloud computing.

A representative example for such vehicular cloud services is the *Pics-on-wheels* system by Gerla et al. [48]. Users of the system can request current pictures from a given location, e.g., their own house. The request is forwarded to vehicles close to the target location, who can decide to accept the task and automatically take pictures using their on-board camera. Participants are compensated for completing such tasks and might thus even consider making a detour for completing them.

In this particular example application, the delivery of requests is basically a geocast service. However, in [48], it is realized using a central server, so that participating vehicles need to continuously report their current locations to the service operator.

5.1.2 Existing approaches

The geocast paradigm was originally proposed by Navas and Imielinski in [88]. It has been studied extensively, with a focus on short-range communication and ad

[1] To give an extreme example, the study cited in [103] comes to the conclusion that in 1977, cruising for parking spaces accounted for 74% of the total traffic in the German city of Freiburg.

hoc networking [75]. However, one- and few-hop communication based on short-range radio is insufficient for realizing applications depending on long-distance communication like the ones discussed in the previous section [53]. For realizing these types of services, the existence of dedicated infrastructure support [88] or a trusted service provider (e.g., as in [48]) is usually assumed.

Using locally deployed infrastructure and short-range radio broadcasts (i.e., following an approach like in the original geocast proposal [88]) highly reliable geocast might be achievable. While beneficial in terms of privacy - geocast messages will simply be broadcast in target regions so that no location data needs to be collected at all, this approach will likely require a high investment for deployment and maintenance. For that reason, it is not considered further here.

Multiple approaches exist for enhancing the privacy guarantees of centralized systems involving the processing of location data (cf. Section 2.3). However, no approaches were known to the author that realize geocast in a centralized manner while ensuring that the service operator is unaware of users' locations. The realization of such an approach is likely impossible; information about the current location of potential recipients is required by the service operator for determining the correct recipients of geocast messages[2]. Obfuscation can be applied when reporting locations. However, this would lead to incorrect messages deliveries (false positives), deteriorating service utility.

Decentralized, overlay-based *geocast* services [11, 51, 56, 95] are a more recent development with significant potential for resolving the inherent drawbacks of centralized smart traffic systems while not requiring any dedicated infrastructure. Roughly, the idea is the creation of a logical *overlay network* on top of IP. In the overlay network, nodes propagate their location to other participating nodes and use this information for choosing overlay neighbors and forwarding messages. Thus, neither a central entity nor additional infrastructure support is necessary.

Several proposals for such geocast overlays exist. In *GeoKad* [95], each node partitions the geographic space in logical concentric rings around its own position. A constant number of nodes from each ring are stored as neighbors. Ring radii increase linearly with the ring index. GeoKad is explicitly designed towards supporting moving nodes: each node sends location updates to its neighbors whenever its position becomes significantly different than the one advertised last. *Geodemlia* [51] also uses rings to divide up the geographic space. However, the radii of its rings increase exponentially instead of linearly. Furthermore, each ring is divided into subdivisions to create a more geographically balanced neighborhood structure. Both Geodemlia and GeoKad use an iterative routing approach inspired by the key-based routing overlay Kademlia [78], even though iterative routing is evidentially slower than recursive methods [55].

[2] Note that approaches like pseudonymization might help alleviate the raised concerns, but would still offer insufficient protection in the face of deanonymization attacks based on location samples collected over an extended period of time (cf. Section 2.2.4).

With *OverDrive* [56], the geocast overlay approach was specifically adapted to smart traffic scenarios, incorporating movement speed into the neighborhood maintenance logic and basing protocol parameters on performance measurements in actual traffic scenarios. The evaluation of OverDrive showed it to be a viable replacement for traditional centralized systems, yielding high overall performance despite the lack of centralized coordination [56]. OverDrive forms the basis for the privacy-preserving long-distance geocast service proposed in this chapter. Due to this, Section 5.2 is dedicated to introducing the protocol in detail.

The idea of tailoring the overlay protocol mechanics to the specific features of the vehicular traffic scenario was further pursued in [11]. The work focuses mostly on performance improvements to previous design and does not consider privacy issues.

Overlay-based geocast systems are interesting in terms of privacy preservation, as no central entity exists that can collect location data. However, none of the introduced systems have previously been analyzed and evaluated in terms of their privacy characteristics. Attacks on the location privacy of their users are, thus, potentially possible.

Privacy challenges have been considered in the context of vehicular ad hoc networking. A central focus here has been on the usage and the mixing of pseudonyms [37, 92]. However, due to their focus on short-range communication, existing solutions from this domain are insufficient for ensuring location privacy in long-distance geocast services. The same is true for generic location privacy techniques designed for systems controlled by singular service operators. An in-depth discussion of such approaches, as well as a discussion of pseudonymization in the context of smart traffic, can be found in Chapter 2.

5.2 OverDrive

This section describes OverDrive, a decentralized, overlay-based geocast service that is adapted to smart traffic scenarios. Information requests for points in geographic space are routed directly via traffic participants until they reach a node in the proximity of that point. OverDrive enables effective long-distance geocast while eliminating the need for a central data sink: location data is shared only with a small and dynamically changing set of other participants. As originally proposed in [56], OverDrive is based around two concepts:

- An overlay neighborhood structure based on a partitioning of geographic space into concentric *rings*, as well as mechanisms for maintaining this structure.
- A routing mechanism for forwarding messages to nodes in a desired geographic area. Messages are forwarded using connections from the overlay neighborhood structure.

An overview of these concepts and their realization in the original OverDrive design is given in the following. Additionally, the use of pseudonyms in OverDrive is discussed.

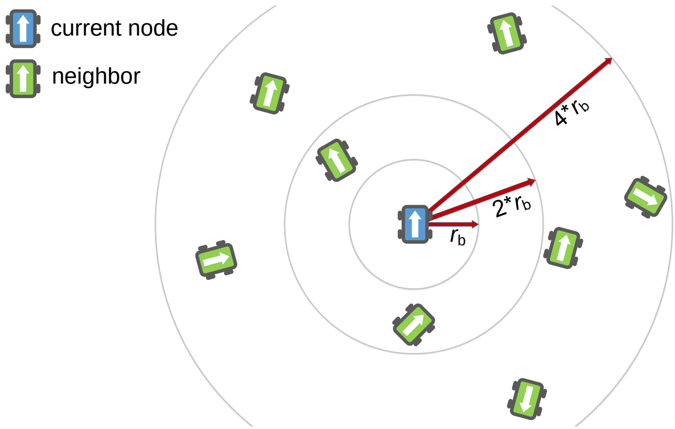


Figure 5.1 Neighborhood structure.

5.2.1 Neighborhood structure and overlay maintenance

This section describes the structure of OverDrive's overlay topology as well as the operations necessary for maintaining it.

5.2.1.1 Neighborhood structure

The overlay neighborhood structure is based on mapping neighbors to a set of k concentric rings around the position of the current node. The radii of these rings start with the *base ring radius* r_b and grow exponentially with the distance from the common center. Based on the distance from the current node, neighbors fall into one of the different rings. This concept is illustrated in Figure 5.1.

Each ring has an index, starting with 0, and the index i for a neighbor with distance d can be determined as follows:

$$i = \begin{cases} 0 & \text{if } d < r_b \\ j & \text{if } r_b \cdot 2^{j-1} < d \leq r_b \cdot 2^j \text{ for } 0 < j \leq k \\ k & \text{if } r_b \cdot 2^k < d \end{cases}$$

The amount of neighbors present in each ring is determined by the parameters n_{des} and n_{max}. The parameter n_{des} indicates the *desired* number of neighbors a node wishes to keep in each ring. As long as n_{des} hasn't been reached, the node will actively search for more nodes, a process described in the next section. The parameter n_{max} indicates the *maximum* number of neighbors that are kept for each ring. For a ring that has a neighbor count between n_{des} and n_{max}, new neighbor connections will be accepted but no active search for new neighbors will be conducted.

5.2.1.2 Discovery of new neighbors

Each node periodically calculates its satisfaction with each of its rings. If, amongst other things, there are less than n_{des} neighbors in a ring, the respective node starts a search for new neighbors for this ring. The search for new neighbors for a ring works by routing a *find neighbors request* to a random point within the geographic area covered by that ring. The request is answered by the node that is closest to that point by sending a *find neighbors response* back to the requester. The find neighbors response contains descriptors of a subset of the responding node's neighbors. The subset is chosen by filling the response with nodes starting from the neighbors in its innermost ring and going outward from there.

Once the response has arrived at the originator of the request, if the current number of neighbors n for the given ring is lower than n_{des}, the node tries to add $(n_{\text{des}} - n)$ new neighbors to this ring. This is done by sending *neighbor connect requests* to $(n_{des} - n)$ of the received neighbor candidates.

Whenever a node X receives a neighbor connect request from another node Y, it needs to decide whether to accept it. If there are less than n_{max} neighbors in the appropriate ring, Y is always accepted. Neighbors are accepted by replying to the neighbor connect request with an appropriate *neighbor connect response*.

5.2.1.3 Updating location data

Since OverDrive is specifically designed for mobile nodes, keeping the information about neighbor locations up-to-date is a critical part of overlay maintenance. *Location updates* are periodically sent out to overlay neighbors. These messages include the current location of the sending node. In addition to this location, recipients store the time at which the last location update was received. Neighbor connections are always bidirectional so that all nodes that learn a node's location also share their own location with it.

5.2.2 Message routing

OverDrive uses a *recursive*, greedy routing scheme for delivering messages towards destinations in geographic space. At every hop, the message is forwarded to the neighbor that is closest to the target point in geographic space. If none of the current hop's neighbors is closer to the target than itself, it assumes the responsibility for the message and processes it. This routing mechanism is used for both *geographic unicast* and *geographic flooding*.

5.2.2.1 Unicast

Geographic unicast enables an application to send a payload to a node in a given target area. The overlay encapsulates the payload in a *geographic unicast message* (GUM) and routes it to the node closest to the center of the target region. On the destination node the payload is passed to the application, which may send a direct response back to the originator.

An example is given in Figure 5.2. The figure depicts a possible application for the geocast service, namely the sending of a query to a point in geographic space (e.g., a road segment) lying ahead of the requester. The query is realized as a GUM. From all of its neighbors, which are chosen based on a partitioning of geographic space into concentric rings, the requester greedily chooses the one neighbor that is closest to the destination region in terms of geographic distance. The GUM is sent to this neighbor, who then forwards it according to the same rule, sending it to the one of

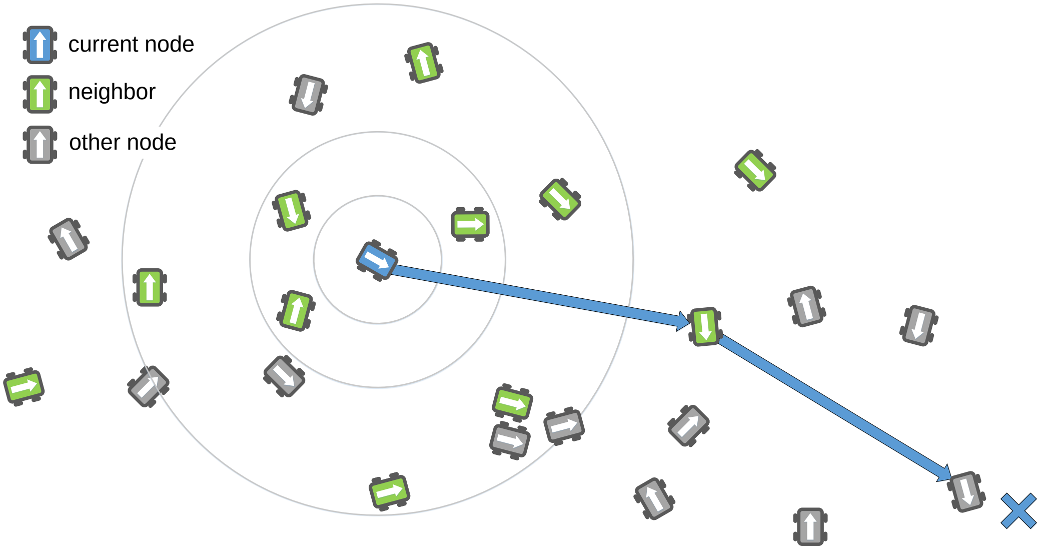


Figure 5.2 Delivery of a geographic unicast message (GUM).

its overlay neighbors that is closest to the destination region. Once the GUM arrives at a node residing in the target area, that node might, depending on the application, decide to answer the query by directly sending a response to the requester.

As in other greedy routing approaches, the routing approach used in OverDrive is theoretically prone to getting stuck in local minima. In other words, it cannot be guaranteed that the actual destination will be reached. No explicit measures for avoiding local minima during routing have been taken in OverDrive's design. Despite this, simulation-based evaluations in scenarios with realistic node movement have demonstrated a high percentage of correct GUM deliveries [56].

5.2.2.2 Flooding

Geographic flooding allows applications to broadcast a payload to all nodes within a certain area. To reach the flooding area, a *geographic flooding message* is routed geographically towards the center of that area (using the same mechanism as for geographic unicast messages). The first node within the destination area that receives the message is called the *initial flooder*. It will send a confirmation that the flooding has started back to the sender of the flooding request. It will then start the actual flooding process, recursively forwarding the request to other peers within the destination area.

Geographic flooding is largely ignored in the remainder of this chapter, as, except for the final flooding step, it is realized in the same way as geographic unicast.

5.2.3 Use of pseudonyms

As in other vehicular networking approaches (e.g., [92]), OverDrive nodes use *pseudonym certificates* (*pseudonyms* for short) to protect the identity of drivers while preventing *sybil attacks* (i.e., bounding the number of pseudonyms per user), enabling the revocation of participation rights and ensuring the authenticity of messages (e.g., through cryptographic signatures). Pseudonymization can be realized using a trusted certificate authority or a decentralized system like *BitNym* (cf. Chapter 6). It is also possible to reuse pseudonym certificates from other domains. For example, certificates used in short-range vehicular networking, as standardized by the ETSI[3] and the IEEE[4], can be used. However, following properties must be realized by the employed pseudonymization approach:

1. Pseudonyms must be unlinkable to user identities and other pseudonyms of the same user.

2. It must be ensured that sybil attacks are not possible. Only a bounden number of simultaneously active pseudonyms must be allowed per user.

3. Mechanisms must be provided for explicitly linking messages to pseudonyms and it must be ensured that only the holder of a given pseudonym can realize such links with that pseudonym. For example, pseudonym representations can include cryptographic public keys usable for verifying signed messages.

4. Pseudonym holders must be able to unlinkably change their pseudonyms.

To the best of the author's knowledge, the pseudonymization approach proposed in Chapter 6 of this thesis is currently the only one to fulfill all of these requirements without requiring the existence of a single trusted entity (and, therefore, a weaker adversary model).

The last requirement, that pseudonyms must be changeable, is required for avoiding identification attacks based on recorded trips linkable to a single pseudonymous user. In OverDrive, new pseudonyms are chosen at the beginning of each trip. Additionally, the option of changing pseudonyms during trips is supported. In order to reduce the linkability of pseudonyms based on observed times and locations before and after a change, pseudonym changes must be coordinated with other participants (cf. Section 2.3.4). OverDrive nodes communicate with each other over IP or PTP (cf. Section 2.6). Therefore, their communication addresses must be changed as well during pseudonym changes in order for the unlinkability between pseudonyms to be ensured.

If OverDrive nodes retain the same set of neighbors after a pseudonym change, their neighbors might be able to link the new pseudonym to the old one based on timing, i.e., the fact that the new pseudonym appears just after the old pseudonym disappeared. However, pseudonym changes should not significantly impact performance, so that a discarding of multiple existing neighbor relationships should be avoided.

Based on the preceding considerations, an approach for the changing of pseudonyms in OverDrive is proposed that consists of the following sequential steps:

[3] http://www.etsi.org/index.php/technologies-clusters/technologies/intelligent-transport

[4] http://standards.ieee.org/develop/wg/1609_WG.html

1. Whenever a pseudonym change is desired, OverDrive nodes attempt to form *mix groups* with nearby neighbors.
2. Nodes in a mix group exchange their complete neighbor sets, i.e., the communication addresses, pseudonyms and known locations of all of their individual neighbors.
3. All nodes in a mix group cease the use of their old pseudonyms and all communication with other nodes.
4. Each node independently chooses (or obtains) a new pseudonym for itself, as well as a new communication address (IP address or PTP identifier).
5. Each node independently chooses a random subset of the set of all neighbors shared in the mix group.
6. Using its new pseudonym and communication address, each node forms new neighbor relationships with all nodes selected in the previous step.
7. Regular operation is resumed.

In essence, this approach is an adaptation of the *SwingSwap* approach [71].

With pseudonyms for every participant, a central challenge for an adversary interested in large-scale surveillance or targeted tracking becomes the linking of pseudonyms to real-world identities, i.e., the *breaking* of pseudonyms.

5.3 Privacy analysis

A privacy analysis of OverDrive is conducted in the following, based on the approach outlined in Chapter 3. While focused on OverDrive, the results of this analysis are applicable to other geocast overlay approaches like Geodemlia and GeoKad.

An adversary model is defined first. The analysis is then performed in two parts separated based on the addressed privacy threats:

1. Detection and disclosure.
2. Linking and identification.

5.3.1 Adversary model

In accordance with Section 3.2, an adversary is assumed that is not colluding with cellular network operators[5]. The main goal of the adversary is the collection of identifiable location samples. Subgoals towards this end and metrics associated with these subgoals are introduced in subsequent sections.

Following assumptions are also made that are specific to the geocast overlay context:

[5] In currently deployed cellular networks, network operators can always determine both the identity and the location (with high precision) of connected users.

- The adversary is able to control multiple *attacker nodes* in the overlay network, i.e. virtual identities within the OverDrive overlay. Sybil attacks are impossible and the maximum number of attacker nodes is bounded to up to 1% of the remaining node population[6].

- Attacker nodes are able to lie about their position. Apart from that, they run the OverDrive protocol like regular nodes.

- Attacker nodes cannot arbitrarily influence their physical location. They are either static or follow the mobility behavior of a regular traffic participant (which can't be influenced by the adversary).

- The adversary is unable to identify nodes using communication metadata or by breaking the employed pseudonymization scheme.

- Unless directed at an attacker node, the adversary is unable to detect or eavesdrop on communication between nodes[7].

- All messages associated with a pseudonym (i.e., a message sent by a given node) are cryptographically signed so that their integrity and authenticity is maintained across forwarding hops.

5.3.2 Detection and disclosure

This section considers the following questions:

1. What data does the adversary gain from *passive observation* during normal operation?

2. What *actions* are possible for him and are expected to have a significant impact?

3. In what way can the previously identified actions be leveraged by the adversary for *increasing* his *knowledge*?

5.3.2.1 Knowledge from passive observation

During normal operation, each OverDrive node maintains overlay connections to a number of other nodes from which it periodically receives location updates. Less frequently, it also receives find neighbors requests and neighbor connect requests. With a similarly low frequency, it will send out find neighbors requests itself and receive find neighbors responses. In addition to the message type, which can be ignored here, all of these messages must include one or more pseudonymized node locations, i.e., $(\text{pseudonym}, \text{location})$-tuples. So, during normal operation, each OverDrive node is able to collect samples of the form $(\text{pseudonym}, \text{location}, \text{time})$

[6] In practice, this is achieved through the use of a sybil-resistant pseudonymization approach (cf. Section 5.2.3).

[7] Otherwise, he might be able to infer all neighbor relationships in the overlay network and use this information as an additional data source when inferring locations through correlation.

from multiple other nodes. The majority of these samples will be collected from neighbors via periodic location updates. As the adversary may control multiple attacker nodes, he is able to collect (pseudonym, location, time)-samples from multiple vantage points in the overlay network.

Depending on the specific scenario (e.g., what kind of smart traffic application is using the geocast overlay), nodes will also frequently receive GUMs. GUMs include a destination region as well as a pseudonym (e.g., a communication address) for directing answers to. However, the relation between the destination region and the location of the pseudonym holder is not obvious. Additionally, it can easily be obfuscated further, e.g., by using different pseudonyms for initiating GUMs than these used for maintaining the geocast overlay and by choosing the first hop for a GUM in a randomized fashion. The quantity, destinations and content of GUMs is, also, scenario-specific and, therefore, difficult to evaluate in a general fashion. For all of these reasons, GUMs are not considered further here.

A different reasoning applies for GUM responses. Here, the answering node discloses its proximity to the destination region. Depending on the scenario, nodes may send out many GUMs during regular operation. In this way, they will likely receive many GUM responses and, thus, multiple additional (pseudonym, location, time)-samples extractable from them. Again, however, answering nodes are not required to use the same pseudonym for answering GUM queries that they use for interacting with overlay neighbors (or, for that matter, any pseudonym at all). Answering GUM queries without the use of linkable identifiers lowers the linkability of GUM responses to other data items and, consequently, the usefulness of GUM responses for the adversary. Additionally, the location data extractable from a GUM response must correspond only to the destination region of the associated GUM request, i.e., is likely of low precision.

5.3.2.2 Possible actions

The purpose of this section is to exhaustively enumerate the actions that are possible for the adversary in the considered adversary model. More specifically, this involves the actions that attacker nodes can perform, as the adversary has no significant capabilities exceeding the injection and control of such nodes. The tackled question is, therefore, what non-standard forms of behavior are possible for attacker nodes that can potentially lead to benefits for the adversary. Recall that the main goal of the adversary is the obtaining of large quantities of identifiable location samples.

As part of the adversary model, the adversary is unable to influence the physical locations of attacker nodes. OverDrive nodes are mainly characterized by their physical location and their communication behavior with other nodes. Consequently, the actions considered here are restricted to different non-standard ways in which attacker nodes can interact with regular nodes. This involves the sending and forwarding of messages that are accepted by regular nodes. Other messages or communication attempts are, in the best case for the adversary, ignored by regular nodes and is not considered further here. Messages accepted by regular nodes are:

- Location updates from neighbors. As a form of non-standard behavior, attacker nodes can fake the location included in these messages, essentially lying about their own locations. However, they must remain consistent with their reported (fake) locations in order to escape detection. Reported locations must be chosen based on the capabilities of real vehicles (e.g., maximum movement speed). Inconsistent location updates are non-repudiable. Nodes that receive such updates can immediately identify the attacker node, blacklist it locally (avoiding any further communication to it) and distribute the proof of the inconsistency to its neighbors, so that they can blacklist the node as well. In the following, it is assumed that location faking, if applied, is always consistent and based on realistic vehicular movement.

- Find neighbors requests. The sending of find neighbors requests triggers the receipt of find neighbors responses (containing location samples from several pseudonymous nodes). Additionally, received find neighbors requests can be discarded instead of forwarding or answering them.

- Find neighbors responses. Attacker nodes can respond to a find neighbors request even if they are aware of better suited nodes to forward it to. Additionally, they can prefer other attacker nodes when forming the returned set of nodes. Lastly, they can also include invalid pseudonyms or communication addresses in find neighbors responses. Since the resulting find neighbors response is of little use to the requester, this is comparable to dropping the request and not sending any response at all.

- GUMs and GUM responses. A similar analysis applies here as for find neighbors requests and responses. The explicit consideration of GUMs and GUM responses is, therefore, omitted in this analysis step.

- Neighbor connect requests. Can be sent to each newly discovered node, effectively removing the limit on accepted neighbors.

- Neighbor connect responses. All neighbor connect requests can be accepted, effectively removing the limit on accepted neighbors.

5.3.2.3 Increasing knowledge

Based on the preceding analysis, attacker nodes are left with the following classes of actions that may be used to increase the adversary's knowledge:

- Fake own locations in location updates and other messages. However, consistency must be maintained and unrealistic jumps in the reported locations are avoided.

- Send out large numbers of find neighbors requests (or, analogously, GUMs).

- Drop find neighbors requests (and, analogously, GUMs) or answer them even if they should actually be forwarded further[8].

- Choosing the nodes to include in find neighbors responses according to non-standard criteria, e.g., preferring other attacker nodes.

- Try to add all newly discovered nodes as neighbors and accept all incoming neighbor connect requests.

- Any combination of the preceding.

Based on these actions, in Section 5.3.4, two specific attacks are proposed that are applicable to the original OverDrive design and similar geocast overlay proposals. The proposed attacks do not cover all possible combinations of the identified adversary actions. In other words, additional attack vectors towards obtaining identifiable location samples using the identified actions might be possible. However, none could be determined that pose a significant and not easily mitigated threat. Additionally, preliminary simulation results demonstrated that the proposed attacks are already tremendously effective even with a modest number of attacker nodes. Independently of any other possible attacks, this necessitates their deeper investigation.

5.3.3 Linking and identification

Linking and identification attacks are not explicitly discussed in the scope of this chapter. Significant open questions exist concerning the detection and disclosure threat that forms the basis for linking and identification. If the adversary is able to collect pseudonymous location samples from nearly all participating users, the same linking and identification threats will apply like in a similar centralized system using pseudonyms. Such a setup has been widely studied in the literature (cf. Section 2.2). A central question tackled in the remainder of this chapter is the mapping of the data gathered through attacks on the OverDrive system (both the original and the improved form introduced later in this chapter) to the data gathered in similar centralized system (in which the adversary is able to collect pseudonymous location samples from nearly all participants). Additionally, one of the specific attacks discussed in the following realizes a simple identification attack for the targeted collection of location samples from a specific victim.

5.3.4 Specific attacks

In the following, two specific attacks towards obtaining identifiable location samples are discussed. They differ in their respective subgoals. The first attempts to establish a *global view* over all nodes, allowing the application of linking and identification attacks from the literature. The second aims at the surveillance of an *individual target* that the adversary can identify directly (e.g., using context knowledge). The effect of both attacks was evaluated using simulations. The evaluation setup and results is discussed in Section 5.6.

[8] Note that this is also a threat to the functionality of the geocast service. However, the focus here is on privacy characteristics, so security concerns are not considered further.

5.3.4.1 Establishment of a global view

A straightforward attack approach is the establishment of a global view comprised of pseudonymous location samples, i.e., (pseudonym, location, time)-tuples, from as many overlay nodes as possible. Given a global view, pseudonyms can be broken using known techniques (cf. Section 2.2.4). The establishment of a global view is possible using the following attack:

1. The adversary controls multiple attacker nodes that may behave like regular traffic participants. As attacker nodes can lie about their location, they can also be co-located in reality with individual node movement being simulated by the adversary.

2. The attacker nodes attempt to become overlay neighbors with as many regular overlay nodes as they can (e.g., by sending out many find neighbors requests and accepting all neighbor connect requests they receive).

3. The attacker nodes forward (through a separate channel) all location updates they receive to the adversary who combines them into one global view.

5.3.4.2 Surveillance of an individual target

A more sophisticated attack approach is the exploitation of inherent properties of the geocast system for identifying and tracking targets with far less resources. In the following, a representative attack from this class is presented. The approach assumes that the adversary targets one user about which he has context knowledge in the form of the location at which he will start his trip. Depending on the scenario, such information can be obtained easily. For example, the home address of the victim might be used.

Given such context knowledge, the adversary faces two challenges: (1) mapping the victim to an overlay node and (2) continuing to track the location of that node. For the first step, following two properties of OverDrive (which are shared by the majority of existing geocast overlays) can be leveraged:

Property 1. *For any given pair of overlay nodes, the probability of them being overlay neighbors is inversely proportional to the geographical distance between them.*

Property 2. *If a node discovers another node for the first time, the probability that this node has just joined the overlay or changed its pseudonym is inversely proportional to the geographical distance between the two nodes.*

Note that property 2 is an implication of property 1. With these properties of geocast overlays, the adversary can attempt to map a victim to an overlay node by placing attacker nodes around the start point of the victim (e.g., by instructing them to lie about their positions). The attacker nodes can then report all new nodes they discover to the adversary. Whenever a new node X is discovered in the vicinity of the victim start position, following reasoning applies:

- X is close to the victim start point → it is also close to the attacker nodes.
- X is close to the attacker nodes and seen by them for the first time → it is likely to have just joined the overlay (cf. property 2).
- X is close to the victim start point and has likely just joined the overlay → it likely belongs to the victim.

Having acquired a likely victim node, the next step for the adversary is to continue tracking it. For this, an approach specific to geocast overlays is introduced - the *follower attack*. This attack is based on property 1. Given a node which the adversary wants to track, one attacker node continuously fakes its position so as to appear in the vicinity of the victim node. Being in the vicinity of the victim node, it is very likely to remain in the victim node's neighborhood and continuously receive location updates from it.

The potential victim node can be continuously followed in a virtual fashion, i.e., without requiring any form of physical proximity. Possibly, this also leads to an increase in the certainty about the victim's identity, as the adversary can collect more location data that can be matched with context knowledge.

Summing up, by combining context knowledge, the careful placement of attacker nodes and the follower attack, an adversary might be able to successfully identify and track a real-world target at a potentially much lesser cost (in terms of required attacker nodes) than surveilling the whole overlay for creating a global view.

5.4 Location privacy enhancements

Two enhancements are now proposed that tackle the major weaknesses enabling attacks like the ones described in Section 5.3. The contributions aim at establishing *data locality*, i.e., ensuring that precise location data is only shared with entities that are physically located in the close vicinity.

First, a mechanism is presented that decreases the accuracy of the location data an adversary is able to acquire (*location obfuscation*). Second, a countermeasure is proposed against malicious nodes that fake their location data in order to receive detailed location updates from targets (*location spoofing detection*).

5.4.1 Location obfuscation

For establishing a global view on the location of all nodes, an adversary's goal is to gain as accurate location data as possible about as many nodes in the network as possible. An efficient way to defend against this attack is to avoid delivering accurate location data to the adversary. This approach does not prevent an adversary from collecting location data about many nodes but will decrease the value of the collected information, i.e., its suitability for breaking pseudonyms or determining the exact location of a given node.

5.4.1.1 General approach

The basic approach here is to decrease the accuracy of the location data shared between two nodes A and B with growing distances between them. OverDrive uses a greedy forwarding algorithm where each node forwards a message to the one of its neighbors that it estimates to be closest to the destination location of the message. Thanks to OverDrive's neighbor selection logic, it is expected that the distances between individual hops decrease with each routing step and that the distance to the destination decreases in smaller and smaller steps with each hop. Thus, the proposed enhancement is not expected to impact geographic routing performance in a significant way.

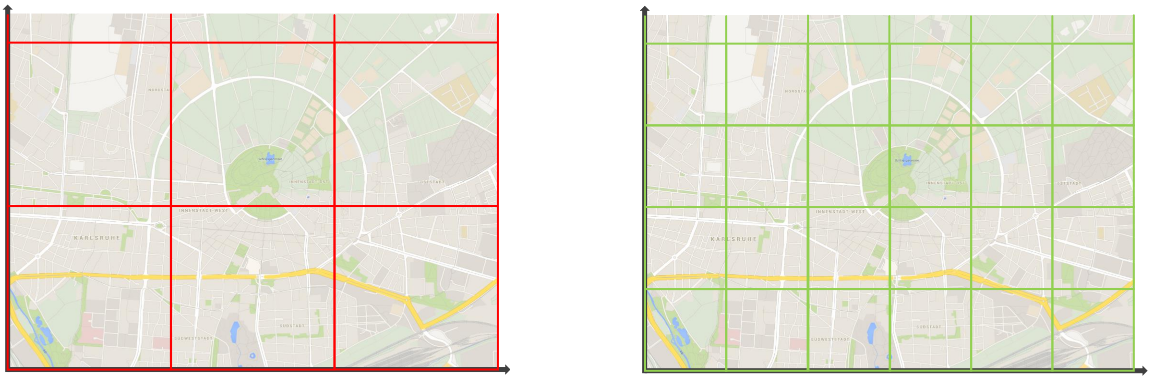
5.4.1.2 Obfuscation regions

The obfuscation approach proposed here is based on the concept of *obfuscation regions*. An obfuscation region is a quadratic geographic region with an edge length of l_{edge}. Instead of transmitting precise location data, the nodes A and B share the center position of an obfuscation region they currently reside in. In order to allow different levels of obfuscation based on the distance between two nodes, the size of the obfuscation region can be varied. For denoting the desired degree of obfuscation, the *zoom level z* is defined so that $l_{\text{edge}} = 2^z$ kilometers. The zoom level is linked to the (presumed) geographic distance to the node with which location data should be shared. More faraway nodes receive larger obfuscation regions and, thus, more heavily obfuscated location data.

5.4.1.3 Obfuscation grid

Given a zoom level and the accurate location of a node, an obfuscation region can be constructed. If each node calculates its obfuscation region by choosing a random quadratic region around its position, an adversary might break the obfuscation by intersecting multiple views collected from different nodes under the adversary's control. To avoid this kind of attack, obfuscation regions must be constructed in such a way that the information gained from combining multiple received obfuscation regions for the same location never exceeds the information contained in the obfuscation region with the lowest zoom level.

To achieve this property, the concept of an *obfuscation grid* is proposed. An obfuscation grid is a division of geographic space into disjoint squares as shown in Figure 5.3. Each of the squares represents a single quadratic obfuscation region. Every node in the overlay uses the same origin for the obfuscation grid, regardless of the used zoom level. Since $l_{\text{edge}} = 2^z$, each obfuscation region at zoom level z can be divided into four disjoint regions at zoom level $z - 1$, as shown in Figures 5.3a and 5.3b. Thus, obfuscation regions never intersect and two obfuscation regions for the same location are either identical (in case their zoom level matches) or the region with the lower zoom level is contained within the other. With this, adversaries cannot gain any additional information from combining multiple obfuscated views of the same location compared to using only the most precise view available.



(a) obfuscation grid at level z

(b) obfuscation grid at level $z-1$

Figure 5.3 Obfuscation grid.

5.4.1.4 Creating obfuscation regions

Two pieces of data are needed for calculating an obfuscation region. First, the location $L(\text{lon}_L, \text{lat}_L)$ that is to be obfuscated. Second, the zoom level z determining the size of the obfuscation region. Furthermore, an *origin point* $O(\text{lon}_O, \text{lat}_O)$ for the obfuscation grid must have been defined. As already discussed, O must be identical for all nodes. For simplicity, the origin point is defined at $(0,0)$ in geographic space, i.e. $O = (0,0)$. The zoom level is determined by the index i of the ring in which neighbor B resides. As a parameter to the system, the *downscaling factor* f_d is introduced so that $z = i - f_\text{d}$. Thus, if B resides in the i'th ring of A's neighborhood structure, A will share its location with B using an obfuscation region with edge length $l_\text{edge} = 2^{(i-d)}$ (in kilometers). By varying the value for f_d, different degrees of obfuscation can be tested (larger values for f_d decrease the level of obfuscation). To calculate the correct obfuscation region using the given data, the latitude/longitude-based location L must first be transformed into the coordinate space of the obfuscation grid, yielding the grid point $L'(x_{L'}, y_{L'})$. Based on the haversine formula for calculating approximate distances on spheres and with r denoting the earth radius, following formula applies:

$$x_{L'} = 2r * \arcsin\left(\cos\left(\text{lat}_O\right)\ \sin\left(\frac{\text{lon}_L - \text{lon}_O}{2}\right)\right)$$

$$y_{L'} = r * \left(\text{lat}_L - \text{lat}_O\right)$$

For $O = (0,0)$, we arrive at:

$$L'(x_{L'}, y_{L'}) = (r * \text{lon}_L, r * \text{lat}_L)$$

Using L', the points P_min and P_max defining opposite corners of the resulting region can now be calculated as:

$$P_{\min} = \left(2^z * \lfloor \frac{x_{L'}}{2^z} \rfloor, 2^z * \lfloor \frac{y_{L'}}{2^z} \rfloor\right)$$

$$P_{\max} = \left(2^z * \lceil \frac{x_{L'}}{2^z} \rceil, 2^z * \lceil \frac{y_{L'}}{2^z} \rceil\right)$$

$$P_{\min} = \left(2^z * \lfloor \frac{r * \mathrm{lon}_L}{2^z} \rfloor, 2^z * \lfloor \frac{r * \mathrm{lat}_L}{2^z} \rfloor\right)$$

$$P_{\max} = \left(2^z * \lceil \frac{r * \mathrm{lon}_L}{2^z} \rceil, 2^z * \lceil \frac{r * \mathrm{lat}_L}{2^z} \rceil\right)$$

Once these points are known, the center point $P(x_P, y_P)$ of the obfuscation region can be calculated as:

$$P = \left(\frac{\left(x_{P_{\min}} + x_{P_{\max}}\right)}{2}, \frac{\left(y_{P_{\min}} + y_{P_{\max}}\right)}{2}\right)$$

The last step is to transform $P(x, y)$ back into geographic coordinates. The transformation is realized by inverting the formulas used for calculating L', which yields:

$$\mathrm{lon}_P = \frac{x_P}{r} \qquad \mathrm{lat}_P = \frac{y_P}{r}$$

5.4.1.5 Determining the ring index

Whenever A wants to share location data with B, it has to determine the correct level of obfuscation to be applied to the location data. Simply calculating a ring index based on A's real position and B's reported position might lead to inconsistencies in cases where A is located close to a ring boundary. A's obfuscated position might not be within the same ring of B's neighborhood structure as A's real position. Therefore, the correct ring index of A in B's neighborhood structure is dependent on the distance d_o between B's reported position and the center of the correct obfuscation region for A. Based on d_o, and analogously to the original OverDrive design, the ring index i for a neighbor is determined as follows:

$$i = \begin{cases} 0 & \text{if } d_o < r_\mathrm{b} \\ j & \text{if } r_\mathrm{b} \cdot 2^{j-1} < d_o \leq r_\mathrm{b} \cdot 2^j \text{ for } 0 < j \leq k \\ k & \text{if } r_\mathrm{b} \cdot 2^k < d_o \end{cases}$$

In the scenario depicted in Figure 5.4, for example, A must send a location obfuscated with a zoom level for the ring with index 2 (the obfuscation region is depicted as a green box), even though it is actually residing in B's fourth ring (ring index 3).

The correct obfuscation region is determined iteratively according to Algorithm 5.1. For each ring index i starting from 0, an obfuscation region for the corresponding

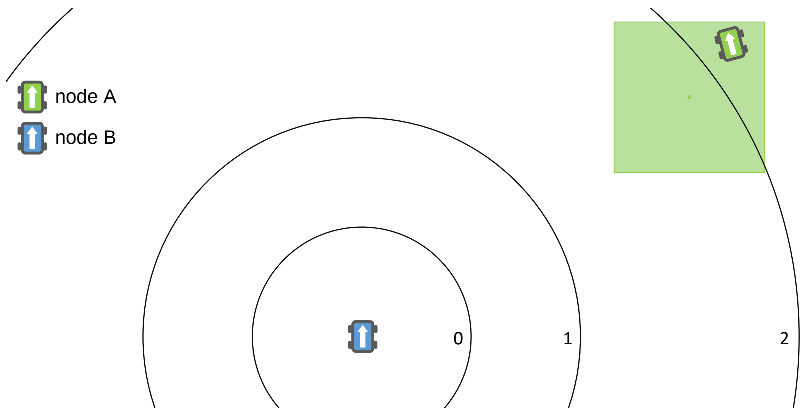


Figure 5.4 The correct ring is 2, even though the actual location is in ring 3.

zoom level z is calculated. If the center P_A of that obfuscation region lies within B's ring with index i (based on the distance d_o between P_A and B's reported location P_B), this is the correct obfuscation region. If not, the check continues with the next highest ring index $i+1$.

Algorithm 5.1: Determining the correct obfuscation region center for sharing the actual location L_A to a neighbor with reported location P_B.

```
$i = 0$;
$z =$ getZoomLevel($i$);
$P_A =$ getObfuscationRegionCenter($z, L_A$);
$d_o =$ getDistance($P_A$,$P_B$);
while $i >$ getRingIndex($d_o$) do
    $i = i + 1$;
    $z =$ getZoomLevel($i$);
    $P_A =$ getObfuscationRegionCenter($z, L_A$);
    $d_o =$ getDistance($P_A$,$P_B$);
end
return $P_A$;
```

5.4.1.6 Neighborhood structure and neighbor scoring

The general neighborhood structure concept remains the same as presented in [56] and Section 5.2. Changes include the type of information shared with neighbors: instead of precise location, bearing and speed, nodes only share the center of the correct obfuscation region. Consequently, the scoring function, used to rank neighbors within the same ring for determining whether an old neighbor should be ejected from the neighborhood in favor of a given new candidate, was simplified as well. Scores are no longer based on bearing and speed, but only on the number of neighbors in the vicinity of the scored node (fewer neighbors in the vicinity lead to higher scores). The simplified scoring function (for a neighbor A) is shown in Algorithm 5.2.

Algorithm 5.2: Calculating the score of a node A.

```
$i$ = getRingIndex($A$);
$P_A$ = getObfuscationRegionCenter($A$);
$c = 1$;
foreach $N \in$ {neighbors in ring i} do
    $P_N$ = getObfuscationRegionCenter($N$);
    if $(N \neq A) \wedge (P_N == P_A)$ then
        $c = c + 1$;
    end
end
return $1/c$;
```

5.4.2 Location spoofing detection

In the following, an approach is presented for identifying malicious nodes that spoof their location, so that adversaries need to be physically close to their victims in order to receive precise location data.

5.4.2.1 Private proximity testing

Solutions for the protection against location spoofing exist that require additional infrastructure support [15] or spot checks [96]. If the problem can be reduced to proximity checks, short-range radio beacons can be used. However, approaches based on short-range radio have a limited reach, especially in environments with many obstructions like buildings.

An alternative approach is to compare environmental features that are unpredictable and unique to a given location and time. Originally proposed in [87], *private proximity testing* (PPT) realizes such a comparison in a privacy-preserving manner. More specifically, PPT enables users to verify whether a certain other user is in their vicinity, without having to reveal their own location. PPT can be realized using the *location tag* and *location sketch* concepts proposed in [87] and [73].

A *location tag* is a set of features that are unique in space and time. The generation of a correct location tag for a location is only possible if an entity is physically present at that location. In [73], location tags are constructed from GSM broadcast traffic. By collecting *immediate assignment* (IA) messages, location tags specific to individual GSM *cells* can be constructed. Using signaling traffic from the *broadcast paging channel* (PCCH), the same is possible for GSM *location areas*, i.e., groups of multiple cells. By comparing location tags generated in this way, reliable proximity tests over distances of 10 km and more are possible.

A *location sketch* is a single value generated from a location tag using the *shingling* technique [73]. It enables the efficient comparison of location tags using *private equality testing (PET)*, i.e., verifying the equality of another party's location tag without either party needing to disclose its location or location tag. Here, the use of a synchronous PET protocol based on El Gamal encryption is proposed, as presented in [87]. Alternative forms of PET are possible. However, the novel contributions presented in this chapter are independent of any specific implementation. The focus here is on the general applicability of PPT and the integration of existing techniques into OverDrive.

5.4.2.2 Integration into OverDrive

For enabling location spoofing detection using PPT, OverDrive nodes need to continuously collect local GSM broadcast traffic - IA messages and traffic on the PCCH. From the collected data they can create location sketches proving their location in a cell (using IA traffic) and location area (using PCCH traffic). Using PET, two nodes can check if their location sketches match without having to share the actual sketches. If their IA-based sketches match, the nodes assume that they reside in the same GSM cell and are, therefore, not significantly more than 4 km apart. The value of 4 km is a conservative estimate to reduce the number of false negatives when verifying the proximity of neighbors. In practice, GSM cells can have radii of up to 35 km [102]. If only their PCCH-based sketches match, they assume to be located in the same location area and not significantly more than 10 km apart. The value of 10 km is, again, a conservative estimate (location areas are comprised of multiple cells). The parameters derived from assumed cell and LA sizes can be fine-tuned with more specific information about the used GSM network.

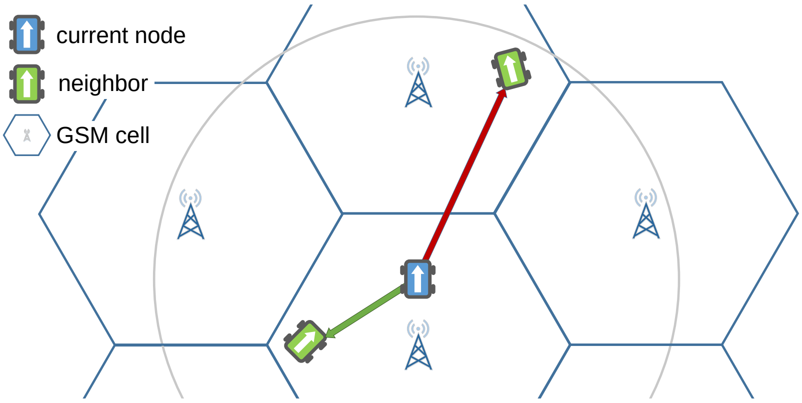


Figure 5.5 Cell verification between a node and its neighbors.

With a *base ring radius* of 2 km (as proposed in [56] and Section 5.6.2.1 of this chapter), it is proposed that nodes use the location spoofing detection for neighbors in the three innermost rings of their neighborhood structure. For the innermost ring (i.e. for neighbors up to 2 km away), they will try to perform a *cell verification*, thus trying to verify that they are located in the same GSM cell as neighboring nodes. A cell verification between a node and two of its neighbors is depicted in Figure 5.5. While one of the neighbors resides in the same cell, leading to a successful verification, the other does not and consequently fails the verification. For nodes in the second and third ring (up to 4 km and 8 km away, respectively), *location area verification* is used. Location area verification is performed in an analogous manner to cell verification, but uses data from the GSM PCCH that is visible within groups of multiple cells.

For more faraway nodes, no location verification is used. Locations shared with nodes in the outer rings are obfuscated into obfuscation regions with an edge length of 4 km[9] or more, making them less valuable for identification attacks or the inference of sensitive information. Without verification, no neighbor receives location updates with a higher precision than that. Likewise, if a node B shares a location that implies that it needs to be allocated to the innermost ring of a node A, but has only proven that it resides in the same location area as A, it only receives location updates with the precision corresponding to the second ring (ring index 1).

Algorithm 5.3: Periodic checks part of location verification.

```
foreach $N \in$ {neighbors in rings 0, 1 and 2} do
    $t$ = getCurrentTime();
    if $N$.cellVerificationPending == true then
        if $t$ - $N$.lastSuccessfulCellVerification > maxCellVerificationDelay then
            addToBlacklist($N$);
        else if $t$ - $N$.lastVerificationAttempt > cellVerificationAttemptTimeout then
            sendLocationVerificationRequest($N$);
        end
    else if $N$.LAVerificationPending == true then
        if $t$ - $N$.lastSuccessfulLAVerification > maxLAVerificationDelay then
            addToBlacklist($N$);
        else if $t$ - $N$.lastVerificationAttempt > LAVerificationAttemptTimeout then
            sendLocationVerificationRequest($N$);
        end
    else
        if (getRingIndex($N$) == 0) $\wedge$
        ($t$ - $N$.lastSuccessfulCellVerification > cellVerificationInterval) then
            cellVerificationPending = true;
            LAVerificationPending = true;
            sendLocationVerificationRequest($N$);
        else if $t$ - $N$.lastSuccessfulLAVerification > LAVerificationInterval then
            LAVerificationPending = true;
            sendLocationVerificationRequest($N$);
        end
    end
end
```

5.4.2.3 Verification process

Each node A periodically performs checks about the verification status of all of its neighbors in its innermost three rings. A detailed description of these checks is presented in Algorithm 5.3. If a cell or location area verification is pending for a neighbor B, a *location verification request* is sent to it. The request message contains the pseudonym of A as well as two encrypted location sketches according to the

[9] For a downscaling factor of $f_{\mathrm{d}} = 1$ (cf. Section 5.6.2.3) and nodes in the ring with index 3.

synchronous PET protocol outlined in [87]. One sketch is based on cellular-level broadcast data, the other on location area-level data. Upon receiving the verification request, neighbor B combines its own location sketches with the ones he received, according to the PET protocol. He sends the result of the operation back to A in a *location verification response*. Based on B's response, A can now check if B is in the same location area or even in the same cell as itself. Together with the location verification response, B also sends a new verification request, thus initiating the verification process in the other direction (A must now form a location verification response himself).

Once the proximity to a neighbor is verified, a more accurate location can be shared with him accordingly. The verification process is repeated periodically in order to protect against follower attacks. Without such a periodic reverification of neighbors, an adversary needs to be physically close to his victim only once, after which he can track the victim's movement by faking his location.

5.4.2.4 Dealing with identified attacker nodes

If, despite repeated attempts, a node A was unable to successfully verify its proximity to a node B claiming to reside within A's innermost 3 rings, the *maximum verification delay* will be reached. In this case, A assumes that B is a malicious node that has spoofed its location data. A then evicts B from its neighborhood structure and adds it to a blacklist of identified malicious nodes. This mechanism is also shown in Algorithm 5.3. While B is in A's blacklist, A doesn't send any messages to B and ignores all messages received from B. After a *retention period*, B is removed from the blacklist again. This approach prevents a malicious node from quickly regaining access to A's neighborhood, while at the same time reducing the impact of falsely accused nodes.

5.4.2.5 Practical considerations and alternative data sources

Based on the proof of concept provided in [73], it is assumed that the continuous collection of both IA and PCCH traffic and the efficient generation of location sketches from collected messages is possible for traffic participants. As the authors point out, however, changes to the GSM stack implementation might be necessary on client devices for the collection of the required broadcast traffic. An additional open question is whether the same networking interface used for data communication can be used for collecting GSM broadcast messages.

As an alternative to GSM-based private proximity testing, location tags can likely also be generated from signaling traffic generated in more recent cellular networking standards (e.g., 4G and beyond). However, to the best of the author's knowledge, no specific proposals for this existed at the time of writing. For cities and densely populated areas, packets broadcast on private wireless LANs are another promising option (as also identified in [87]).

5.5 Implementation and simulation model

A prototype of *OverDrive* was previously ([56]) implemented for the overlay simulation framework *OverSim* [3], with the goal of evaluating the system's performance characteristics. In order to evaluate the privacy enhancements proposed in this thesis, they were implemented as extensions to this prototype. In this section, several models are first introduced that are part of OverSim and are relevant to the evaluation presented in Section 5.6. Following that, implementation details about the location spoofing detection extension are presented.

In addition to the implementation of simulation models and privacy enhancements, an interactive demonstrator of the OverDrive protocol was developed that visualizes OverDrive's neighborhood structures and routing approach. Details concerning this demonstrator are presented in Appendix B.

5.5.1 Scenario-specific simulation models

OverSim was originally designed for non-mobile nodes using Ethernet or DSL connections. Thus, in [56], OverSim's underlay abstraction was extended by several new models: (1) a *network model*, that reflects the characteristics of data transmissions over 3G cellular networks, (2) a *mobility model* providing geographic information (such as position, speed, and direction) and modelling node movement and (3) a *churn model*, to handle the arrival and departure of nodes at the beginning and the end of car trips.

In the following, the focus is on the aspects of these models that are most relevant for realistically evaluating the privacy-related characteristics of OverDrive.

5.5.1.1 Network model

The purpose of the network model used for evaluation is to reflect the characteristics of common 3G cellular networks. The simulated area is divided into hexagons with an average cell radius of 2 km, representing mobile network cells. This division is used for calculating communication latencies based on the geographic position of nodes. On a side note, the same grid parametrization is also used for simulating GSM-based PPT (cf. Section 5.5.2).

5.5.1.2 Mobility model

A mobility model was implemented that associates nodes with geographic locations. This positioning information is available to overlay implementations and applications running on the simulated nodes, thus emulating the existence of positioning sensors like GPS receivers.

Additionally, a *pathfinding* movement model was implemented for modeling vehicular movement. Using this model, each node randomly chooses a source and destination location within the simulated area. Then, the fastest path is calculated between these two nodes, taking into account the different driving speeds on the individual roads comprising the path. Nodes move along this path until they reach their respective destination. To improve simulation performance, nodes can select their paths randomly from a pool of precomputed paths.

The road network underlying the simulation can be extracted from map data available through the *OpenStreetMap* project[10].

5.5.1.3 Churn model

The observed degree of node churn depends on the desired number of nodes in the population as well as the average path length. When the pathfinding mobility model is used, the end of a path also signals to the churn model that the node should be deleted. The node will leave the overlay immediately and a new one will join from another position.

5.5.2 Location spoofing detection

For assessing the impact of the location spoofing detection mechanism, an abstract model for the GSM-based PPT technique proposed in [73] was implemented. Specifically, *oracles* are used per node that, given a location tag, can determine if it was generated in the same GSM cell or location area as the node that the oracle belongs to.

A hexagonal grid with a cell radius of 2 km is used to model the cell structure of the used GSM network, and a hexagonal grid with a cell radius of 5 km to model the partitioning of the network into location areas. This model represents a conservative approximation to real GSM networks, which have a high variance in cell sizes and location area span (both usually larger than in the proposed model) [102, 108].

5.6 Evaluation

In this section, an evaluation of both the original OverDrive system and the novel enhancements proposed in Section 5.4 is presented. The main focus lies on the evaluation of privacy-related characteristics and on determining the effectiveness of the novel location privacy enhancements.

The general setup of the evaluation is outlined first and, following that, the discovery of a suitable parametrization of the enhanced OverDrive protocol is described. Based on the common setup and parameters, simulation models are constructed for the attack scenarios presented in Section 5.3: (1) the establishment of a global view with as accurate location data as possible about as many nodes in the network as possible and (2) the identification and tracking of a single victim using context knowledge. Both attacks are evaluated, comparing the original OverDrive design with a version enhanced with location obfuscation and location spoofing detection.

5.6.1 General evaluation setup

For simulating mobile nodes (OverDrive-enabled vehicles), the simulation models proposed in Section 5.5 are used. As an underlying road network, the highway network of the German state of Baden-Württemberg is used, which features around 5300 km of road in an area of around 56 000 km^2.

[10] http://www.openstreetmap.org/

For reducing simulation overhead, nodes do not calculate their paths through the road network on the fly, but instead choose them from a pool of 10^5 precomputed paths. Focusing on scenarios that have a higher percentage of longer drives, half of the paths that had a shorter driving time than the total simulation time were filtered out during path computation.

Unless noted otherwise, all results presented here were gathered using simulations with 10 000 honest mobile nodes. For some experiments, an additional number of adversary-controlled attacker nodes was introduced into the overlay.

Unless noted otherwise, no application apart from the overlay component was running on regular nodes. Consequently, no application-specific location leaking was evaluated and adversaries can only exploit the properties of the OverDrive protocol. On each attacker node, an attacker application was running that realizes the logic of the currently evaluated attack, the coordination of attacker nodes and the collective gathering of information.

For building up the overlay, nodes are initially inserted into the simulation at a low rate of 2 nodes per second. After 200 nodes have been added, nodes are inserted at a rate of 20 nodes per second until the desired node population size is reached. The bootstrapping of nodes, i.e., their initial integration into the overlay, is realized by providing each new node with the address of a random node that is already part of the overlay.

For each simulated parameter combination, four independent simulation runs were performed, each covering a period of 4200 s (simulated time). Measurements started after a 600 s warm-up period. Unless noted otherwise, the presented plots depict average values with error bars indicating 95% confidence intervals. The parameterization of the modified OverDrive component is kept similar to the optimal configuration of the original OverDrive component as determined in [56]. Changes were made concerning the parameterization of OverDrive's neighborhood structure for the modified version of the protocol (cf. Section 5.6.2).

The simulation parameters used for generating the results depicted in this chapter are summarized in Table C.1 (Appendix C).

5.6.2 Parametrization of OverDrive

The following section describes how suitable parameters for the OverDrive component and the novel privacy extensions were found. The parameters determined here are used for all simulation studies in this chapter.

5.6.2.1 Original OverDrive system

In order for new results to be comparable with the ones in [56], a parametrization for the original OverDrive protocol is used that resembles the optimal parameters proposed there. In [56], parameters were found using a comprehensive *performance versus cost* (PVC) evaluation, balancing delivery ratios with bandwidth requirements.

Amongst others, following parameters were determined as suitable for the Baden-Württemberg scenario[11]:

- A *desired number of neighbors* per ring of $n_{\mathrm{des}} = 4$. Nodes actively search for new neighbors for a ring if less than n_{des} neighbors reside in it.
- A *maximum number of neighbors* per ring of $n_{\mathrm{max}} = 20$. Nodes do not maintain more than n_{max} overlay neighbors per ring.
- A *base ring radius* (the radius of the innermost ring) of $r_{\mathrm{b}} = 2\,\mathrm{km}$. Since the rings used in OverDrive's neighborhood structure are concentric with exponentially increasing radii, this implies the second ring having a radius of 4 km, the third 8 km etc.

As a preliminary step to the evaluation of extensions proposed in this chapter, a similar simulation study was conducted to determine a parametrization of the enhanced OverDrive system that strikes a balance between privacy gain and performance impact.

5.6.2.2 Performance-centric parameters

In order to keep the simulation overhead at a reasonable level, suitable values were first determined for parameters with lesser expected impact on the system's privacy characteristics. This includes, for example, the number of nodes a node will accept as neighbors. As in [56], a PVC evaluation was used to determine parameter combinations with a good trade-off between routing success and bandwidth consumption. Using this approach, following combination of parameters was found to be suitable for the given scenario[12]:

- A desired number of neighbors $n_{\mathrm{des}} = 8$. (Each node actively searches for new neighbors for a ring if that ring has less than 8 neighbors.)
- A maximum number of neighbors $n_{\mathrm{max}} = 32$. (Nodes never maintain more than 32 neighbors per ring.)
- A base ring radius of $r_{\mathrm{b}} = 2\,\mathrm{km}$, which was also identified as an optimal value in [56].

This parametrization was used in all subsequent simulations.

[11] A few less relevant parameters are omitted here. A more complete list is presented in Table C.1 (Appendix C).

[12] The chosen parameters differ from those found for the original OverDrive system, as the proposed location privacy enhancements have an impact on geographic routing (due to obfuscated neighbor locations), maintenance overhead (location updates can be sent more rarely) and neighbor selection. A more complete list of parameters is presented in Table C.1 (Appendix C).

5.6.2.3 Privacy-relevant parameters

A suitable obfuscation level was determined for the obfuscation extension, i.e. a value for the downscaling factor f_{d}. Towards this goal, the impact of different values for f_{d} on the adversary's success at establishing a global view on the network was considered (the evaluation scenario is described in Section 5.6.3). Simulation results confirmed that the average error in the location data known to the adversary grows with the degree of applied obfuscation. However, using a high degree of obfuscation also tampers with the system's performance in delivering geocast messages. Based on the results, a downscaling factor of $f_{\mathrm{d}} = 1$ was selected, leading to an improvement to the regular OverDrive design in both performance (cf. Section 5.6.5) and adversary uncertainty.

Suitable parameters for the location spoofing detection extension were needed as well. Here, the main optimization goal was to decrease the additional communication overhead while increasing the chance that two proximate nodes will correctly verify each other as such. Location verification is prone to false negatives when nodes reside in different cells or location areas despite their proximity. Simulations were performed using the same base scenario as in the remaining parts of this chapter, varying the relevant parameters. Based on the gathered simulation results, a parametrization was selected in which nodes require a successful cell-based verification every 600 s and consider a node malicious if the verification has failed for 150 s. For LA-based verification, 900 s and 300 s are, respectively, used. Nodes are removed from blacklists after a retention period of 300 s.

5.6.3 Establishment of a global view

In the following, the evaluation of the difficulty for an adversary to construct a global view of the OverDrive network, including the positions of all (pseudonymized) nodes, is presented.

5.6.3.1 Evaluation scenario and metrics

Simulations were conducted with 10 000 mobile nodes and an additional population of attacker nodes that exhibit the same mobility pattern as regular nodes. Different sizes of the attacker node population were evaluated up to a maximum of 100 nodes. All attacker nodes were considered to be under the control of one adversary that combines their views on the overlay network into one global view. For evaluating the location spoofing detection, attacker nodes were additionally assumed to be lying about their location, i.e., never being physically present at the locations they claim to be. This assumption models an adversary without the resources to use actual vehicles for gathering surveillance data.

In order to establish a global view on the overlay network, attacker nodes attempt to become overlay neighbors to as many regular nodes as possible, thus learning their geographic positions. The positions are then sent to a centralized *attacker observer* who combines the input of all attacker nodes into one global view of the network. The completeness of this view, in terms of the metrics introduced shortly, is verified every minute.

A central evaluation metric for evaluating the attack success in this scenario is the percentage of nodes known to the adversary, referred to as the *surveillance coverage*. Additionally, the *distance disparity* metric is introduced, which describes the distance between the node position known to the adversary pos_{att} and the actual position of the node pos_{real} at any given time. Given the distance (in km) between two geographic locations P and Q as $d(P, Q)$, the distance disparity disp for node X can be calculated as:

$$\text{disp}(X) = d(\text{pos}_{\text{att}}(X), \text{pos}_{\text{real}}(X))$$

5.6.3.2 Results

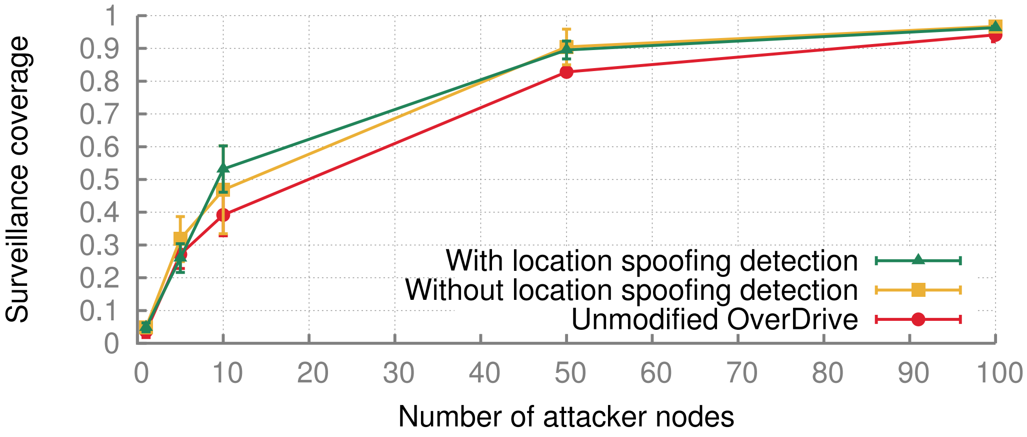


Figure 5.6 Surveillance coverage in relation to the number of attacker nodes.

Figure 5.6 depicts the average surveillance coverage measured using the unmodified OverDrive from [56], OverDrive with enabled obfuscation and OverDrive with both enabled obfuscation and enabled location spoofing detection. Since the surveillance coverage metric expresses the percentage of nodes for which the adversary has location data but gives no information about that location data's precision, the impact of applying obfuscation and location spoofing detection is negligible. The use of obfuscation even leads to an increase in surveillance coverage. This increase can be explained through the fact that the parametrization used for the obfuscation-enabled OverDrive (with $n_{\max} = 32$) causes a larger number of nodes to be accepted as neighbors. Concerning the effect of location spoofing detection, it should be noted that the blacklisting of nodes that fake their location is ineffective at reducing surveillance coverage. First, blacklisting is only local, i.e., only nodes that have failed to verify the proximity of an attacker node cease communication with it. Second, nodes are removed from blacklists after a retention period to reduce the impact of verification false negatives.

More importantly, a significant improvement can be noted concerning the precision of the locations known by the adversary. Figure 5.7 depicts a cumulative distribution of measured distance disparity values, in a scenario with 100 attacker nodes and averaged between simulation runs with identical parameters. The plot depicts the distance disparity plotted against the fraction of all known nodes with a smaller or

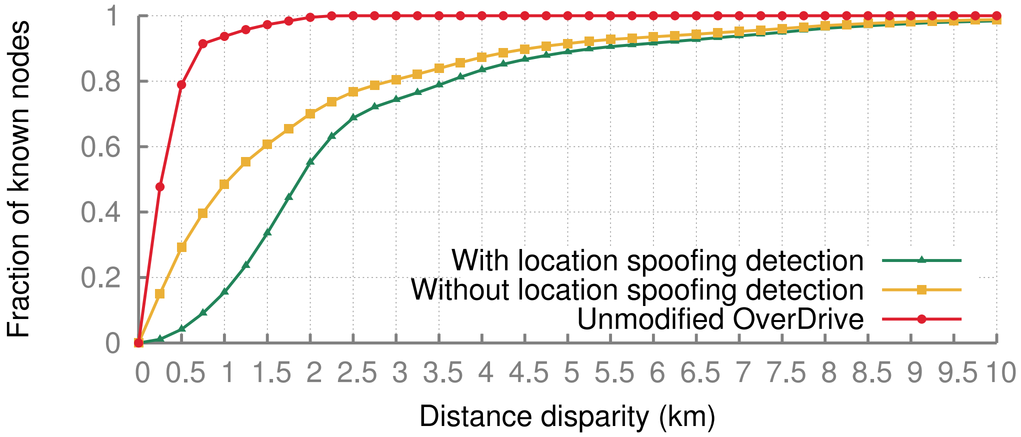


Figure 5.7 Cumulative distribution of the distance disparity of all nodes known to an adversary with 100 attacker nodes.

equal distance disparity. It can be seen that with the unmodified OverDrive system, the adversary knows the positions of 80% of the nodes known to him with a precision of less than 500 m. When using the obfuscation-based privacy enhancement, the adversary reaches this accuracy with only about 30% of the nodes known to him. With location spoofing detection, more than 54% of the node positions known by the adversary are wrong by more than 1.5 km. Here, location spoofing detection prevents nodes from sharing accurate location data with attacker nodes, as the latter always fake their location. Note that these numbers also include nodes which happen to be near the center of their obfuscation region, thus yielding a low distance disparity even at a large level of obfuscation. Especially in populated areas, the measured levels of uncertainty make the collected location data unusable for breaking pseudonyms or determining the destinations of pseudonymized nodes. Thus, establishing a global view becomes significantly less useful for obtaining identifiable location samples.

5.6.4 Identification of an individual target

This section presents the evaluation of the difficulty for an adversary to identify the pseudonymized node belonging to a specific victim. The adversary is assumed to have context knowledge about his victim in the form of the location at which it will start its trip (cf. Section 5.3.4.2).

Instead of building a global view, the adversary's approach here is to identify a pseudonymized victim using context knowledge and only a minimal number of attacker nodes.

5.6.4.1 Evaluation scenario and metrics

An evaluation scenario was constructed based on the attack described in Section 5.3.4.2. A network with 10 000 regular OverDrive nodes was simulated. These nodes start their trips from random locations as described in Section 5.5.1. An additional population of 100 *victim nodes* was introduced, that behave like regular OverDrive nodes

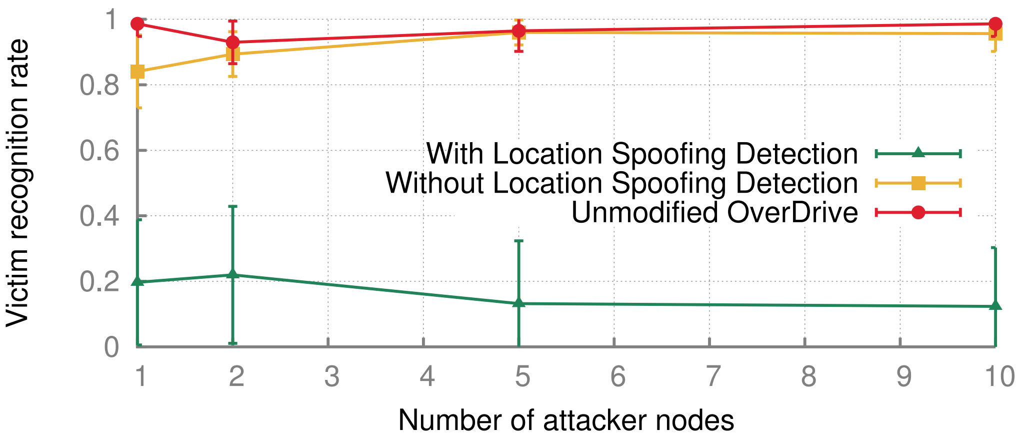


Figure 5.8 Attack success for identifying individual target.

but enter the network at wider intervals (randomly distributed around 2 minutes) and at a common fixed *victim start point*. The start point is chosen randomly at the beginning of each simulation and is known by the adversary.

The victim start point is the only information the adversary has in order to distinguish between victim nodes and regular nodes.

The adversary introduces a set of up to 10 stationary attacker nodes to the network, that fake their location to random positions within a radius of 1 km around the victim start point. The attacker nodes continuously report new nodes they discover via overlay maintenance traffic like neighbor discovery messages. Based on the reasoning in Section 5.3.4.2, if the adversary learns about a node for the first time while that node is within 1 km of the victim start point, that node is marked as a potential victim. Based on this recognition approach, the *victim recognition rate* can be measured - the ratio of victim nodes that were correctly identified by the adversary. As the attacker success greatly depends on the choice of a victim start point, e.g., because it influences the likelihood of regular nodes appearing in the vicinity of that point as well, four times as many simulation runs were performed for this experiment, i.e., a total of 16 per configuration.

5.6.4.2 Results

Figure 5.8 depicts the results for this experiment. The adversary achieves a victim recognition rate of above 90% for the unmodified version of OverDrive, due to the unrestricted sharing of accurate location data by victim nodes. The recognition rate is not 100%, because victim nodes move away from their start position and are not always immediately discovered by attacker nodes. When using the obfuscation-enabled OverDrive system without location spoofing detection, the recognition rate remains similarly high. This effect is due to the fact that the attacker nodes pretend to be very close to the victim, which causes the victim node to share more accurate location data. When using the location spoofing detection mechanism, the adversary scores a much lower recognition rate of only about 20%. Here, nodes will not share accurate location data with neighbors with whom the physical proximity has not been verified. Since attacker nodes are not physically in the area of the victim start point, the adversary will receive location data with an average error of around 1 km (corresponding to an obfuscation region with an edge length of 4 km), which greatly diminishes the success rate of state-of-the art identification attacks.

Location spoofing detection also hinders a subsequent tracking of victims. Malicious neighbors in the innermost rings are blacklisted after multiple verification attempts have failed. Nodes in the outer rings, on the other hand, have a lower chance of remaining in the neighborhood due to the overlay maintenance logic. Thus, even if an adversary achieves a higher recognition rate by using additional side channels (e.g., physical observation), the subsequent tracking via a follower attack is no longer practical.

5.6.5 Impact on performance

Here, results are presented concerning the impact of the proposed extensions on system performance.

5.6.5.1 Evaluation scenario and metrics

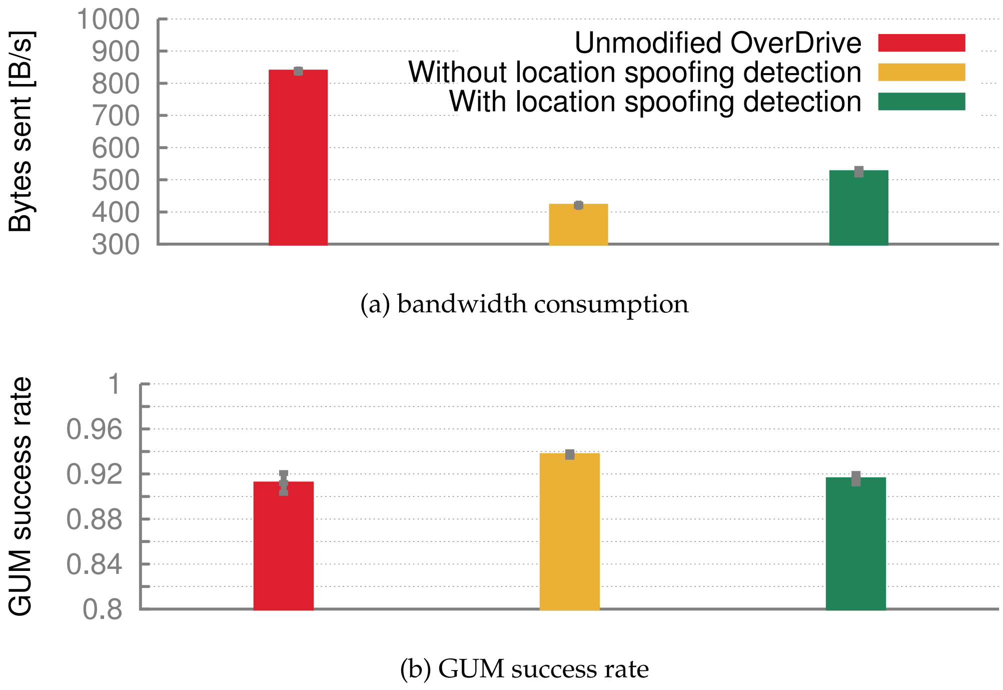
In line with [56], the main focus here is on two metrics: the consumed bandwidth of the system measured in sent bytes per node, and the success rate for GUMs. For measuring both in a realistic environment, a test application was running on each node, that sends GUMs to randomly placed circular areas and keeps track of successfully delivered messages. In line with [56], the GUM success rate $\mathrm{SR}_{\mathrm{GUM}}$ is defined as the ratio between the number of messages that were successfully delivered m_{succ} and the number of messages that could have been successfully delivered. The number of messages sent to areas without any nodes (resulting in unavoidable errors), denoted by m_{unavoid}, is not counted towards the GUM success rate. Thus, with m_{total} denoting the total number of messages sent,

$$\mathrm{SR}_{\mathrm{GUM}} = \frac{m_{\mathrm{succ}}}{m_{\mathrm{total}} - m_{\mathrm{unavoid}}}$$

The presented results were gathered using simulations with 10 000 honest mobile nodes. Again, four independent simulation runs were performed for each evaluated configuration.

5.6.5.2 Results

Despite the fact that the modifications presented here are aimed at improving the privacy characteristics of OverDrive, they also have a positive impact on performance. Figure 5.9 depicts results measured for the OverDrive system presented in [56] in comparison with values measured for the enhanced versions of the system that were presented here. While the average success rate increases slightly, the bandwidth consumption of the system drops significantly when obfuscation is enabled. With obfuscation, location updates to neighbors need to be sent significantly less often, namely only on changes in the reported obfuscation region. Thus, nodes



(a) bandwidth consumption

(b) GUM success rate

Figure 5.9 Performance impact

are able to maintain more neighbors without causing an increase in bandwidth consumption, while causing an increase in success rates due to a higher interconnection in the overlay. The activation of location spoofing detection has a more moderate impact on performance. The GUM success rate drops slightly, as proximate nodes start sharing precise location data only after a successful location verification. The bandwidth consumption increases by about 100 B/s. Compared to the sending of location updates, the private proximity test protocol needs to be performed less frequently, which explains its lower impact on bandwidth consumption.

5.7 Conclusion

This chapter introduces key mechanisms needed for realizing privacy-preserving long-distance geocast services that do not rely on individual service providers or dedicated infrastructure support. Through the proposed location obfuscation concept, the precision of location data shared with peers in a decentralized system can be decreased with increasing distances to those entities, enforcing data locality. Through the integration of location spoofing detection using GSM broadcast traffic, the information gain for an adversary from faking his position is reduced significantly. Both proposed techniques were designed as extensions to the overlay-based geocast service OverDrive. Through simulation studies, their effect on location privacy was evaluated as well as their impact on performance. The results demonstrate that even strong adversaries controlling hundreds of nodes are effectively prevented from identifying and tracking users with an acceptable level of certainty.

6. Sybil-resistant pseudonymization and pseudonym change

As an important challenge in the context of cooperative services, the preservation of user privacy must be balanced with the protection against abuse from malicious participants. The protection against sybil attacks, where adversaries create large numbers of fake virtual identities (*sybils*), is especially relevant in this context. Sybil attacks enhance the potential magnitude of abuse and enable malicious users to avoid punishment (e.g., blacklisting) by switching to fresh identities. In this way, sybil attacks can greatly reduce the utility of smart traffic services.

The issuing of *pseudonyms* to users is a common solution to the challenge of hiding user identities while enabling access control and preventing sybil attacks. To prevent pseudonym deanonymization through continuous observation and correlation, frequent and unlinkable *pseudonym changes* must be enabled.

Existing approaches for realizing sybil-resistant pseudonymization and pseudonym change are either inherently dependent on trusted third parties (TTPs) or assume the existence of significant resources at end-user devices. TTPs take the form of certification authorities and pseudonym issuers and are used for enforcing issuing criteria and ensuring the correct (sybil-free) distribution of pseudonyms. Upon compromise of such a TTP, large-scale sybil attacks become possible and the trustworthiness of issued pseudonyms is reduced. Centralized pseudonym issuers thus become attractive targets for attacks, resulting in high operational costs for maintaining their security. Additionally, the notion of universal trust anchors shared by all system participants is questionable when considering mobile users in a globally interconnected world. In this chapter, the use of distributed append-only bulletin boards, as provided by block chain networks, is explored as an alternative to centralized TTPs for realizing sybil-resistant pseudonymization and pseudonym change.

Related work is introduced first in Section 6.1. Following that, in Section 6.2, a novel approach towards TTP-free and abuse-resistant pseudonymization and pseudonym change is proposed. To the best of the author's knowledge, this is the first proposed pseudonymization and pseudonym change system that both prevents sybil attacks and doesn't rely on a TTP for ensuring the correctness and security of any of its operations. *BitNym*, a specific instantiation of the approach that leverages the security of the unmodified *Bitcoin* [84] network, is presented in Section 6.3. TTP-independent mechanisms for realizing sybil-free *initial access control* (IAC), *pseudonym validation* and *pseudonym mixing* are discussed and proposed. The sybil-resistance of BitNym is then investigated in Section 6.4. Section 6.5 applies the analysis approach introduced in Chapter 3 to identify potential privacy threats applying to BitNym and the general approach. Suitable metrics are furthermore defined. Section 6.6 introduces a proof-of-concept prototype of BitNym and a simulation-based study investigating pseudonym mixing and anonymity set development. Section 6.7 concludes this chapter with a summary of contributions and results.

This chapter is largely based on [44].

6.1 Related Work

A discussion of existing approaches to preventing sybil attacks can be found in Section 2.5.1. Here, a concise overview of such works is given, focusing on approaches that realize pseudonymity and pseudonym change.

Approaches based on blind signatures [60] and zero-knowledge proofs [14] have been proposed for enabling the centralized issuing of pseudonyms that are not identifiable by the pseudonym issuer (i.e., the TTP does not have to be trusted for preserving the pseudonymity of users). Here, a TTP is also required for changing pseudonyms, as the old pseudonym must be marked invalid to prevent sybil attacks. Alternative approaches like e-token dispensers [13] or the issuing of pools of pseudonyms with non-overlapping validity periods still inherently require a TTP for enforcing issuing criteria and ensuring that the issuing of pseudonyms is sybil-free.

As an alternative to TTP-based access control, decentralized approaches for preventing sybil attacks were proposed. Proof-of-resource schemes like *proof-of-work* [10] and *CAPTCHA* [113] are ineffective for systems involving end-users, as determined adversaries can amass orders of magnitude more resources than regular users are ready to spend for continuously using a system [69]. Approaches based on the *social graph* between users [112] are more promising. However, existing designs either mandate the identifiability of users or are suited only for scenarios in which nearly all users are continuously reachable and active within a large-scale peer-to-peer system.

The usage of *proof-of-burn*, i.e., the provable destruction of a non-replenishable resource, was proposed for establishing sybil-free and trustworthy online identities [54] using the Bitcoin cryptocurrency [84]. However, resulting identities are linkable to the originating users. Even if funded via anonymous sources, changing such identities is not possible without repeating the initial investment (which should be significant in order to effectively limit attackers).

In [47], Garman et al. propose a scheme for realizing decentralized anonymous credentials. As in the proposal developed here, the authors build upon cryptocurrency systems like Bitcoin for avoiding the dependence on TTPs. However, the proposal is based on computation-intensive zero-knowledge proofs, making it less suited for more resource-constrained devices, and requires a TTP during the initial setup phase. The challenge of initially issuing pseudonyms in a TTP-independent and yet sybil-resistant manner is discussed only marginally. Also, the blacklisting of malicious users is not supported. To the best of the author's knowledge, this chapter introduces the first pseudonymization system that is fully decentralized, resistant to sybil attacks and supports efficient and unlinkable pseudonym changes as well as the blacklisting of malicious pseudonym holders.

6.2 General approach

In the following, the goals of the presented approach are introduced as well as technologies and assumptions that it is based on. A rough overview of the abstract system is then given.

6.2.1 Goals and offered functionality

An approach towards TTP-free and abuse-resistant pseudonymization and pseudonym change is proposed. At its core, it aims at providing users with non-identifiable *pseudonyms* that can be used in cooperative services of any kind. In addition to smart traffic applications, i.e., in systems like *OverDrive* (cf. Chapter 5), the provided pseudonyms can be used for authenticating to online services, in peer-to-peer systems and for protecting collaborative sensing in the Internet of Things. In accordance to Pfitzmann et al. [94], and as in the remainder of this thesis, "pseudonym" is understood here as an identifier of a subject other than one of the subject's real names. While hiding the identity of users, pseudonyms generated by the presented approach should offer security against abuse, by effectively preventing sybil attacks and supporting the punishment of malicious pseudonym holders. More specifically, the goal is the realization of following properties:

1. Unlinkability of pseudonyms to user identities (i.e., non-identifiability) and to other pseudonyms by the same user.

2. Limitation to a bounded number of simultaneously active pseudonyms per user.

3. Authenticity of the linking between a pseudonym and its holder.

4. Possibility of unlinkable pseudonym changes.

5. Possibility of blacklisting pseudonym holders.
6. Complete independence of trusted third parties (TTPs) for realizing any of the preceding properties.

Properties 1 to 5 have been widely discussed in the literature [94] and are shared by multiple existing and proposed systems. Their combination with property 6, on the other hand, is, to the best of the author's knowledge, unique to the presented solution.

6.2.2 Decentralized append-only bulletin boards

The followed strategy for avoiding the reliance on TTPs is to build upon recent results on realizing distributed consistency in highly adversarial environments. More specifically, the presented solutions are built upon decentralized cryptocurrency systems like *Bitcoin* [84], in which a globally consistent *transaction* log is collaboratively maintained within a network of non-colluding peers. Transactions are typically grouped in *blocks*, yielding a cryptographically secured *block chain* as the practical manifestation of the network consensus. In the scope of this thesis, Bitcoin-like networks are therefore referred to as *block chain networks*. A high-level overview of the block chain paradigm and its underlying trust assumptions was given in Section 2.4.5. In the following, a few more technical details are elaborated.

Transactions in block chain networks are typically composed of:

- *outputs* - the number of funds exiting the transaction in combination with a challenge that needs to be solved for *spending* them - and
- *inputs* - references to preceding outputs in combination with a valid solution.

Figure 6.1 depicts an example involving two linked Bitcoin transactions. The first (*tx 1*) references two outputs of transactions that are not depicted, giving the whole transaction a value of 119.9 millibitcoin (mBTC)[1]. In Bitcoin, *transaction fees* are typically payed for including a transaction on the block chain. These fees are harvested by so called *miners*, peers that invest computational resources for maintaining the security of the system. The fee included in a transaction is determined by subtracting the sum of all inputs from the sum of all outputs. While the fee is, thus, chosen by the transaction's author, transactions with fees that are too low might not be included in upcoming blocks or only with a significant delay.

In the depicted example, the whole value of *tx 1* is concentrated in one output. The challenge in this output is such that it can be solved only by the holder of a specific cryptographic key (in this case Alice). Alice presents a solution in the first (and only) input to the second transaction (*tx 2*). In this transaction, she allocates 50 millibitcoin to an output encoded in such a way that only Bob can present a correct solution. In effect, the 50 millibitcoin are *transferred* to Bob. The remaining value, or change money, is encoded in the second output of *tx 2*. This output is encoded so that only Alice can present a solution. In effect, the change is transferred back to Alice.

[1] 0.1199 bitcoin (BTC).

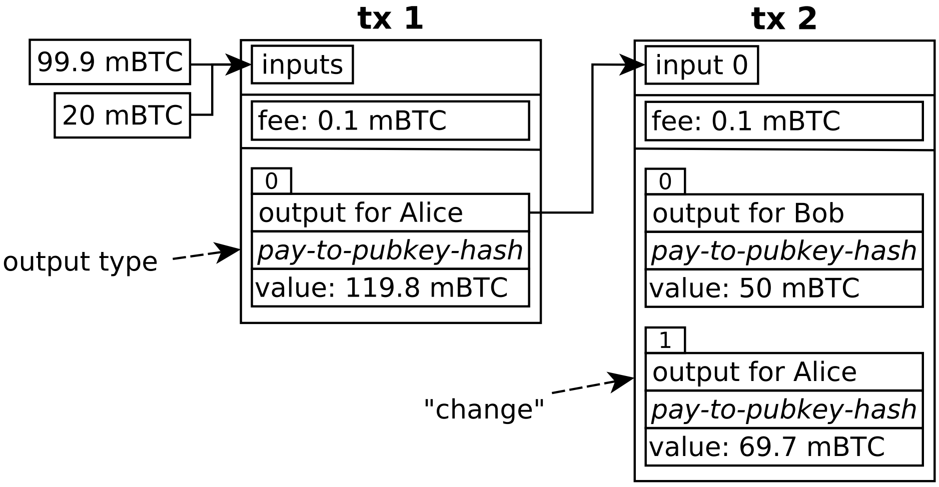


Figure 6.1 Example transaction chain. Typical payment scenario.

Typically, outputs contain the public part of an asymmetric cryptographic key pair (respectively a hash thereof) and can be spent upon proving the possession of the corresponding secret key[2]. In addition, the inclusion of arbitrary data in transactions is possible. At the very least, without dedicated support from the underlying block chain protocol, transaction outputs can be used for this purpose (possibly rendering the funds allocated to the output unspendable). In this spirit, and despite the original design goal of facilitating online payments, the block chain paradigm has been previously adapted for a wide variety of different applications. Using *colored coins* [99], for example, the ownership of arbitrary assets can be encoded and transferred by marking (*coloring*) cryptocurrency units. Other examples include name services[3], anonymous credentials [47] and the timestamping of cryptographic commitments [24]. In the context of this trend, it can be observed that block chain networks effectively realize general-purpose, decentralized append-only bulletin boards.

Despite their independence of TTPs, block chain networks are highly resilient to malicious tampering. This property is mainly due to the extensive use of cryptography for securing ownership and the transfer of funds as well as dedicated mechanisms for ensuring the append-only feature of the transaction log in the face of sybil attacks. Sybil attacks are prevented by tying the levels of influence individual peers are able to exert on the block chain to the possession of limited resources. In Bitcoin, for example, computing power in the form of extensive proofs of work is required for adding new blocks to the block chain (also referred to as *mining*)[4]. An adversary must control more computing resources than the remainder of the network in order

[2] In Bitcoin, this is realized via the *pay-to-pubkey-hash* output type depicted in Figure 6.1.

[3] `https://namecoin.info/`

[4] Note that this is different from creating a new transaction. Transactions are simply broadcast to the network. Valid transactions are then later included in blocks.

to cause significant disturbances. In the case of Bitcoin, this requires a significant investment.

The general approach outlined here can be applied to any type of block chain network. Still, for the remainder of this chapter, the focus is on Bitcoin for developing a specific design that is immediately deployable. By layering on top of the Bitcoin network, the high level of security stemming from its large population of non-colluding mining peers can be leveraged.

6.2.3 Assumptions and adversary model

It is assumed that the cryptographic building blocks used in the underlying block chain network are secure. For Bitcoin, this includes, most prominently, SHA-256 and ECDSA with the secp256k1 curve. It is also assumed that users are able to generate secure asymmetric key pairs and maintain the confidentiality of their secret keys. It is also assumed that users can form communication links to services and between each other without leaking identifying information like IP addresses. This assumption can be realized in practice by building upon anonymous communication services like Tor and PTP (cf. Section 2.6.2).

Concerning the assumed adversary model, the focus is on adversaries that attempt to either disrupt the pseudonymization system, launch sybil attacks or link pseudonyms to user identities. Adversaries are considered that can observe and modify all communications but are unable to identify pseudonym holders based on communication metadata, i.e., are effectively stopped by the used anonymous communication services. Adversaries might collude with (or compromise) individual users, in which case that users' pseudonyms are known to them, but with no more than 50% of the user population. Additionally, it is assumed that adversaries cannot attack the underlying block chain network, disturbing its functionality as an append-only bulletin board. In the case of Bitcoin, for example, it is assumed that no adversary can control more than 50% of the computing resources ("hash power") in the network.

6.2.4 Overview

The presented pseudonymization and pseudonym change approach is based on three core building blocks: *genesis pseudonym creation*, *pseudonym change* and *pseudonym validation and use*. These building blocks and the interplay between them is depicted in Figure 6.2. The building blocks directly address the desired properties discussed in Section 6.2.1. A central challenge that is tackled is to ensure the resistance to sybil attacks (property 2) in all three building blocks while remaining independent of TTPs (property 6).

At the core of the presented approach, *pseudonyms* are encoded in the outputs of transactions. Ideally, *cryptocurrency addresses* from the underlying block chain network can be reused. For example, if Bitcoin is used, pseudonyms can be chosen to correspond to Bitcoin addresses and be represented by the hash of an ECDSA public key. Users can authenticate their holding of a pseudonym in the same way they would prove their ownership of the corresponding address (satisfying property 3).

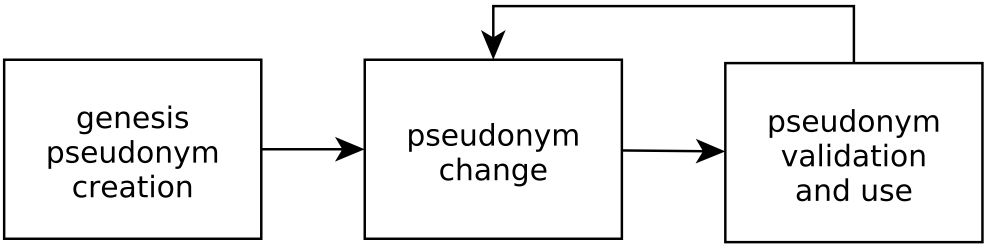


Figure 6.2 Overview of presented pseudonymization and pseudonym change approach.

The *validity* of a pseudonym, however, depends on additional factors. A pseudonym is only valid if the output it is encoded in exists on the block chain and hasn't been spent. Additionally, a chain of transactions leading from the output to a valid *genesis pseudonym transaction* (GPTx) must be provided. The latter is a special transaction encoding a proof that a predefined set of issuing criteria has been met by a user. A *genesis pseudonym* is included in one of the outputs of a GPTx and might, depending on the used access control approach, not be completely unlinkable to a user identity. Thus, for ensuring the unlinkability of pseudonyms and allowing unlinkable pseudonym changes (satisfying properties 1 and 4), state of the art techniques for anonymizing cryptocurrency transactions are adapted for realizing *pseudonym mixing*. After a successful mix involving several pseudonym holders, a user will likely begin using a different GPTx for proving his current pseudonym's validity. For ensuring that pseudonym changes do not enable sybil attacks, unambiguous transaction chaining rules are defined so that each GPTx can be used for validating only one currently active pseudonym.

6.3 BitNym

This section introduces *BitNym*, a specific design of TTP-free and sybil-resistant pseudonymization and pseudonym change based on the Bitcoin network. The discussed techniques are, in principle, directly applicable to other block chain networks. In accordance to 6.2.4, BitNym is divided into the building blocks *genesis pseudonym creation*, *pseudonym validation and use* and *pseudonym change*. The possibility for *blacklisting* pseudonym holders based on malicious behavior is discussed as well.

6.3.1 Genesis pseudonym creation

The number of genesis pseudonyms a user has created forms the upper bound for the number of pseudonyms he is able to simultaneously hold. Thus, *initial access control* (IAC) needs to be performed in the course of genesis pseudonym creation, to ensure that only legitimate users receive pseudonyms and that sybil attacks can be prevented. In the following, technical details regarding the realization of GPTxes in BitNym are introduced, followed by a discussion of approaches for handling IAC.

6.3.1.1 Genesis pseudonym transaction

As noted in Section 6.2.2, transactions in Bitcoin are composed of one or more *inputs* and one or more *outputs*. For the GPTx, inputs come from regular Bitcoin transactions (i.e., transaction without any special semantic in the context of BitNym). These are required for *funding* - without the provision of transaction fees, transactions cannot be written to the block chain. As the genesis pseudonym can be changed immediately after creation, funds are not required to come from an anonymous source.

The GPTx is required to contain two special outputs: the *marker output* and the *pseudonym output*. The outputs are defined by their ordering within the transaction: the first output is the marker output and the second the pseudonym output. GPTxes may also have additional outputs, but these are ignored in the context of BitNym.

The marker output is provably unspendable (any funds allocated to it are lost) and has the capacity for holding 40 bytes of arbitrary data. These qualities are realized by using an *OP_RETURN* code in the beginning of the output script[5]. Similar marker outputs are also used in the colored coin approach. The marker outputs in BitNym contain a magic number in the first two bytes (that is the same for all GPTxes), for easing the detection of GPTxes. The remaining 38 bytes are used for storing a proof that the IAC criteria have been met.

The pseudonym output is based on the *pay-to-pubkey-hash* pattern, i.e., it is a regular output used for funding a Bitcoin address. The destination Bitcoin address is generated by the user (who then also holds the corresponding private key) and forms the genesis pseudonym. The amount of bitcoin allocated to the pseudonym output determines the *value* of the genesis pseudonym. The value of a pseudonym roughly determines the number of pseudonym changes its holder can perform without recharging it with additional funds. An example GPTx is depicted in Figure 6.3.

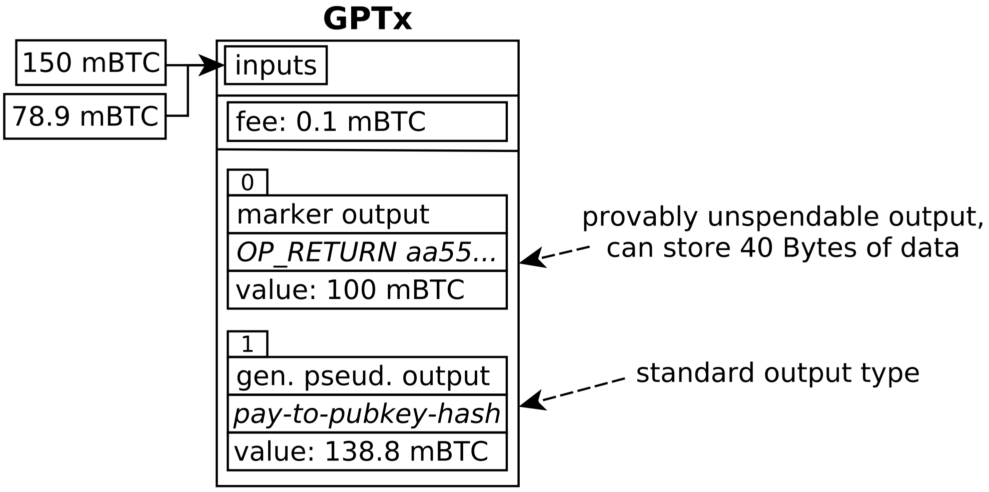


Figure 6.3 Example genesis pseudonym transaction (GPTx).

[5] Bitcoin uses a scripting system for encoding the requirements for spending an output.

6.3.1.2 Initial access control

The initial access control (IAC) building block defines the criteria based on which the validity of a GPTx is determined. In a sense, it provides a certification service for genesis pseudonyms. Thus, as a naive solution, it can be realized using a TTP that verifies if a user meets some predefined issuing criteria (e.g., that he can present an identity document and that he hasn't received a genesis pseudonym before). Upon verification, the TTP can provide a cryptographically signed validity acknowledgement that the user can include in the marker output of his GPTx.

One of the central advantages of the pseudonymization approach presented in this chapter is that a dependence on a TTP is not inherently given. In the following, several approaches for realizing IAC without a TTP are discussed that are compatible with BitNym. The approaches focus primarily on ensuring that sybil attacks are prevented, i.e., that every human user is able to create only a bounded number of valid genesis pseudonyms. Following approaches are possible:

- *Proof-of-work.* Users can be required to solve a computational puzzle (as in [10] or [4]) for producing a valid GPTx. While easy to realize, the security of this approach is questionable. Specifically, regular users often feature both restricted computational resources and a low time budget while adversaries can rent or buy specialized hardware and leave it running for longer periods of time. Thus, the puzzle difficulty will likely be either too high, deterring honest users and slowing adoption, or too easy, making large-scale sybil attacks possible for adversaries with moderate computational resources.

- *Proof-of-burn.* A related approach to proof-of-work is the use of proof-of-burn, i.e., the provable destruction of valuable resources. In the context of Bitcoin, proof-of-burn is implemented by provably rendering a certain amount of funds unusable for future transactions [54]. In BitNym, proof-of-burn can be realized by allocating a certain amount of Bitcoin to the marker output of the GPTx (cf. Figure 6.3). In contrast to proof-of-work, proof-of-burn enables the instant creation of genesis pseudonyms and adversaries are unable to leverage economy of scale effects: every genesis pseudonym "costs" the same for everybody. Large-scale sybil attacks are thus rendered unattractive even with small proof-of-burn requirements. Additionally, spending "money" for registering an online identity may cause users to reconsider malicious actions if blacklisting is possible [54]. Thus, while not perfect (participation rights and influence become tied to economic wellbeing), proof-of-burn is a viable option for IAC and is also used in the proof-of-concept prototype presented in this chapter.

- *Social graph-based IAC.* A third avenue for exploration is to leverage social connections for deciding about the trustworthiness of new users and preventing sybil attacks. For example, genesis pseudonyms can be considered valid only if backed by sufficient "is not a sybil" acknowledgements from established users. Additionally, such acknowledgements can be issued only to users fulfilling some set of predefined access criteria that are verified by established users (e.g., access might be restricted to members of a given organization). Acknowledgements can be issued pseudonymously, therefore not leaking any social graph information. For ensuring that a small group of users cannot introduce an arbitrary number of sybils, a high threshold must be chosen for the number of required acknowledgements. This requirement can quickly become unpractical; honest users might not have sufficient social contacts. Thus, a *preselection mechanism* is necessary. The use of a delay-tolerant *darknet* is envisioned for facilitating this. In a darknet, only trusted users (e.g., users with whom a social connection exists) are accepted as peers and only direct neighbors are aware of each other's identities. In this way, the social relationships between users can remain hidden.

The darknet can be used by new users to convince a large number of existing users of their own non-sybilness, collecting acknowledgements from them. As a more specific proposal, individual darknet links can be used only a bounded number of times for forwarding queries, therefore limiting the impact of sybil clusters and ensuring that the preselection mechanism is in itself sybil-free. Existing results on realizing sybil-resistant DHTs by leveraging the social graph between users (e.g., [81]) hint at the potential feasibility of this idea.

6.3.2 Pseudonym validation and use

In order for a user to be able to use a pseudonym, he must prove that:

1. He is the holder of the pseudonym.
2. The pseudonym is valid.

For 1, it is sufficient to use the pseudonym as an identity certificate, i.e., using the corresponding private key for signing messages. For 2, the user needs to construct a *proof* based on a valid GPTx and a transaction in which the Bitcoin address corresponding to the pseudonym has received funds. For ensuring the unlinkability of pseudonyms to user identities, validity proofs shouldn't disclose any additional information about a pseudonym holder other than the fact that his pseudonym is valid. A specific approach for building such proofs is introduced in the following.

6.3.2.1 Validation path

In Bitcoin, an input is always linked to one specific output of a preceding transaction. In this way, transactions are linked. By defining a mapping between the inputs and outputs within the same transaction, a path in the transaction graph can be defined. In BitNym, such a mapping if defined via the ordering of inputs and outputs, i.e., via the input and output indices. The i'th output is mapped to the i'th input of the same transaction. The resulting path is referred to as the *validation path*. An example validation path is depicted in Figure 6.4. The pseudonym encoded in the marked pseudonym output is validated using a path containing two *mix* transactions (*mix tx 1* and 2) and a GPTx created by Alice. However, the pseudonym is not necessarily held by Alice[6]. As long the marked pseudonym output hasn't been spent in a subsequent transaction, the pseudonym encoded in it remains valid.

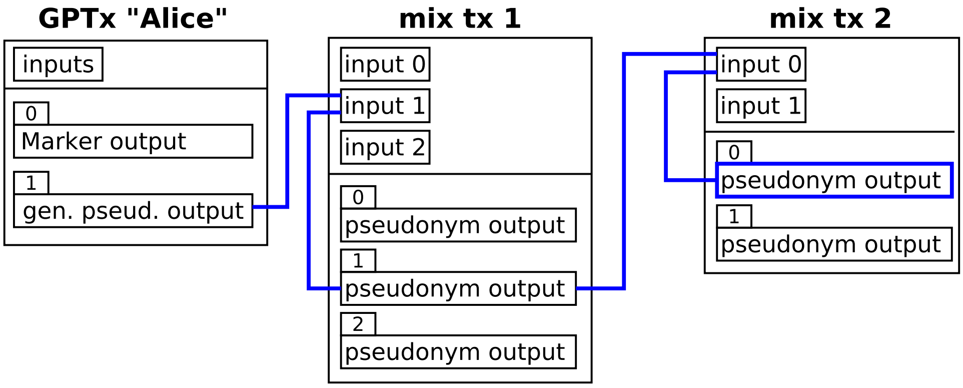


Figure 6.4 GPTx created by Alice and validation path involving that transaction. The validated pseudonym is not necessarily held by Alice.

6.3.2.2 Constructing a proof

Pseudonyms holders must be able to present a validation path from their current pseudonym to a valid genesis pseudonym. More specifically, a *proof message* must be constructed that includes:

- A transaction with an output containing the pseudonym.
- A GPTx.
- A list of transactions that form a valid validation path between the output containing the pseudonym and the provided GPTx.

Depending on the scenario, the size of proof messages can be greatly reduced by including only the address of the output (i.e., a transaction hash in combination with an output index) that contains the pseudonym to be validated. A recipient with efficient reading access to the block chain (e.g., a full Bitcoin node) can then obtain the necessary transactions and form the validation path himself.

6.3.2.3 Verifying a proof

The recipient of a proof message needs to verify the following criteria:

1. The provided transactions form a valid validation path.
2. The transaction containing the pseudonym is included in the block chain.

[6] The purpose of mix transactions is to maintain the unlinkability of pseudonyms across changes and in this way prevent an unambiguous linking of pseudonyms to user identities (which could be inferred from GPTxes). Pseudonym mixing is discussed in detail in Section 6.3.3.

3. The output containing the pseudonym is unspent, i.e., has not been used in a follow-up transaction.

4. The provided GPTx is valid.

Criteria 2 and 3 require reading access to the block chain. For clients with constrained resources, that cannot form full nodes in the Bitcoin network, such access can be provided by querying full nodes (the de facto standard operating mode of end-user Bitcoin clients). Note that the verification of criteria 1 and 2 is sufficient for ensuring that all transactions in the proof message are included in the block chain. Transaction inputs include cryptographic hashes of preceding transactions, so that no fake validation path starting from a transaction on the block chain can be constructed (under the assumption of a secure hash function). The verification of the GPTx (criterion 4) depends on the used IAC mechanism. When using proof-of-burn, for example, the recipient needs to verify that the amount of bitcoin allocated to the marker output of the GPTx meets a previously established burn requirement.

6.3.3 Pseudonym change

As genesis pseudonyms might be identifiable (e.g., due to the funds used for creating them), unlinkable pseudonym changes are necessary for ensuring that pseudonyms hide the identities of their holders. Additionally, pseudonym changes help with the prevention of correlation attacks where pseudonyms are broken following longer periods of observation.

6.3.3.1 Simple change and Bitcoin mixing

As discussed previously, each valid pseudonym is encoded in an output of a transaction on the block chain. Changing a pseudonym involves creating a new transaction that spends that output. However, consecutive transactions on the Bitcoin block chain are, by design, completely linkable to each other. Thus, for ensuring the unlinkability between old and new pseudonyms, pseudonym change transactions must be created in a coordinated fashion so that an external observer cannot determine if two pseudonyms linked on the block chain are also held by the same user.

This challenge is tightly related to the anonymization, or *mixing*, of Bitcoin funds, which has already been tackled in a wide range of works [6, 8, 9, 77, 100]. While the strategy used here is to build upon such works, existing approaches cannot be leveraged directly. First, the unlinkable payment of transaction fees is non-trivial when considering the double role of transaction outputs as encoders of pseudonyms and holders of funds. Second, it must be ensured that only users with valid pseudonyms are able to contribute to mixes and that no entity is able to "steal" access rights, e.g., by manipulating the input and output ordering of formed transactions.

6.3.3.2 Pseudonym mixing protocol

In the following, a specific pseudonym mixing protocol is proposed that is based on the Bitcoin mixing approach *CoinShuffle* [100]. In CoinShuffle, groups of users collaboratively form *mix transactions* to which every user contributes inputs and outputs. The mapping between inputs and outputs is randomized so that both external adversaries and mix group members (for mix groups with 3 or more non-colluding members) cannot conclusive determine who provided funds to which output. In addition to being fully decentralized, CoinShuffle has the benefit of being fully compatible with Bitcoin, requiring only standard output scripts. In comparison to commitment-based approaches like *Xim* [8], CoinShuffle mixes are faster and cheaper, as only one transaction per mix needs to be written to the block chain. By not relying on computation-intensive cryptographic primitives (unlike, for example, *Zerocash* [6]), the CoinShuffle approach is, furthermore, better suited in scenarios with resource-constrained user devices.

The presented pseudonym mixing protocol is divided into four phases that are discussed in the following:

1. *Discovery of mixing partners.* For creating a meaningful mix transaction, at least one additional pseudonym holder is required that also wishes to change his pseudonym. Different approaches are possible for discovering such mixing partners. A generic one involves the establishment of a block chain-based broadcast channel, using OP_RETURN outputs to mark transactions belonging to the channel and include arbitrary announcement data. Pseudonym holders monitor the broadcast channel and either respond to an existing announcement or place a new one themselves. A similar approach to mixing partner discovery is used in [8].

   Alternatively, in some scenarios, suitable mixing partners can be found via side channels. In peer-to-peer systems, for example, pseudonym holders are typically already aware of a number of other participants. In a smart traffic context, nearby vehicles can be discovered using short-range radio beacons. Additionally, in some scenarios, side channels that might weaken the unlinkability of pseudonym mixing must be taken into account. In smart traffic, for example, mixing partners should be spatially close to each other to avoid position-based linking (cf. Section 2.3.4).

2. *Verification of mixing partners.* All members of a mix group must prove the validity of their respective pseudonyms to each other as described in Section 6.3.2. As an additional benefit, this reduces the impact of uncooperative mix group members that block subsequent protocol steps (a form of denial-of-service attack), as malicious pseudonym holders face the danger of being blacklisted (blacklisting is discussed in Section 6.3.4).

3. *Creation of a mix transaction.* This step is largely based on the CoinShuffle protocol [100]. The participants within a mix group exchange inputs and outputs (containing freshly generated Bitcoin addresses), out of which one transaction is cooperatively formed. Every participant verifies if his own inputs and outputs are correctly included in the resulting transaction and, if so, provides the

necessary signature for his inputs. The exchange of inputs, outputs and signatures is performed anonymously using techniques reminiscent to decryption mix nets (cf. Section 2.4.6). Consequently, given a group size larger than 2 and assuming that group members do not collude, no link between inputs and outputs can be established.

As an important modification to CoinShuffle, a different mechanism is used for distributing the value of inputs and paying transaction fees. Every participant is allowed to contribute only one output to the resulting transaction. The value of outputs is not chosen freely by mix group members, ensuring that they remain with a similar amount of bitcoins after the mix, but divided equally between all outputs. In this way, the value-based linking between inputs and outputs is prevented. The value v used for all outputs is equal to the total value available after the payment of the transaction fee f divided by the number of participants m. With v_i denoting the value associated to the input with index i, we arrive at the following formula:

$$v = \frac{(\sum_{i=0}^{m} v_i) - f}{m}$$

The resulting mix transaction is broadcast to the Bitcoin network. Once it has been included in the block chain, the old pseudonyms become invalid and the new ones (encoded in the outputs of the mix transaction) valid.

4. *Construction of new proofs.* Once all pseudonyms are exchanged, each participant needs to construct a proof for his new pseudonym. The proofs for all old pseudonyms participating in the mix have already been exchanged during the verification of mixing partners. According to the validation path logic introduced in Section 6.3.2, every new proof consists of one of these old proofs in combination with the currently formed transaction. Which proof should be used is determined by the index of the output containing a user's new pseudonym - if it has the index i, the proof starting with the output referenced in the i'th input of the transaction is used. It is a common case that users use a different genesis pseudonym for validation after each mix.

   Figure 6.5 depicts an example of such a path change. The holder of the marked pseudonym in the mix transaction *mix tx 2* participates in *mix tx 4*. He contributes input 2 and output 1 to that transaction. Thus, his new pseudonym has an entirely different validation path with a different GPTx.

6.3.3.3 Parametrization of mixing protocol

Two configurable parameters influence the achievable anonymity set growth per mix, the possible number of mixes for a given pseudonym value and the speed of pseudonym changes. These are the desired *mix group size* and the *acceptable difference* to the pseudonym values of other mix group members. Larger mix groups lead to higher anonymity gains per mix. However, larger minimum mix group sizes

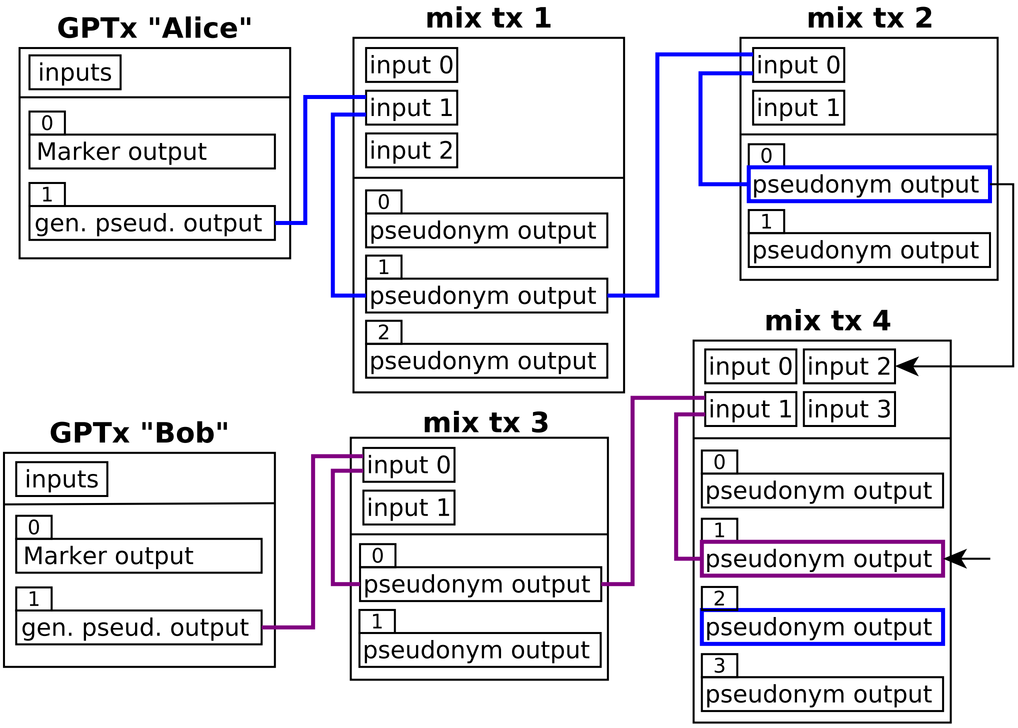


Figure 6.5 Validation path change after mix.

also lead to longer waiting times until a group has been formed. A low acceptable difference leads to a lower number of possible mixing partners and, therefore, also increases waiting times. Note that there is no reason to refuse mixing with a pseudonym whose value is higher than one's own, as this would result in a value gain. In Section 6.6, the impact of different mix group sizes and acceptable difference values on anonymity, costs and communication overhead is investigated in detail.

Concerning the speed of pseudonym changes, instant changes would be ideal. However, coordination with other users is necessary and mix transactions must be written to the block chain. These conditions lead to unavoidable waiting times. An approach for enabling fast pseudonym changes is to allow each user to simultaneously hold two or more pseudonyms. If, for example, GPTxes spawn two instead of one pseudonym, one pseudonym can be actively used while the pseudonym mixing protocol is conducted opportunistically using the other. Once a change is appropriate (e.g., for scenario-specific reasons like passing a mix zone), the roles of the pseudonyms can be swapped.

6.3.4 Blacklisting

In many scenarios, it is necessary for users to be punishable upon malicious behavior. Even in the context of BitNym itself, it is beneficial if participants that deliberately disrupt the pseudonym mixing protocol can be blacklisted for a certain time. Blacklisting is easily supported by BitNym. Pseudonyms can be publicly marked as malicious, e.g., using a block chain-based broadcast channel as in [8]. Blacklisted pseudonyms will fail the pseudonym validation step. Additionally, once a pseudonym is blacklisted, other participants will refuse to cooperate with it in the scope of pseudonym mixes. Thus, the pseudonym holder is permanently blacklisted, respectively needs to create a new genesis pseudonym (which, dependent on the used IAC mechanism, implies high costs or is even impossible).

Open challenges with blacklisting include ensuring that the blacklisting mechanism cannot be used for maliciously censoring users. A voting-based approach is a possible solution for making correct blacklisting decisions without a TTP. For example, pseudonyms might need to be reported a certain number of times before being considered malicious. By layering upon BitNym, it can be assumed that participating pseudonymous users are non-sybil and not actively colluding with malicious intent. Thus, fair voting and consensus procedures become possible. If a darknet is maintained for social graph-based IAC (cf. Section 6.3.1.2), it can be applied for speeding up the collection of votes and making it harder for malicious users to remain unpunished. Specifically, victims can recruit voters from their social circle by distributing proofs of a malicious user's misbehavior.

It must also be ensured that punishment cannot be evaded by carefully timing pseudonym changes. Lock times per pseudonym might need to be enforced, during which no changes are possible.

6.4 Sybil-resistance analysis

This section analyses whether BitNym effectively prevents sybil attacks. It is proven that for a given set of genesis pseudonyms G and any given time, the set of valid pseudonyms P is such that $|P| \leq |G|$. If $|P| \leq |G|$, the sybil-resistance of BitNym depends on the sybil-resistance of the used IAC mechanism (e.g., proof-of-burn).

Only one assumption is required: that, at any given time, the Bitcoin block chain contains only valid Bitcoin transactions. More specifically, it must be guaranteed that transaction outputs can be spent only once and that transaction inputs can reference only one output. This assumption reflects the reality of the Bitcoin network. Blocks containing invalid transactions are ignored by correctly functioning clients.

Attempting the actual proof that $|P| \leq |G|$, we now assume that a sybil attack is possible and $|P| > |G|$.

In order for a pseudonym p to be valid, i.e., in order for it to hold that $p \in P$, p must be encoded in an unspent transaction output on the Bitcoin block chain. Additionally, a validation path $\text{path}(p)$ (based on the Bitcoin transaction graph) must be presented that leads to a genesis pseudonym g, or:

$$\forall p \in P \, \exists g \in G : g \in \text{path}(p)$$

If $|P| > |G|$, at least two currently valid pseudonyms must exist that share the same genesis pseudonym for validation, or:

$$\exists p_1, p_2 \in P,\ p_1 \neq p_2 : |\,\text{path}(p_1) \cap \text{path}(p_2) \cap G| \geq 1$$

Let $g \in G$ be the genesis pseudonym for which it holds that $g \in \text{path}(p_1)$ and $g \in \text{path}(p_2)$. Validation paths are formed deterministically by linking, within transactions, each input to the output with the same index in the transaction, and, between transactions, each output to the input that spends it (cf. Section 6.3.2.1). Therefore, only one possible path through g exists and only one output in this path can be unspent at any given time. So, if $\text{path}(p_1) \subset \text{path}(p_2)$, the output in which p_1 is encoded must have been spent (recall that $p_1 \neq p_2$). Therefore, p_1 is not a valid pseudonym ($p_1 \notin P$), which contradicts its original definition. Analogous reasoning applies for the case that $\text{path}(p_2) \subset \text{path}(p_1)$. It follows that $|P| > |G|$ is wrong and, therefore, $|P| \leq |G|$.

Assuming a sybil-proof IAC mechanism and a correctly functioning Bitcoin network, BitNym is therefore sybil-resistant.

6.5 Privacy analysis

A privacy analysis of BitNym is conducted in the following, leaning on the approach outlined in Chapter 3. The assumed system model and adversary capabilities are first clarified. Several analysis steps are then applied. Based on the privacy threats addressed through them, the steps are separated into two groups:

1. Detection and disclosure.

2. Linking and identification.

6.5.1 System and adversary model

A large population of honest users is assumed, for which following assumptions apply:

- The majority of human users is honest and not colluding with malicious intent (i.e., with the adversary). This assumption reflects the main trust assumption behind cooperative services (cf. Section 2.4.1).

- Each user possesses a complete, authentic and up-to-date view over the Bitcoin block chain. It is abstracted away from the fact that users might (due to resource constraints) use a *simple payments verification* (SPV) approach for interfacing with the block chain instead of maintaining a full Bitcoin node.

- Users can broadcast new transactions to the Bitcoin network. New transactions are, as long as a modest customary fee is provided, always quickly included in newly mined blocks (and, in this way, published on the block chain).

- Users are able to communicate pseudonymously with each other, using, e.g., PTP (cf. Section 2.6.2). The adversary cannot deanonymize such communication or compromise its confidentiality, integrity or authenticity.

- Each user has created at least one genesis pseudonym and engages in periodic pseudonym changes. Concerning the value used for creating genesis pseudonyms and the implication of value loss on privacy properties, a further discussion is conducted in Section 6.5.3.3.
- A sybil-proof IAC mechanism is used. Consequently, all currently active pseudonyms are assumed to be sybil-free. In effect, the number of currently active pseudonyms that the adversary is able to control is limited by the number of users with which he colludes.

The assumptions from Section 6.2.3 apply. This includes the assumption that adversaries can't tamper with the state of the Bitcoin block chain in a malicious way, i.e., that the security of Bitcoin is guaranteed. The Bitcoin network is assumed to operate correctly at all times.

Unlike in other parts of this thesis, the goal of the adversary in this chapter is not the collection of identifiable location samples, as no location samples are generated at all by the considered system. Location samples might, however, be shared under pseudonyms realized through BitNym. Therefore, the main goal of the adversary here is to break pseudonyms, i.e., discover the identity of their holders. Following capabilities are assumed for the adversary in the following:

- He has full reading access to the Bitcoin block chain (which is public).
- He can control a small number of currently active pseudonyms.
- He can break all genesis pseudonyms, i.e., identify all published GPTxes[7].

6.5.2 Detection and disclosure

Starting with the actual analysis, the following questions are considered in the scope of this section:

1. What knowledge does the adversary gain from *passive observation* during normal operation?
2. What *actions* are possible for the adversary within the system model?
3. Do the actions possible for the adversary enable an *increasing* of his *knowledge* and if yes - in what way?

6.5.2.1 Knowledge from passive observation

The adversary is aware of all transactions on the Bitcoin block chain. Therefore, he knows both all currently active pseudonyms and the transaction graph (or pseudonym changing graph) that led up to them. As defined in the adversary model,

[7] Depending on the used IAC mechanism, this assumption may only hold for adversaries with significant additional knowledge. Bitcoin addresses are pseudonymous and Bitcoin transactions therefore difficult to link to user identities.

he is additionally aware of the identities behind all GPTxes on the block chain. He might also be aware of the identities behind other pseudonyms, i.e., due to additional context knowledge. In Section 6.6.2, a simulation study is presented that aims at generating realistic changing graphs that can be used for the subsequent analysis.

Assuming that the small number of identities that the adversary can control act like regular users, they will engage in pseudonym mixing with others. For mixes in which some of the participating pseudonyms collude with the adversary, he has an information gain compared to observing only the transaction graph. Namely, by being able to identify some of the outputs of such transactions, he lowers the amount of uncertainty caused by the mix. However, as the adversary is able to control only very few active pseudonyms, the overall information gain (taking into account all mix transactions) is expected to be insignificant and is therefore ignored in the following.

6.5.2.2 Possible actions

The adversary can interact with the block chain, e.g., write arbitrary valid transactions to it. However, only valid pseudonym transactions and GPTxes are considered by regular users. The posting of other types of transactions on the block chain has no impact exceeding that of regular operation and is ignored in the following.

The adversary can communicate with pseudonymous users, attempting to identify them using a side channel. However, this exceeds the context considered in this chapter and is not considered further.

The adversary can attempt to attack the functionality of the mechanism used for discovering mixing partners. However, this won't be considered further, as this discovery mechanism is scenario-specific and not fully specified in the scope of this chapter. In a similar context, the adversary can attempt a targeted attack on a specific pseudonym. He can attempt to enter the mix groups of this pseudonym, therefore lowering the entropy of its mixing. This attack vector too depends on the employed discovery mechanism for mixing partners. If mixing partners are chosen randomly and no sybil attack is possible, an infiltration of mix groups will be difficult. An approach like [38] can additionally be used for preventing attacks on the randomness of mixing partner selection.

Lastly, the adversary can attempt a denial-of-service attack on the pseudonym mixing protocol, preventing users from effectively changing their pseudonyms. However, deliberate disturbances can be detected by participating users, leading to a blacklisting recommendation for the involved pseudonyms. Additionally, the impact of such an attack will likely be severely restricted by the number of active pseudonyms available to the adversary.

6.5.2.3 Increasing knowledge

As was identified in the previous section, none of the actions that exceed regular operation and are possible within the considered system model and scope, are expected to have any significant impact and, hence, lead to notable benefits for the adversary. Consequently, no possibilities for increasing the adversary's knowledge through additional actions must be considered further.

6.5.3 Linking and identification

Based on the previous analysis, the adversary is aware of the complete pseudonym changing graph and knows the identities behind all GPTxes[8]. The question approached in this section is now: what can he learn from that data? And, more specifically: how well suited is that data for achieving the goal of identifying individual pseudonyms? Based on the approach outlined in Section 3.3, the following possibilities should be considered in this context:

1. Direct identification.

2. Linkability.

3. Identification attacks based on linked data.

Approach 1 is only possible for GPTxes and if no pseudonym mixing is applied during pseudonym changes (which would allow the non-ambiguous linking of pseudonyms and GPTxes). Bitcoin transactions in general and pseudonym transactions in specific do not contain any form of identifiable data. Approach 3 is not considered, as it is targeted at scenarios in which richer data (e.g., location samples) is included in considered data items, or in cases where additional context knowledge is assumed.

In the following, the focus is on discussing linkability. More precisely, the feasibility of linking pseudonyms to GPTxes (which, within the assumed adversary model, implies their identification) is considered. Suitable metrics for this threat are worked out. Additionally, the impact of transaction fees and pseudonym funding is discussed.

The metrics introduced here were used in a series of simulation studies (described in Section 6.6.2) that provide empirical insights about the linking threat.

6.5.3.1 Anonymity metrics

A metric is required for quantifying the degree to which pseudonyms in BitNym are unlinkable to GPTxes created by their holders. Such a metric can be constructed based on the notion of anonymity as defined by Pfitzmann et al. [94], i.e., the indistinguishability within a set of subjects - the *anonymity set*. The size of the anonymity set of a user is a common metric for the non-identifiability of his pseudonym. In the following, it is referred to simply as *anonymity*.

In the context of BitNym, different approaches for defining anonymity are possible that depend on the assumed goals of the adversary. When considering *backward*

[8] For analysis, and without loss of generality, genesis pseudonyms act as placeholders for any pseudonym that is already identifiable by the adversary (e.g., via additional knowledge).

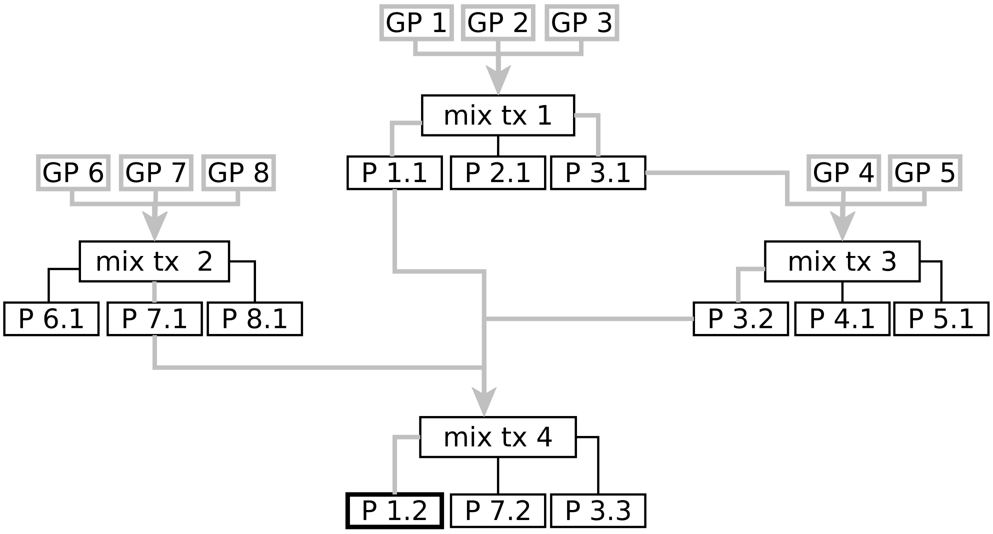


Figure 6.6 Backward anonymity increase for creator of genesis pseudonym *GP 1*.

anonymity, the adversary aims at determining which genesis pseudonym was previously held by the holder of a pseudonym. A somewhat opposite approach, *forward anonymity*, assumes an adversary that, given a specific inactive pseudonym (e.g., a genesis pseudonym), wishes to determine the pseudonyms which that pseudonym's holder held afterwards. Based on the adversary goal of identifying pseudonym holders, the focus here is on backward anonymity. Most of the subsequent analysis applies for forward anonymity as well.

6.5.3.2 Backward anonymity

The *backward anonymity set* of a pseudonym is the set of genesis pseudonyms that could have been created by that pseudonym's holder. Consequently, the *backward anonymity* of a pseudonym is the size of that set. As was discussed previously, the adversary is assumed to know the identities behind all genesis pseudonyms. Therefore, by measuring backward anonymity, the identifiability of pseudonyms is quantified.

Genesis pseudonyms have a backward anonymity of 1. It is usually increased following pseudonym mixes, as the backward anonymity set after a mix is the union of all anonymity sets of the participating pseudonyms. Given a pseudonym mix transaction with the set of participating pseudonyms M and the anonymity sets A_p for all $p \in M$, the *anonymity increase* Δa_p for a $p \in M$ can be written as:

$$\Delta a_p = \left| \bigcup_{q \in M} A_q \setminus A_p \right|$$

The increase of backward anonymity across pseudonym mixes is also depicted in Figure 6.6. At first, the creator of the genesis pseudonym *GP 1* has no anonymity.

Through the mix transaction *mix tx 1* his anonymity set increases to 3, because the pseudonyms *P 1.1*, *P 2.1* and *P 3.1* cannot be unambiguously linked to any of *GP 1* to 3. After *mix tx 4*, the anonymity set of the creator of *GP 1* increases to 8, as *P 1.2* can be held by the creator of any one of the 8 GPTxes.

6.5.3.3 Transaction fees and pseudonym value

In the predominant majority of cases, the creation of block chain transactions requires the provision of transaction fees. If fees are not payed from pseudonym inputs as proposed in Section 6.3.3, they need to be provided from other sources via additional transaction inputs. If these additional inputs are not anonymized, the breaking of pseudonyms becomes possible. For example, if a user contributes fees to two linked mix transactions, it becomes clear which pseudonym belonged to the user between mixes.

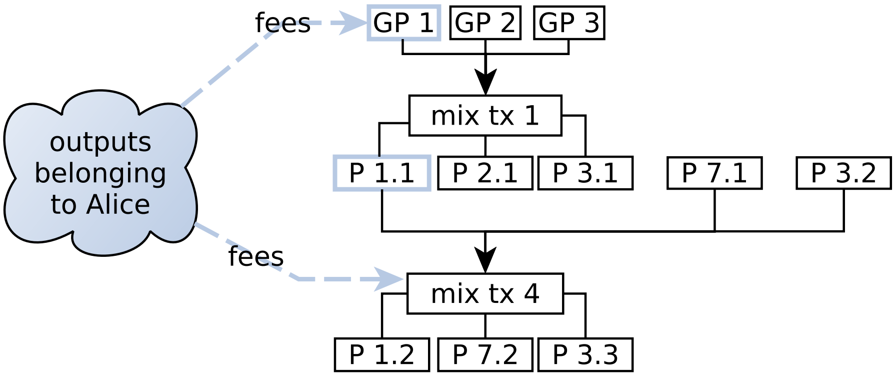


Figure 6.7 Anonymity loss through external payment of transaction fees.

An analogous scenario is depicted in Figure 6.7. Here, the genesis pseudonym *GP 1* is identifiable as belonging to Alice, as it is funded by one of her outputs. By sending funds to the mix transaction *mix tx 4*, she additionally makes the pseudonym *P 1.1* identifiable. Based on the transaction graph, and assuming that Alice has created only one genesis pseudonym, no other hypothesis about the identity behind *P 1.1* is possible.

BitNym avoids the external funding of mix transactions by using the value stored in pseudonym outputs. More complex alternatives to this approach are possible. Pseudonym holders can contribute to pools that are then used for paying fees. Or a parallel coin mixing mechanism can be implemented that ensures that fee payment funds are anonymous. In addition to increasing complexity, the adoption of such approaches will likely result in a higher average fee requirement (mix transactions will require additional inputs, increasing their size and, therefore, required fees). At the same time, its relative benefit is questionable.

BitNym also supports the *recharging* of pseudonyms. Here, a non-mixing change is conducted, with additional inputs increasing the pseudonym's value. For analysis purposes, this is similar to creating a new genesis pseudonym - the pseudonym created in such a transaction might become linkable to the holder of the provided funds. If the provided funds are linkable to the pseudonym holder, however, recharging doesn't only make the recharged pseudonym identifiable, but also removes its holder from the anonymity sets of all currently valid pseudonyms (assuming a one-to-one relationship between valid pseudonyms and users). The anonymity level of former mixing partners of the recharging user might, in this way, be lowered by 1.

6.6 Evaluation

BitNym was evaluated by implementing a proof-of-concept prototype and measuring its communication and computation overhead. Additionally, a simulation study was performed for evaluating the effect of pseudonym mixing in scenarios with multiple thousand participants.

6.6.1 Prototype

For gaining insights about the overhead and applicability of BitNym in practical scenarios, a proof-of-concept prototype was implemented that is compatible with the Bitcoin network. The prototype is capable of creating GPTxes using proof-of-burn as an IAC mechanism, and performing the pseudonym mixing protocol with mixing partners given as input. The discovery of mixing partners is not part of the prototype, and neither is blacklisting.

6.6.1.1 Implementation

For communicating with the Bitcoin network and accessing the Bitcoin block chain, the prototype uses the *BitcoinJ* library[9] (version 0.12). The SPV mode of BitcoinJ is used, i.e., the prototype doesn't maintain a complete block chain but contacts full nodes for required information. Thus, the prototype is suited for resource-constrained devices. Cryptographic primitives required for the mixing protocol are realized using the cryptographic library *Bouncy Castle*[10] (version 1.51).

As one of the steps of validating a pseudonym, it must be verified that the transaction output holding the pseudonym is unspent. The verification can be performed efficiently by checking that the output is part of Bitcoin's *unspent transaction output* (UTXO) set that is maintained by all full nodes. An extension to the Bitcoin protocol has been proposed for allowing SPV clients to query full nodes for entries in this set[11]. While already supported by some nodes at the time of writing (most notably, nodes based on the *Bitcoin XT* implementation of Bitcoin[12]), the proposed extension has the main drawback that responses to UTXO queries are not authenticated in any way. Full nodes must be trusted for providing a correct view on the UTXO set and can, if malicious, present a spent output as unspent. In order for SPV queries for entries in the UTXO set to be secure without trusting the queried full node, changes to

[9] https://bitcoinj.github.io/

[10] https://www.bouncycastle.org/

[11] https://github.com/bitcoin/bips/blob/master/bip-0064.mediawiki

[12] https://bitcoinxt.software/

Bitcoin are necessary. A promising approach is for mined blocks to include commitments to the current UTXO set, using a hash tree [80, 98]. SPV clients can determine (with a reasonable degree of certainty) whether a block header received from a full node corresponds to a block actually included in the block chain. Therefore, if block headers contain a commitment to the UTXO set, querying nodes can verify received UTXO entries against this commitment.

UTXO checks were deactivated in the presented prototype. Instead, an additional block is requested from a full node to approximate the latency of making a UTXO query. Not verifying that the transaction output holding the pseudonym of a communication partner is unspent has following implications:

- A malicious user can present one of his older pseudonyms that is no longer valid (i.e., its corresponding transaction output has been spent). This ability enables a form of sybil attack in which a malicious user can use all pseudonyms that he ever held simultaneously. However, the practical impact of such an attack is likely limited, as transmitted validation paths will often conflict with a false claim about the state of older outputs.

- It is not possible to completely validate the pseudonyms of potential mixing partners. However, if one of the mix group members is malicious and tries to contribute to the mix transaction from a spent pseudonym output, this will be easily detected by miners and the published mix transaction will not be published on the block chain. A malicious user can, therefore, only block individual mixes from completing, delaying pseudonym changes.

A possible workaround to the issue of securely finding out whether a given pseudonym output has been spent without needing to trust (or maintain) a full node, is to render pseudonym outputs temporarily unspendable. In practice, this can be achieved by including Bitcoin's *OP_CHECKLOCKTIMEVERIFY* opcode into pseudonym outputs, locking them so that they cannot be spent before the supplied time. Another approach is to monitor all active pseudonym addresses, effectively maintaining a subset of the UTXO set containing only pseudonym outputs. This approach hast the benefit of making the transmission of (potentially long) validation paths in proof messages unnecessary. Neither of these approaches was implemented in the presented prototype.

6.6.1.2 Performance

In the following, evaluation results gathered with the prototype are presented. The main goal is to give a first impression about the expectable performance. The time requirement of pseudonym changes was not evaluated, as it is dominated by the time until the final mix transaction is included in the block chain (around 9 minutes on average [47]). Other contributing factors are the time until mixing partners are found (the discovery of mixing partners was not part of the prototype) and the CoinShuffle mixing protocol (evaluated in [100]).

The focus here is on the evaluation of pseudonym validity checks. These need to be performed not only by pseudonym holders, but also by all entities wishing to validate BitNym pseudonyms. On a regular notebook computer (2.5 GHz CPUs, the tested functionality was single-threaded), verifying a proof consistently took between 130 ms and 550 ms (with a median of 198 ms) for different proof messages. Proof sizes had no discernible effect on validation duration. The majority of time is spent on requesting block data from a Bitcoin full node (for verifying that the pseudonym transaction was included in the block chain) and verifying, with the help of a full node, if the pseudonym output is unspent (which was approximated by requesting an additional block, cf. Section 6.6.1.1).

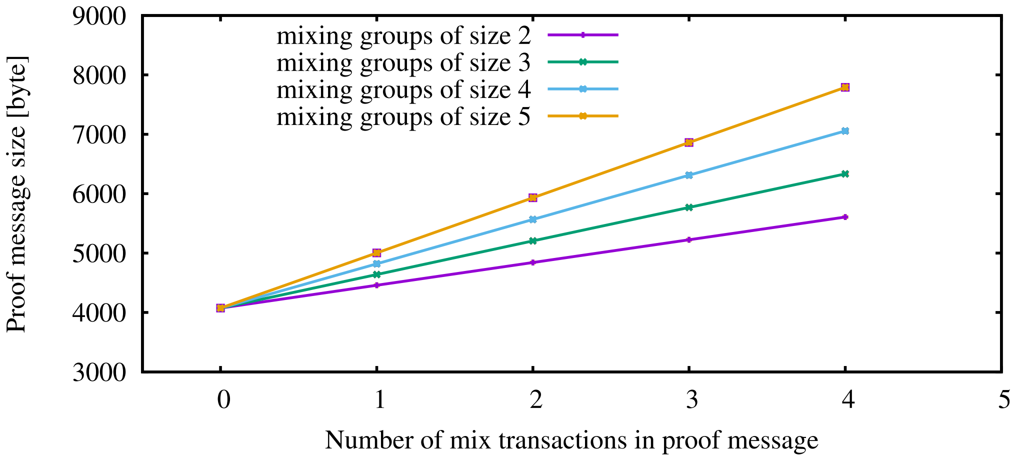


Figure 6.8 Size of proof messages.

The sizes of proof messages generated by the prototype are depicted in Figure 6.8. The sizes of proof messages grow linearly with the number of (mix) transactions included in the proof, starting from 4075 bytes for a proof containing only a GPTx and increasing by 380 bytes for a mix transaction with a mix group size of 2. Larger mix group sizes lead to a larger increase in proof message size per included transaction, e.g., the increase is around 570 bytes for a mix transaction with 3 participants. The standard serialization routines of Java were used in the prototype, so that improvements to the proof message sizes are likely possible. Still, even with this implementation, proofs containing more than hundred mix transactions are possible at a proof message size of below 50 kilobytes.

6.6.2 Large-scale pseudonym mixing

For answering questions about the scalability of BitNym and the levels of anonymity achievable when using it, a series of simulation studies was conducted.

6.6.2.1 Simulation environment

A lightweight time-discrete simulator was developed for evaluating different properties of pseudonym mixing on a macroscopic scale. The simulator can be used for measuring, amongst other things, the influence of mixing on the degree of anonymity and the size of proof messages (affecting communication overhead).

No actual interfacing with the block chain is simulated. Time is divided in slots. A time slot corresponds to one block interval, i.e., the time interval between two subsequent blocks on the block chain. During each slot (simulated) users perform one of following actions:

- create new GPTxes (during the beginning of the simulation)
- announce that they are looking for mixing partners
- answer existing announcements, initiating a pseudonym mix
- recharge pseudonyms with depleted value for allowing further mixing

Simulation parameters can be divided into scenario-specific and BitNym-specific. Scenario-specific parameters include:

- The pseudonym *start value* for all users. A value of 200 000 satoshi[13] was used in all presented simulations (0.002 bitcoin; worth around 0.8 euro at the time of writing).
- The *number of users*. Unless noted otherwise, populations of 10 000 users were simulated.
- The average pseudonym *change rate* for all users, i.e., at which average intervals users announce a desire to form a mix transaction. An average of 288 time slots was used in all presented simulations. This corresponds to 48 hours (time slots correspond to Bitcoin block intervals, which average to 10 minutes).
- The *change rate deviation*. This is the standard deviation of change intervals from the average change rate. A value of 72 was used by default, representing a high variation. The impact of different values was evaluated as well.

BitNym-specific parameters include:

- The desired *mix group size*.
- Whether mix group sizes should be *limited* to the mix group size or if larger groups should be allowed.
- The *acceptable difference* to the pseudonym values of mix group participants. For example, with an acceptable difference of 50 000 satoshi, a pseudonym with a value of 100 000 satoshi will mix only with pseudonyms with a value between 50 000 and 150 000.

[13] 10^8 satoshi amount to 1 bitcoin.

At the beginning of each simulation, as a *warm-up phase*, each user creates a GPTx. Creation times are uniformly distributed within a period corresponding to the average expected *lifetime* of a pseudonym. The lifetime of a pseudonym is determined by the configured pseudonym start value and the value subtracted from pseudonyms for the payment of transaction fees.

As the posting of transactions on the Bitcoin block chain requires the payment of fees, in every mix transaction, an average of 5000 satoshi is subtracted from the pseudonym values of all participants. The value of 5000 satoshi is chosen as a conservative estimate of the fees required in reality. By the fallback fee calculation logic of the original Bitcoin client software, for example, 20 000 satoshi are payed per kilobyte of transaction size[14]. Based on the prototype results, each participant adds around 190 B to the resulting mix transaction. A per participant fee of 5000 satoshi therefore includes a 24% margin of safety, ensuring a quick inclusion on the block chain. After the pseudonym's value has dropped to below 5000, it is considered *dead* in the simulated context. For a configured pseudonym start value of 200 000 satoshi, the average pseudonym lifetime is, thus, 40 changes.

Once a pseudonym dies, it is recharged by its holder. A regular transaction (i.e., not a mix transaction) is created that increases the pseudonym's value by the pseudonym start value. The exact point in time of the recharge is determined using the same logic as the decision about when a new mix transaction should be initiated, i.e., randomly based on the configured change rate and change rate deviation.

Pseudonym change and mixing is not implemented in detail. Only the outward effect of the mixing protocol outlined in Section 6.3.3 is reproduced. Mix groups are formed centrally by the simulator in every time step, based on the published announcements. The simulator is honest and honors all announced requirements, i.e., concerning the desired mix group size and the acceptable difference to mixing partners. If multiple possibilities for forming mix groups exist, groups are formed randomly. Once a mix group is formed, the corresponding mix transaction is published instantly, i.e., in the same time slot.

Two basic types of results are output by the simulator. First, such where the development of a measured value is shown relative to the elapsed simulated time. Here, the initial warm-up phase is clearly visible. The second type of results expresses the relationship of a value to the number of mix transactions performed by users. Measurements for this second type of result are made just after completing a mix. In order to reduce the impact of the initial creation phase on this result type, only pseudonyms created 3 average lifetimes after the start of the simulation are considered.

After the first 3 lifetimes, the simulation continues for another 10 lifetimes so that a total of 13 lifetimes is simulated. Based on a pseudonym start value of 200 000 and a change rate of 288 time slots, this corresponds to 149 760 time slots or 1040 days.

[14] `https://github.com/bitcoin/bitcoin/blob/v0.12.0/doc/release-notes.md`

All results presented in the following represent averages of 4 simulation runs per parameter configuration.

The simulation parameters used for generating results figures in the remainder of this chapter are summarized in Table C.2 (Appendix C).

6.6.2.2 Selection of BitNym-specific parameters

An initial selection of BitNym-specific parameters was performed as a base for subsequent investigations. Recall that, based on the simulation setup, following BitNym-specific parameters are configurable:

- minimum mix group size
- whether mix group sizes should be limited
- acceptable difference to the pseudonym values of mix group participants

Towards the goal of maximizing backward anonymity, an acceptable difference of 200 000 satoshi was initially chosen. This value corresponds to the configured pseudonym start value and, therefore, implies that no value-based constraints will be put on potential mixing partners. Constraints on potential mixing partners would reduce the pool from which mixing partners are chosen and are therefore expected to increase the likelihood that mixes leave the anonymity sets of participants unchanged.

The parameters related to group size have direct implications for the sizes of pseudonym proofs (a likely source of overhead in practice). With a mix group size of g, the backward anonymity of users is increased by up to factor g after each mix. At the same time, the proof sizes of mix participants increase by an average of $190g$ byte (based on the results from Section 6.6.1.2). For an ideal scenario with an infinitely large user population, the anonymity level $a()$ per 190 byte block x is expected to be:

$$a(x) = g^{\frac{x}{g}} = e^{\frac{x \ln g}{g}}$$

Finding a value for g that maximizes anonymity per byte boils down to finding a g that maximizes $\frac{\ln g}{g}$. Using standard function analysis, we arrive at $g = e$. However, e is not a natural number and, therefore, not a suitable value for BitNym's mix group size parameter. Substituting with the two closest natural numbers 2 and 3, we find that $\frac{\ln 2}{2} < \frac{\ln 3}{3}$ and, therefore, conclude that a mix group size of 3 maximizes anonymity per byte in an ideal scenario.

However, a realistic scenario differs from the ideal case in that the number of users, and therefore the maximum reachable anonymity set, is limited. A simulation study was performed using varying configurations related to group size. The backward anonymity of each node was determined in each time step, and averaged. This average anonymity level was then divided by the average proof size length of all pseudonyms valid in that time step. The translation of validation path length (in number and group size of mix transactions) to proof message sizes (in byte) was based on the results gathered with the prototype[15]. Figure 6.9 depicts results from this study for group sizes *gs* of exactly 2, 3 and 5 as well as a configuration with an unlimited maximum group size and a minimum group size of 2. It can be seen that mix groups of size 2 outperform larger mix groups. Also, avoiding a limit on group sizes (which implies larger average group sizes) causes lower anonymity per byte values than configurations with group sizes fixed to 2, 3 or 5. Therefore, for all subsequent simulations discussed in this chapter, mix groups of exactly size 2 were used. A further benefit of this parametrization is that mix groups can be formed faster, as only one mixing partner must be found.

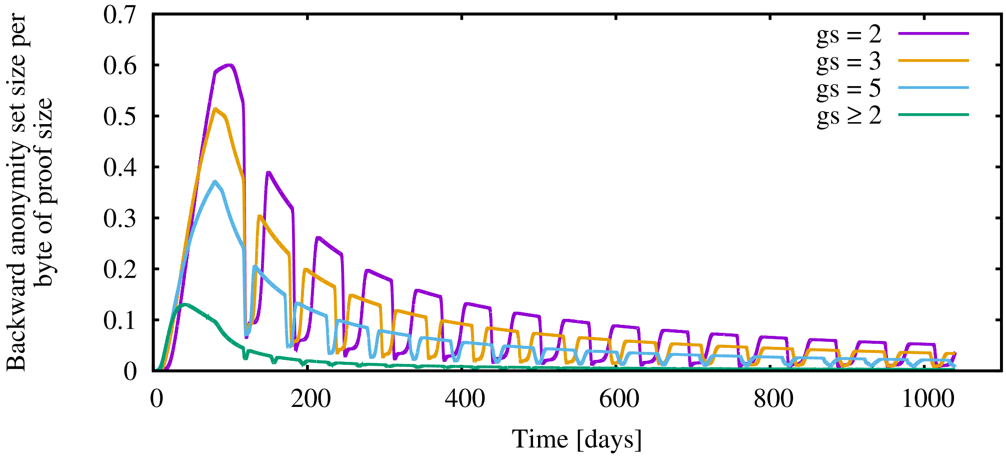


Figure 6.9 Backward anonymity per byte of pseudonym proof, for different configurations of mix group size *gs*.

Figure 6.9 also hints at the existence of periodic drops in the measured anonymity levels. Drops in anonymity are due to the fact that, once a dead pseudonym becomes recharged, its holder is removed from all anonymity sets in the simulation. His own anonymity set is reset to include only himself. With both of this measures, the assumption is modeled that recharges are fully linkable to user identities. As the depicted anonymity levels are averaged across the whole user population, significant drops in the displayed values suggest the existence of synchronization effects. The existence of synchronization effects is also suggested by Figure 6.10. The figure depicts the number of currently alive (i.e., with a value of more than 5000 satoshi) pseudonyms during each time slot of the simulation. The majority of pseudonyms die and become recharged during a comparably short time frame. This observation suggests that their pseudonym values synchronize and are nearly identical near the end of their lifetime. The effect that a significant portion of nodes die in close succession to each other is less strongly expressed in a configuration without a maximum limit on mix group size (gs $\geq$ 2), suggesting that such a configuration leads to less synchronization effects and, thus, more stable anonymity levels.

[15] 4075 bytes for GPTxes and 190 bytes for each change and participant (cf. Section 6.6.1.2).

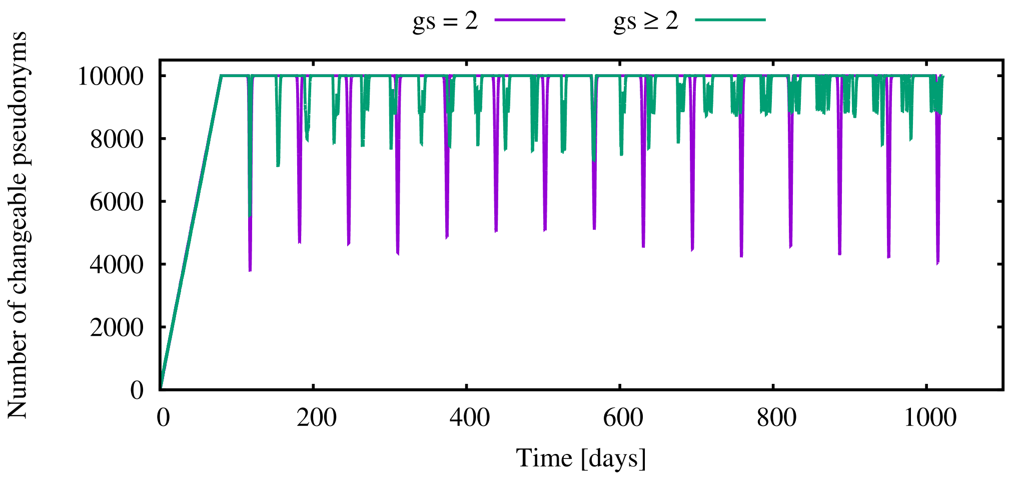


Figure 6.10 Number of alive pseudonyms, for different configurations of mix group size *gs*.

However, any potential benefits in terms of stability or the rate of anonymity increase must be seen in perspective to the simultaneously induced overhead. For additional illustration of the latter, Figure 6.11 depicts the average proof sizes measured during the course of the simulation study described at the beginning of this section. After 1000 simulated days (144 000 time slots), the measured average proof size for the configuration with unlimited mix group sizes is nearly an order of magnitude larger than for the configuration with a mix group size of exactly 2.

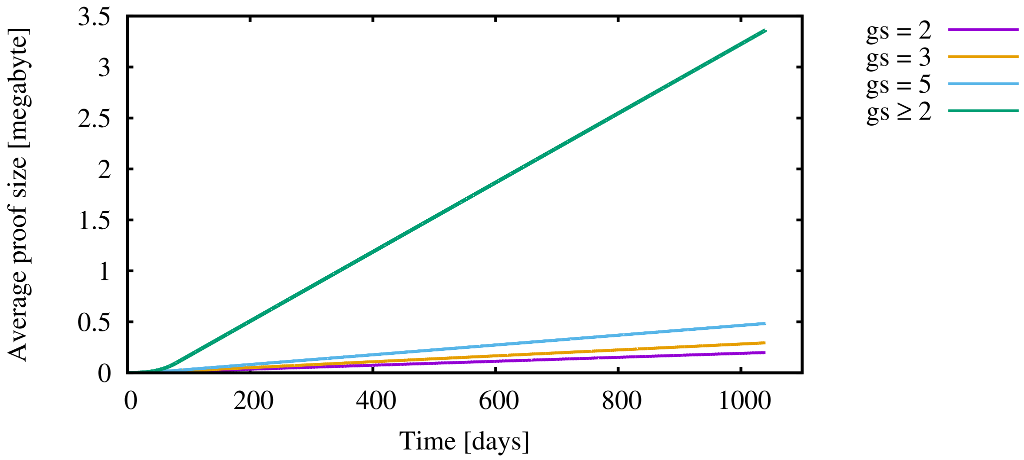


Figure 6.11 Average proof size, for different configurations of mix group size *gs*.

6.6.2.3 Further investigations

In the following, the impact of further parameters on backward anonymity is investigated, namely change rate deviation, acceptable difference and number of users.

The following figures depict the backward anonymity of the average user in relation to the number of mix transactions in which he participated since his last recharge. At change 40, which is marked as a vertical line in the diagrams, pseudonyms, on average, run out of funds. The values after and closely before 40 are thus less representative as only few pseudonyms make it past the 40th change. This effect is explicitly depicted in Figure 6.12. It depicts the percentage of pseudonyms that are still alive after a given number of mix transactions since their last recharge. The figure depicts values for two different configurations of acceptable difference *ad* to mixing partners. The first plotted configuration (ad = 200 000) corresponds to the parametrization recommended in Section 6.6.2.2. It can be seen that the majority of pseudonyms run out of funds before their 40th mix transaction and that less than 5% survive longer than 70 mixes. With the second configuration (ad = 100 000), achieved lifetimes deviate less strongly from the expected value of 40. No lifetimes of above 50 mix transactions were achieved. Results obtained with lower values for the acceptable difference were indistinguishably similar (for the plotted metric) to the ones obtained for an acceptable difference of 100 000 satoshi, and are therefore not shown.

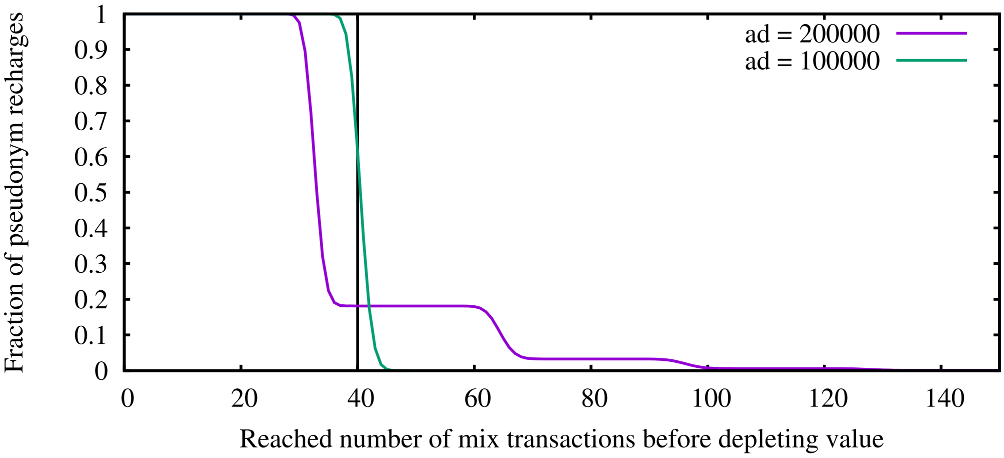


Figure 6.12 Cumulative distribution of pseudonym lifetime after a recharge (i.e., of time until a new recharge is required), for different configurations of acceptable difference *ad* (in satoshi).

Figure 6.13 depicts the impact of change rate deviation *crd* (the standard deviation of change intervals in simulation time slots) on the development of anonymity sets. Recall that the parametrization used in the rest of this chapter uses a value of 72 for this parameter (crd = 72). The aforementioned synchronisation effects, due to converging pseudonym values leading to closely clustered recharging times and, therefore, periodic drops in measured anonymity, are clearly visible. Both drops and rises in anonymity can be witnessed. Based on the obtained results, it is furthermore difficult to rank the configurations with high change rate deviations. While anonymity rises faster after a recharge when high change rate deviations (e.g., crd = 144) are used, anonymity also starts dropping earlier. This shift can be explained through the fact that more nodes near the end of their expected lifetime are able to mix with freshly recharged nodes. While, in this way, newly recharged users receive an early bump in anonymity, more pseudonyms die before the 40th change as they have al-

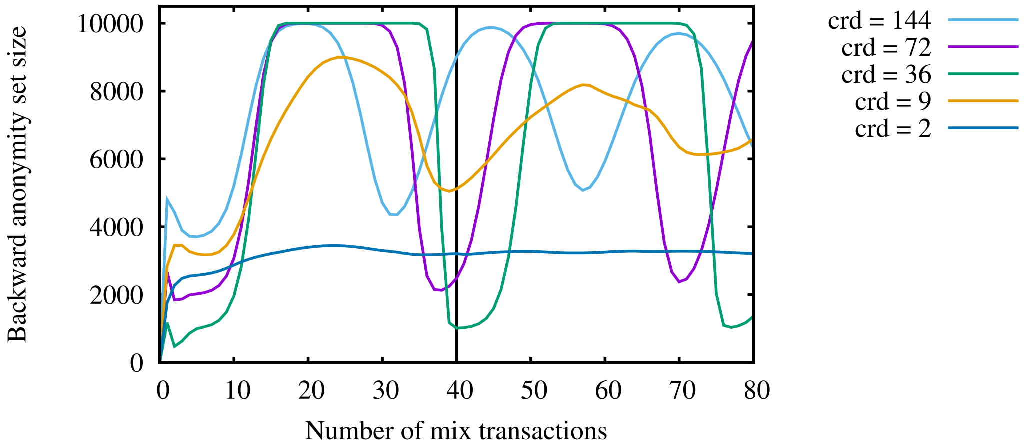


Figure 6.13 Impact of change rate deviation *crd* (in time slots) on backward anonymity.

ready lost much value shortly after being recharged. What can clearly be seen in Figure 6.13, however, is that low levels of change rate deviation, e.g., 9 and below, are detrimental to anonymity development and maximum achievable anonymity. This relationship is understandable, as a low change rate variation leads to a limiting of mix groups to members of small communities with synchronized pseudonym change intervals. It should be noted that change rate deviation is, to a large extent, a parameter of the application scenario of BitNym, and might be influenceable only in a limited fashion by a practical implementation (e.g., by introducing artificial random delays before initiating the discovery of mixing partners).

The impact of the acceptable difference *ad* on the pseudonym values of mixing partners (in satoshi) is shown in Figure 6.14. As expected, a higher acceptable difference, i.e., a higher readiness to lose bitcoins during a mix, leads to a faster anonymity growth. Staring from an acceptable difference of 50 000 satoshi, the maximum reached anonymity level is reduced below the theoretical maximum and decreases further with lower values for the acceptable difference. Again, the reason is that the pool of users out of which mixing partners are taken is enlarged, including more users that are close to the end of their expected lifetime and can, therefore, contribute larger anonymity sets to a mix. Interestingly, however, the configuration with de facto no limit on the acceptable difference (ad = 200 000) leads to an early drop in average anonymity compared to the parametrizations with acceptable differences of 100 000 or less, i.e., starting already around the 30th mix. Two aspects must be considered here. First, measurements for plots that express the relationship of a value to the number of completed mixes are taken only immediately after each successful mix. The observation that anonymity levels don't seem to significantly decline for lower acceptable difference values is, therefore, an illusion due to the fact that the anonymity levels of dead pseudonyms are not shown (the anonymity levels of dead pseudonyms are, due to the previously noted synchronization of pseudonym values, likely falling rapidly with each time slot). The second aspect that must be considered is that while anonymity drops earlier for the parametrization with a de facto unlimited acceptable difference (ad = 200 000), it also rises faster in the beginning. More users with multiple pseudonym mixes behind them are able to mix with freshly recharged pseudonyms, significantly increasing the anonymity of the freshly recharged pseudonyms and, in turn, gaining the ability to perform more pseudonym mixes than average. This effect is also reflected in the previously discussed results on pseudonym lifetime distribution depicted in Figure 6.12.

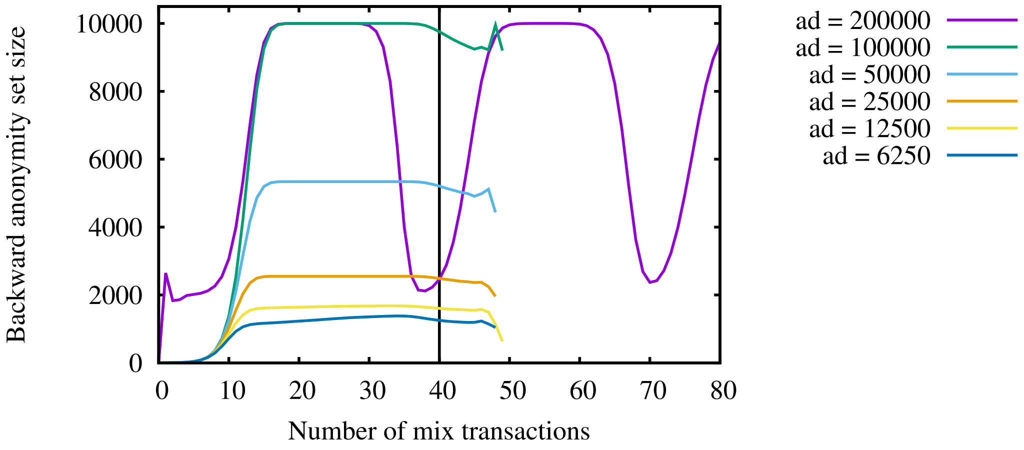


Figure 6.14 Impact of acceptable difference *ad* (in satoshi) on backward anonymity.

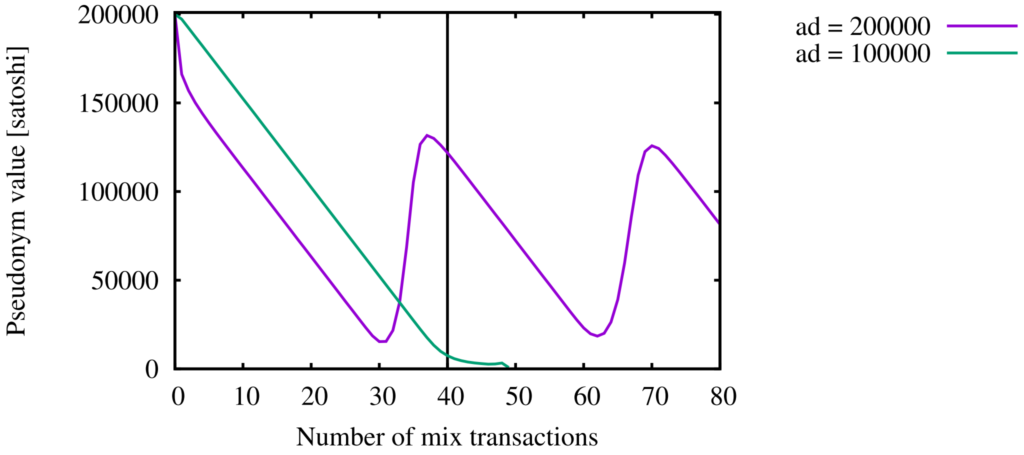


Figure 6.15 Impact of acceptable difference *ad* (in satoshi) on pseudonym value.

On an individual level, the acceptable difference to the pseudonym values of mixing partners encodes the readiness to lose more than the 5000 satoshi of transaction fees for participating in a mix. Independently of the acceptable difference, however, only the transaction fees are in reality lost to the BitNym user population, so that the average loss per mix and mix participant is still only 5000 satoshi. Figure 6.15 indicates the validity of this claim. It depicts the average pseudonym values of users after different numbers of completed mixes. Results for simulated acceptable difference values of below 100 000 satoshi are not shown, as they were indistinguishably close to the values for 100 000 satoshi. For the parametrization with a de facto unlimited acceptable difference (ad = 200 000), pseudonym values drop sharply for the first few mixes. This drop is likely due to mixing with "older" pseudonyms, i.e., such with already low pseudonym values. As was previously discussed, however, such pseudonyms usually have large anonymity sets that they can share with mixing partners. As a pendant to the sharp drop in the beginning, the few pseudonyms surviving past the 30th mix (for ad = 200 000) receive a rise in pseudonym value through mixing with freshly recharged pseudonyms.

Finally, simulations with varying numbers of users were conducted. Figure 6.16 depicts the development of anonymity set sizes (on a logarithmic scale) across mixes for different numbers of users *nu*. Note that the maximum achievable level of anonymity is equal to the number of users in the scenario. Due to the approximately exponential growth of backwards anonymity with every mix, the respective maximum anonymity levels are reached quickly even for larger user populations. Figure 6.17 depicts a more global view on anonymity levels, with anonymity (on a logarithmic scale) plotted against simulated time. Again, results for different numbers of users are shown. It can be seen that the periodic drops in anonymity observed earlier are limited in strength and reduce the average anonymity of the user population by a factor of less than 10.

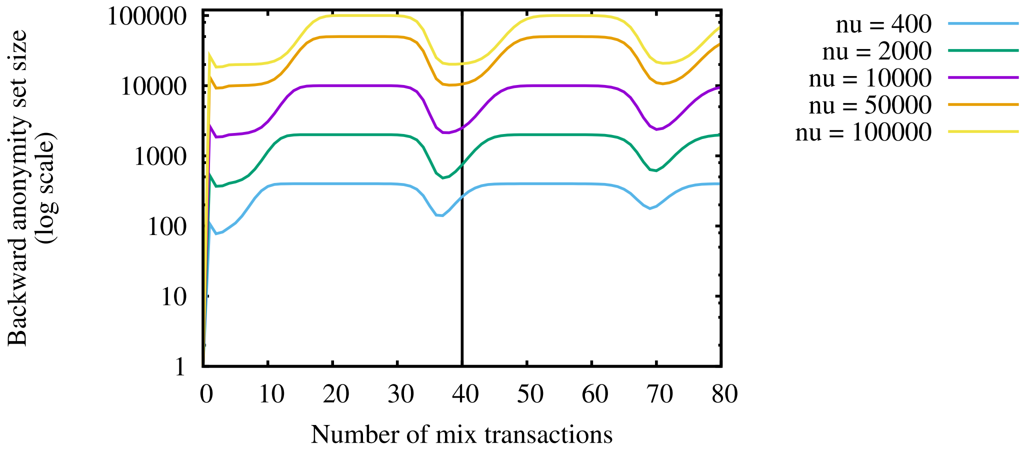


Figure 6.16 Backward anonymity development for different numbers of users *nu*.

It is also interesting to see whether user population size affects the size of pseudonym proofs. Towards answering this question, Figure 6.18 plots average proof sizes against simulated time for different numbers of users. As expected, no difference between the compared parametrizations can be observed, as proof sizes are affected by the number of mix transactions per time interval which is independent of the size of the user population. Additionally, it can be seen that for the simulated scenario, proof sizes after 1000 simulated days are, on average, still below 200 kB.

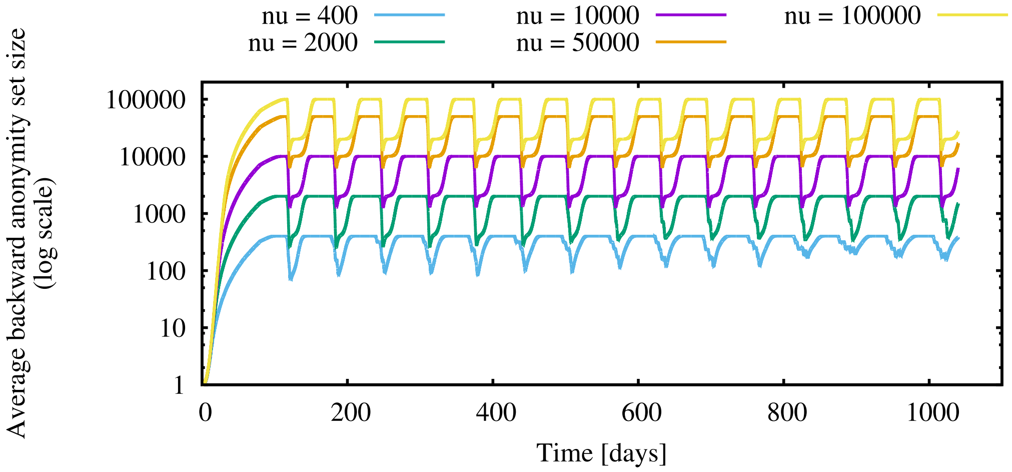


Figure 6.17 Backward anonymity development for different numbers of users *nu*.

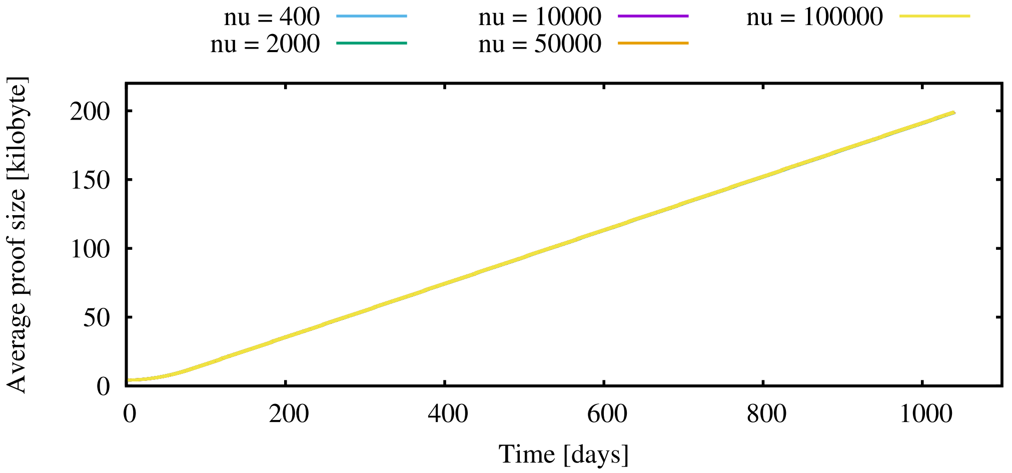


Figure 6.18 Average proof size, for different numbers of users *nu*. The measured values are indistinguishably close.

6.7 Conclusion

This chapter tackles the challenge of enabling unlinkable and TTP-independent pseudonymity while remaining resilient to large-scale sybil attacks. An approach was outlined that is suitable for offering privacy-preserving and abuse-resistant authentication for cooperative services in smart traffic and beyond. Sybil-resistance is achieved by leveraging cryptocurrency block chains as decentralized bulletin boards. Furthermore, this chapter presented BitNym, a specific realization of the proposed approach based on the Bitcoin network. Via a prototype of BitNym, the practical feasibility of the approach was demonstrated. Using simulations of larger user populations, it was demonstrated that large anonymity sets can be reached quickly.

7. Summary and perspectives

Modern vehicles become increasingly intelligent and interconnected. This enables new qualities of cooperation, as cooperation becomes decoupled from the attention of human users. In the domain of *smart traffic*, for example, smart vehicles can autonomously cooperate to optimize traffic flow, request and provide services and exchange sensor data. In addition to promising significant quality of life improvements, such *cooperative services* have the potential for meeting long-standing societal challenges, like reducing environmental impacts and meeting ever increasing mobility demands.

However, the increasing amount of data shared by users also enables new dimensions of *privacy* intrusions. Systems become inherently susceptible to leaks of privacy-relevant data and exploits by determined adversaries if privacy issues are not considered early during system design.

The class of privacy-relevant data most characteristic to smart traffic is that of timestamped *location data*, i.e., data about the location at which some user has been at a given point in time. Location data is required for the provision of a wide variety of smart traffic services, from traffic monitoring and optimization to the delivery of messages based on geographic criteria. At the same time, timestamped location data is highly sensitive from a privacy standpoint. It can reveal habits, interests, social contacts and even health status and political engagement. Consequently, the *location privacy* of users should be preserved.

The *utility* of smart traffic services is often proportional to the amount and quality of shared data. Enabling users to participate anonymously or pseudonymously might be insufficient for preserving privacy in this context. As repeatedly shown in the literature, given a sufficiently high sampling rate and precision, the identities behind anonymously shared locations can be inferred using correlation and easily obtainable context knowledge. As an additional challenge, identifiable data is often needed for detecting and punishing *abuse*, i.e., attacks on the correct functioning of services.

Many existing works and practically deployed systems rely on the existence of trusted entities, e.g., trusted service providers. In addition to being attractive targets for adversaries, trusted entities are often in a position for large-scale privacy breaches. By collecting large amounts of data, for example, correlation attacks become possible through which sensitive information can be inferred from sets of otherwise uninteresting data. In the scope of this thesis, *decentralization* was investigated as a means to eliminate the need for single trusted entities. Decentralized system design avoids centrally controlled data sinks as well as the difficulty of establishing trust anchors in systems of global scale. Additionally, the realization of *data locality* becomes possible, i.e., it can be realized that precise location data is disclosed only to entities that are also provably nearby.

7.1 Results

The presented thesis focuses on three functional building blocks that are both relevant to upcoming smart traffic services and highly challenging in respect of balancing privacy protection and utility: *cooperative planning*, *geographic addressing* and *the decentralized provision of pseudonymous identifiers*. As an important distinction to many related works and practically deployed systems, adversaries are assumed that can compromise centrally controlled system components for obtaining privacy-relevant data. This assumption reflects the threat arising from security breaches, governmental subpoenas and dishonest service operators. It is assumed, however, that the majority of users will not be compromised and, while possibly selfish or curious in isolation, never actively collude with malicious intent.

The specific contributions are the following:

- A *privacy-preserving cooperative route planning* system was introduced. Despite enabling the anonymous publication of plans, the possibilities for abuse (in the form of malicious users that distribute false plans) are limited. Plans are published anonymously as a series of promises concerning segments of the planned route. Abuse resistance is realized by requiring the use of blindly signed *promise coins* for making each promise. Following an initial registration, during which users receive a first batch of promise coins, new promise coins are issued only upon promise fulfillment. In this way, honest users retain their right to participate in the system while malicious users get banned quickly. Promise coins are issued using blind signatures and, therefore, not identifiable or linkable. Consequently, and as confirmed by analysis, promises are anonymous and not easily linked to trips. Through performance measurements of the involved cryptographic operations and latency measurements of the employed anonymous communication channel, it was furthermore shown that the proposed privacy-preserving cooperative route planning system is practically feasible.

- Mechanisms for enabling *privacy-preserving long-distance geocast* were investigated that do not rely on centrally controlled service providers or dedicated infrastructure support. Geocast is a communication abstraction that enables the addressing of communication partners based on geographic criteria, e.g., the addressing of all vehicles in a given region. A comprehensive privacy analysis of the overlay-based geocast service *OverDrive* was performed. In OverDrive, user locations are not shared with a single service provider but with a small dynamic set of other users. Based on the results of the analysis, a location obfuscation concept was proposed that enables the precision of location data to be decreased depending on the distance at which it is shared. Through the integration of location spoofing detection using GSM broadcast traffic, the correctness of this obfuscation can be ensured even in the face of adversaries that fake their own position. The novel extensions to OverDrive were evaluated through simulation. The results demonstrate that, through the proposed enhancements, the large-scale surveillance and the targeted tracking of OverDrive users becomes infeasible even for adversaries controlling hundreds of overlay nodes. From a performance standpoint, OverDrive was shown to represent a viable alternative to more traditional, centralized approaches.

- It was investigated how *sybil-resistant pseudonymization and pseudonym change* can be realized without relying on the existence of trusted third parties. The issuing of pseudonyms is an established approach for protecting the privacy of users while limiting access and preventing sybil attacks. Unlinkable pseudonym changes are required to prevent the identification of pseudonym holders through continuous observation and correlation. A novel approach towards pseudonymization and pseudonym change was proposed that is independent of individual trusted entities without putting an unrealistic burden on users. Robustness and sybil-resistance is achieved by leveraging cryptocurrency block chains as decentralized append-only bulletin boards. In addition to the general approach, *BitNym*, a specific design which leverages the security of the unmodified Bitcoin network, was presented. Via a prototype of BitNym, the practical feasibility of the approach was demonstrated. Using simulations of large user populations, it was demonstrated that anonymity sets encompassing nearly the complete user population are achievable.

A privacy analysis and evaluation of all investigated building blocks was conducted based on a two-step approach. First, the privacy-related threats *detection* and *disclosure* were considered. The type and quality of data an adversary is able to collect was investigated, which includes identifying possible active attacks that can help an adversary to increase his knowledge. Second, possibilities for the *linking* and *identification* of data items available according to the first step were investigated, focusing on the goal of obtaining identifiable location samples.

7.2 Perspectives for future work

A major challenge towards privacy-preserving smart traffic services, that was not tackled in the scope of this thesis, is that of location privacy within cellular communication networks. While first works that enable the anonymous usage of such networks exist [107], a more serious investigation is necessary. Developed solutions must be sufficiently well designed and understood so that they can be included in upcoming cellular networking standards.

The route planning aspects of the cooperative route planning system proposed in Chapter 4 exceed the scope of this thesis and remain unexamined. Open questions remain concerning both the practical realization of such a system and its potential impact on traffic flow and environmental pollution. Conducting investigations based on realistic traffic simulations might be a promising approach for addressing both of these aspects.

Several open questions follow from the novel pseudonymization and pseudonym change approach proposed in Chapter 6. This includes that of decentralized initial access control - how to ensure that each human user is able to register to the system only once. Leveraging the properties of social connections, as also described in this thesis, is a promising direction for further investigation in this respect. Furthermore, the BitNym prototype should be developed further, possibly in combination with the PTP library described in Chapter 2. With moderate effort, an easy to use piece of software can likely be created that can find widespread practical deployment.

The applicability of the developed building blocks to other scenarios is also a promising avenue for exploration. The promise coin approach proposed for cooperative route planning might be applicable to other contexts in which plans need to be published in an (ideally) anonymous fashion while the overall system must remain resilient to abuse. Lessons learned from investigating long-distance geocast overlays can be applied to areas in which nodes must be addressed based on other privacy-relevant properties. As long the reasoning of data locality can be applied, i.e., that entities sharing a similar set of properties are more trustworthy in respect to learning one's own properties, an application of the proposed techniques is conceivable. The BitNym system is, of course, already very generally applicable. An evaluation of the approach in a specific application context is nonetheless an interesting research opportunity.

At the end, most technologies for improving the privacy of smart traffic users are only as useful as the size of their user population. As often stated in a colloquial manner, *anonymity loves company*. The size of a system's user population directly determines achievable anonymity sets, the size of the "crowd" in which users can disappear. Achieving a wider adoption of systems that honor and protect the privacy of their users is a vital and significant challenge.

A. Connectivity and privacy in deployed cellular networks

In 2014, a study was performed investigating the connectivity and privacy properties of the major cellular networks available in Germany. More specifically, the study aimed at answering following questions:

1. Whether the establishment of peer-to-peer connections to other cellular network users is possible.
2. Whether IP address changes can be initiated by users (for ensuring the unlinkability of pseudonym changes).
3. Whether IP addresses used in cellular networks are identifiable by external adversaries.

The setup and results of the study are presented in the following. A summary of the results is given in Section 2.6.1.

This appendix is based on [58] and the appendix of [43].

A.1 Evaluating deployed cellular networks

Several previous works exist that evaluate connectivity properties in commercially deployed cellular networks. In a study conducted in 2007 [76], the authors investigate to what extent and in what form *network address translation* (NAT) is deployed in six exemplary cellular networks from different continents. With NAT, address information is transparently changed by middleboxes along the data path. In this way, for example, multiple hosts in a private network can share one public IP address. NAT middleboxes maintain state in the form of *bindings* in order to be able to correctly translate addresses for incoming packets. Bindings are typically set only for outbound connections. Thus, unsolicited connections to devices behind NAT are typically not possible and *NAT traversal* techniques need to be applied in order for a peer-to-peer connection to a device behind NAT to be established. The findings in [76] indicate that NAT characteristics vary greatly between networks, with half of the tested networks implementing no NAT at all and two of the six carriers implementing NAT configurations that make NAT traversal and the establishment of peer-to-peer connections impossible. Similar results have been found in a 2011 study by Wang et al. [114]. Using a crowdsourcing approach, the authors evaluate, amongst other things, the properties of NAT middleboxes and firewalls in 107 cellular networks around the world. According to their results, from 72 cellular networks employing NAT, NAT traversal is feasible in 53.

While these studies provide valuable insights into the connectivity properties of commercially deployed cellular networks, several open questions remain. First, the mechanisms of assigning IP addresses as well as the possibility for initiating IP address changes have not been evaluated. Second, the cited studies do not capture potential developments from the last few years. Third, they do not provide country-specific information, e.g., concerning cellular networks available in Germany.

Thus, between August and September 2014, an own study was conducted using prepaid SIM cards from the four cellular networks available in Germany. In the following, the individual networks are referred to as *N1* to *N4*. The main goal of the study was the investigation of the networks' properties concerning the establishment of peer-to-peer connections and the requesting of new IP addresses for ensuring the unlinkability of pseudonyms following pseudonym changes.

The employed methodology is based on a setup with two cellular network-enabled clients and a publicly reachable test server. An evaluation framework was developed for automatically running large numbers of tests within this setup. The server acts as an introduction and coordination point and, in some tests, is used as a STUN server, for assisting in NAT traversal and for determining the external IP addresses of clients. In the following, the tests and test results concerning the establishment of peer-to-peer connections and the implemented policies regarding the assignment and changing of client IP addresses are discussed.

A.2 Establishing peer-to-peer connections

Due to an increasing use of IP middleboxes (e.g., firewalls and NAT) in provider networks, the availability of Internet connectivity does not automatically imply that the creation of peer-to-peer connections is possible as well. Cellular network operators, for example, rely heavily on NAT to both improve the utilization of their public IP address pools and shield connected users from unsolicited communication attempts.

Using the test setup outlined in Section A.1, the following properties were evaluated for the networks N1 to N4:

	N1	N2	N3	N4
N1	direct connection ✓	connection reversal ✓	connection reversal ✓	connection reversal ✓
N2		port guessing ×	port guessing ×	port guessing ∼
N3			port guessing ×	port guessing ∼
N4				UDP hole punching ✓

Table A.1 Feasibility peer-to-peer connection establishment.

1. Whether NATs were deployed at all.
2. The employed type of *endpoint filtering*, i.e., based on which criteria incoming packets are associated to an open binding and, thus, forwarded through the NAT.
3. The employed *NAT mapping*, i.e., based on what principle external (public) ports are chosen for new bindings.

Based on these properties, suitable approaches for establishing peer-to-peer connections were determined and tested in practice, e.g. state of the art NAT traversal techniques. The best suited techniques for each combination of networks, as well as estimations of their practical feasibility, are depicted in Table A.1.

According to the gathered results, N1 offered network access without NAT. Furthermore, it was possible to consistently establish direct connections between clients in N1. The establishment of connections between clients in N1 and clients behind NAT was possible as well using the *connection reversal* [105] method.

The remaining three carriers employed NAT using *address- and port-dependent* endpoint filtering. Clients behind this type of NAT can establish peer-to-peer connections if the external port allocated by the NAT can be predicted for both of them. The feasibility of such a prediction is dependent on the deployed type of NAT mapping.

According to the gathered findings, N4 employs an *endpoint-independent* type of NAT mapping (also known as *cone*). Here, every outbound packet with the same source address and source port is translated to the same external source address and source port, independently of the destination address or port (and vice versa for incoming packets). For two clients located behind this type of NAT, *UDP hole punching* [105] can be used, as was confirmed by tests in practice.

The remaining two networks N2 and N3 employ NAT with random port mapping. Here, every new binding receives a completely random external source port. This mapping type is the most unfavorable for realizing NAT traversal and establishing peer-to-peer connections, as it allows no efficient predictions to be made. Instead, *port guessing* needs to be used, where peers make attempts using different ports until a connection is established. This approach is highly time-intensive and limited by the data rate available to clients as well as network parameters like the number of NAT bindings a client is allowed to establish. For the parameters measured in N2, N3 and N4, establishing a peer-to-peer connection using port guessing can take from 27 minutes on average (N4 to N2 or N3) to 72 days on average (N2 to N3). It is, thus, concluded that this approach is infeasible for most peer-to-peer application scenarios. Consequently, clients of N2 and N3 cannot effectively participate in most peer-to-peer systems if direct peer-to-peer connections over IP need to be used (alternative approaches for establishing peer-to-peer connections are discussed in Section 2.6.2).

In addition to employing NAT, the networks N2, N3 and N4 did not allow direct connections between network-internal addresses. Thus, if two clients reside in the same network, they still need to perform NAT traversal in order to establish a peer-to-peer connection.

A.3 IP address changes and linkability

In addition to determining the feasibility of establishing peer-to-peer connections in cellular networks, we investigated the implemented policies regarding IP address assignment and the possibility for users to initiate IP address changes. Using the test setup described in Sec. A.1, it was discovered that new IP addresses from the operators' individual subnets were assigned upon every reconnection to the cellular network. This effect was witnessed in all considered networks. Thus, users can easily initiate IP address changes, ensuring unlinkability on pseudonym changes, by disconnecting and reconnecting to the network. However, as an important detail, both the new and the old address belong to subnets owned by the network operator. Thus, an adversary might be able to link addresses to cellular networks, which can reduce the unlinkability offered by pseudonym changes.

Further measurements were conducted in which the location of the measuring station was varied between two locations located 90km away from each other. The change in locations had no detectable influence on the IP addresses assigned to clients.

B. Interactive visualization of OverDrive

An interactive demonstrator of the OverDrive protocol was developed that visualizes OverDrive's neighborhood structures and routing approach. Through a novel extension to the overlay simulation framework *OverSim* [3], node movement in geographic space can be visualized as well as the underlying road network. Screenshots of this visualization can be seen in Figure B.1.

An earlier version of the work described here was previously presented in [39].

B.1 Functionality

The visualization extension supports two *views* - the default *visualization view*, presenting the movement of all simulated nodes on the underlying street map, as well as an *application view* tracking the movement of one particular vehicle.

In addition to giving a global overview of all nodes and their movement (cf. Figure B.1a), the *visualization view* also enables the inspection of OverDrive protocol internals. Individual nodes can be selected, displaying their overlay neighbors and the concentric rings used in OverDrive's neighbor selection logic (cf. Figure B.1b). The neighbors of the selected node (which is highlighted in purple) are colored in light blue. Additional green markers are used to represent the knowledge of the selected node about its neighbors' locations. The green markers represent obfuscated locations. By clicking on the map while a node is selected, the sending of a GUM can be initiated. The GUM originates from the selected node and is forwarded to the destination point according to OverDrive's forwarding logic. The path that the GUM takes is shown in the visualization (cf. Figure B.1b). Flooding messages are initialized and visualized in an analog manner, by switching the message sending mode in the visualization interface.

Once the application view is selected (multiple views can be open simultaneously on different devices or browsers), one of the simulated vehicles is randomly chosen and the view shifts to present that vehicle's perspective, tracking its movement (cf. Figure B.1c). The view is similar to that of popular navigation applications, with a map centered on the current position of the tracked vehicle. The initiation of GUMs to arbitrary points on the map is also enabled. These messages get routed according to OverDrive's routing logic. Once a GUM reaches its final recipient, the location of this recipient is shown on the application view. Clearly, this functionality is only a placeholder for more sophisticated real-life applications like the ones discussed in Section 5.1.1.

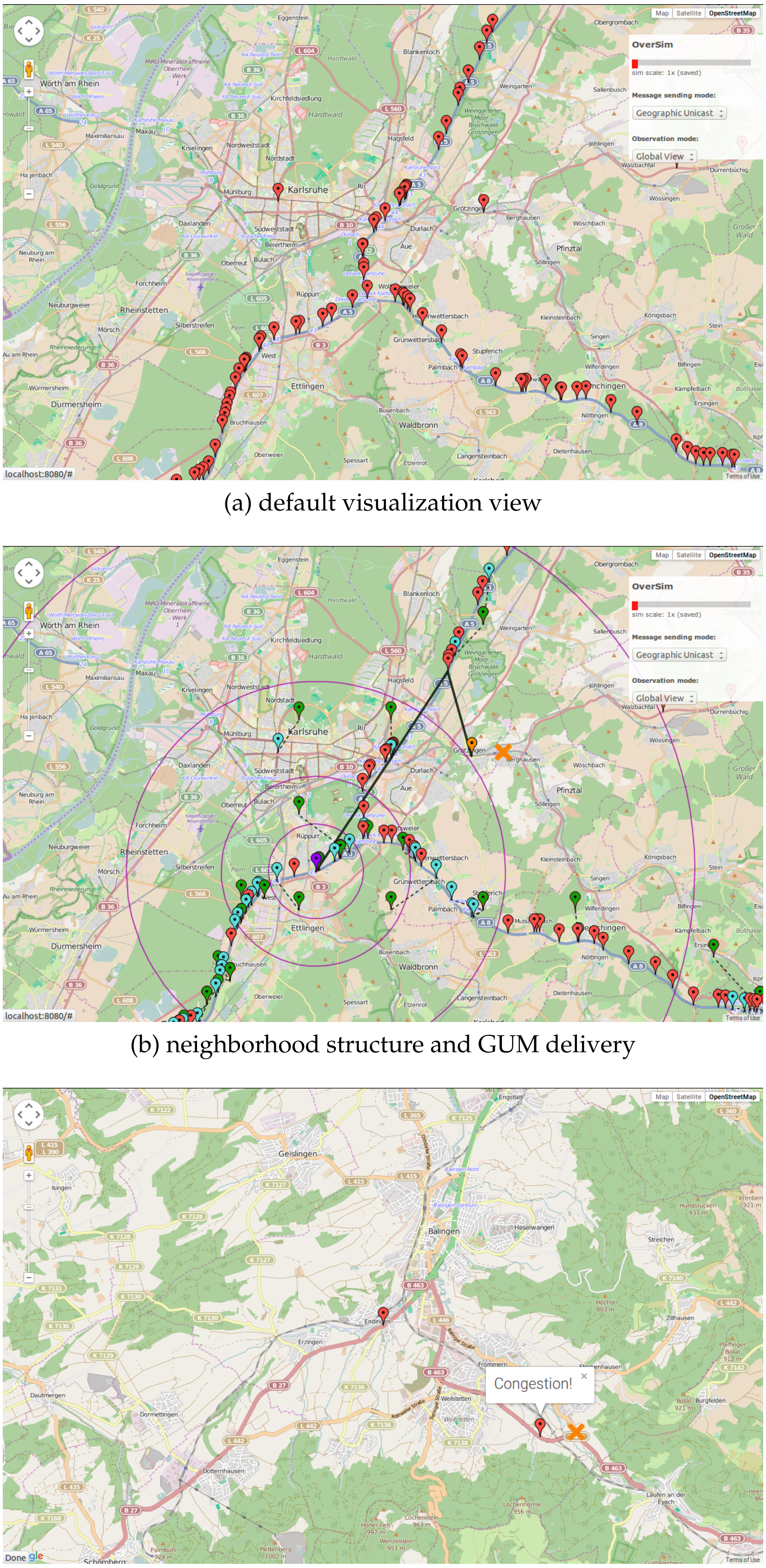
B.2 Extensions to OverSim

For realizing the interactive visualization of OverDrive, several extensions were made to OverSim. A new *dynamic scheduler* was implemented that enables the explicit setting and on-the-fly changing of the simulation speed. In the visualization of OverDrive, this functionality is used for aligning the simulation time with real time and speeding up the simulation when this is more convenient for inspecting the inner workings of OverDrive. In addition to the new scheduler, a lightweight *web server*[1] was integrated into OverSim that enables the visualization of OverDrive nodes and data structures as well as the tunneling back, via a web interface, of commands into the running simulation in OverSim. The actual web-based views are based on the *Google Maps JavaScript API*[2] and map material from the OpenStreetMap project. From the views, the running simulation can be influenced directly, e.g., by initiating the sending of GUMs or by changing the simulation speed.

These additions to OverSim can easily be reused for visualizing other protocols and scenarios in which node movement and the geographic location of nodes is of importance. In addition to overlays for smart traffic scenarios, this includes, for example, works in the area of user-centric networking [2].

[1] http://code.google.com/p/mongoose/

[2] https://developers.google.com/maps/documentation/javascript/



OverSim

sim scale: 1x (saved)

Message sending mode: Geographic Unicast

Observation mode: Global View

Congestion!

(a) default visualization view

(b) neighborhood structure and GUM delivery

(c) application view

Figure B.1 Screenshots of interactive visualization.

C. Simulation parameters

Parameter	**Value**
General parameters	
# regular nodes	10 000
simulation duration	4200 s
warm-up period	600 s
initial node creation interval	0.5 s
accelerated node creation interval	0.05 s
# nodes after which accelerated creation starts	200
Parametrization of regular OverDrive	
n_{des}	4
n_{max}	20
r_{b}	2 km
# rings	8
ring satisfaction check interval	60 s
Parametrization of obfuscation-enabled OverDrive	
n_{des}	8
n_{max}	32
f_{d}	1
r_{b}	2 km
# rings	8
ring satisfaction check interval	30 s
Parametrization of location spoofing detection	
verification attempt timeout (cell)	15 s
verification attempt timeout (LA)	30 s
maximum verification delay (cell)	150 s
maximum verification delay (LA)	300 s
verification interval (cell)	600 s
verification interval (LA)	900 s
cell radius	2 km
LA radius	5 km
blacklist retention period	300 s
Parametrization of attacker nodes for Figures 5.6 and 5.7	
# attacker nodes	$\{1, 5, 10, 20, 50, 100\}$
n_{des}	*unlimited*
n_{max}	*unlimited*
fake own location	*yes*
Parametrization of attacker nodes for Figure 5.8	
# attacker nodes	$\{1, 2, 5, 10\}$
n_{des}	*unlimited*
n_{max}	*unlimited*
fake own location	*yes*
Parametrization of victim nodes for Figure 5.8	
# victim nodes (target)	100
victim node creation interval	120 s

Table C.1 Simulation parameters used in Chapter 5.

Parameter	**Value**
Default parameters	
# users	10 000
simulation duration	13 lifetimes / 1040 days
pseudonym start value	200 000 satoshi
mining fee per mix transaction and participant	5000 satoshi
average change rate	288 time slots
change rate deviation	72
mix group size	2
allowed difference to mixing partners	200 000 satoshi
base proof size	4075 B
proof size increase per mix participant	190 B
Parameter variations for Figures 6.9, 6.10 and 6.11	
mix group size	{ 2, 3, 5, $\geq$2 }
Parameter variations for Figure 6.13	
change rate deviation	{ 144, 72, 36, 9, 2 }
Parameter variations for Figures 6.12, 6.14 and 6.15	
allowed difference to mixing partners	{ 200 000, 100 000, 50 000, 25 000, 12 500, 6250} satoshi
Parameter variations for Figures 6.16, 6.17 and 6.18	
# users	{ 400, 2000, 10 000, 50 000, 100 000 }

Table C.2 Simulation parameters used in Chapter 6.

Bibliography

[1] Elli Androulaki, Seung Geol Choi, Steven M Bellovin, and Tal Malkin. Reputation systems for anonymous networks. In *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (PETS 2008)*, pages 202–218. Springer, July 2008.

[2] Ingmar Baumgart and Fabian Hartmann. Towards secure user-centric networking: Service-oriented and decentralized social networks. In *Proceedings of the First International Workshop on Socio-Aware Networked Computing Systems at 5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SaSo)*. IEEE, October 2011.

[3] Ingmar Baumgart, Bernhard Heep, and Stephan Krause. OverSim: A scalable and flexible overlay framework for simulation and real network applications. In *Proceedings of the 9th IEEE International Conference on Peer-to-Peer Computing (P2P '09)*, pages 87–88. IEEE, September 2009.

[4] Ingmar Baumgart and Sebastian Mies. S/Kademlia: A Practicable Approach Towards Secure Key-Based Routing. In *Proceedings of the 13th International Conference on Parallel and Distributed Systems (ICPADS'07)*, volume 2, pages 1–8. IEEE, December 2007.

[5] Jörg Becker, Dominic Breuker, Tobias Heide, Justus Holler, Hans Peter Rauer, and Rainer Böhme. Can we afford integrity by proof-of-work? Scenarios inspired by the Bitcoin currency. In *The Economics of Information Security and Privacy*, pages 135–156. Springer, 2013.

[6] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2014.

[7] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *Proceedings of the 2nd IEEE Annual Conference on Perva-*

sive Computing and Communications Workshops (PERCOMW04), pages 127–131. IEEE, March 2004.

[8] George Bissias, A Pinar Ozisik, Brian N Levine, and Marc Liberatore. Sybil-Resistant Mixing for Bitcoin. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES)*, pages 149–158. ACM, November 2014.

[9] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. Mixcoin: Anonymity for Bitcoin with accountable mixes. In *Financial Cryptography and Data Security*, pages 486–504. Springer, March 2014.

[10] Nikita Borisov. Computational puzzles as sybil defenses. In *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P)*, pages 171–176. IEEE, September 2006.

[11] Giacomo Brambilla, Marco Picone, Michele Amoretti, and Francesco Zanichelli. An Adaptive Peer-to-Peer Overlay Scheme for Location-Based Services. In *Proceedings of the 13th IEEE International Symposium on Network Computing and Applications (NCA)*, pages 181–188. IEEE, August 2014.

[12] Walter Bronzi, Raphael Frank, German Castignani, and Thomas Engel. Bluetooth Low Energy performance and robustness analysis for Inter-Vehicular Communications. *Ad Hoc Networks*, 37(1):76–86, February 2016.

[13] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS)*, pages 201–210. ACM, October 2006.

[14] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 21–30. ACM, November 2002.

[15] Bogdan Carbunar and Rahul Potharaju. You unlocked the Mt. Everest badge on foursquare! Countering location fraud in Geosocial Networks. In *Proceedings of the 9th International Conference on Mobile Adhoc and Sensor Systems (MASS 2012)*, pages 182–190. IEEE, October 2012.

[16] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of Crypto 82*, pages 199–203. Springer, 1983.

[17] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[18] Chen Chen, Daniele E Asoni, David Barrera, George Danezis, and Adrain Perrig. HORNET: high-speed onion routing at the network layer. In *Proceedings*

of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS), pages 1441–1454. ACM, October 2015.

[19] Tat Wing Chim, Siu-Ming Yiu, Lucas CK Hui, and Victor OK Li. Privacy-preserving advance power reservation. *IEEE Communications Magazine*, 50(8):18–23, August 2012.

[20] Chi-Yin Chow, Mohamed F Mokbel, and Xuan 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. ACM, November 2006.

[21] Delphine Christin, Christian Roßkopf, and Matthias Hollick. uSafe: A privacy-aware and participative mobile application for citizen safety in urban environments. *Pervasive and Mobile Computing*, 9(5):695–707, October 2013.

[22] Delphine Christin, Christian Roßkopf, Matthias Hollick, Leonardo A Martucci, and Salil S Kanhere. IncogniSense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive and Mobile Computing*, 9(3):353–371, June 2013.

[23] Rutger Claes, Tom Holvoet, and Danny Weyns. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):364–373, March 2011.

[24] Jeremy Clark and Aleksander Essex. CommitCoin: Carbon dating commitments with bitcoin. In *Financial Cryptography and Data Security*, pages 390–398. Springer, March 2012.

[25] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, December 2009.

[26] George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Le Metayer, Rodica Tirtea, and Stefan Schiffner. Privacy and Data Protection by Design - from policy to engineering. Report, European Union Agency for Network and Information Security (ENISA), January 2015.

[27] George Danezis and Prateek Mittal. SybilInfer: Detecting Sybil Nodes using Social Networks. In *Proceedings of the 16th Annual Network and Distributed System Security Symposium (NDSS)*. USENIX, January 2009.

[28] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1):3–32, March 2011.

[29] Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008.

[30] Jochen Dinger and Hannes Hartenstein. Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration. In *Proceedings of the First International Conference on Availability, Reliability and Security (ARES 2006)*, pages 8–pp. IEEE, April 2006.

[31] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[32] Danny Dolev and Andrew C Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, March 1983.

[33] John R Douceur. The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer, 2002.

[34] Kurt Dresner and Peter Stone. Multiagent traffic management: An improved intersection control mechanism. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS'05)*, pages 471–477. ACM, July 2005.

[35] Falko Dressler, Hannes Hartenstein, Onur Altintas, and Ozan Tonguz. Inter-vehicle communication: Quo vadis. *IEEE Communications Magazine*, 52(6):170–177, June 2014.

[36] David Eckhoff, Reinhard German, Christoph Sommer, Falko Dressler, and Tobias Gansen. SlotSwap: strong and affordable location privacy in intelligent transportation systems. *IEEE Communications Magazine*, 49(11):126–133, November 2011.

[37] David Eckhoff and Christoph Sommer. Driving for Big Data? Privacy Concerns in Vehicular Networking. *IEEE Security & Privacy*, 12(1):77–79, January 2014.

[38] Sören Finster. Smart Meter Speed Dating, short-term relationships for improved privacy in Smart Metering. In *Proceedings of the 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 426–431. IEEE, October 2013.

[39] Martin Florian, Simeon Andreev, and Ingmar Baumgart. Demo: OverDrive - an Overlay-based Geocast Service for Smart Traffic Applications. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking (MobiCom'13)*, pages 147–150. ACM, September 2013.

[40] Martin Florian and Ingmar Baumgart. Privacy in overlay-based smart traffic systems. In *IEEE 38th Conference on Local Computer Networks (LCN) Workshops*, pages 912–917. IEEE, October 2013.

[41] Martin Florian, Sören Finster, and Ingmar Baumgart. Privacy-Preserving Cooperative Route Planning. *IEEE Internet of Things Journal*, 1(6):590–599, October 2014.

[42] Martin Florian, Felix Pieper, and Ingmar Baumgart. Establishing Location-Privacy in Decentralized Long-Distance Geocast Services. In *Proceedings of the 2014 IEEE Vehicular Networking Conference (VNC)*. IEEE, December 2014.

[43] Martin Florian, Felix Pieper, and Ingmar Baumgart. Establishing location privacy in decentralized long-distance geocast services. *Ad Hoc Networks*, 37, Part 1:110–121, February 2016.

[44] Martin Florian, Johannes Walter, and Ingmar Baumgart. Sybil-Resistant Pseudonymization and Pseudonym Change without Trusted Third Parties. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 65–74. ACM, October 2015.

[45] Miguel Freitas. twister-a P2P microblogging platform. *arXiv preprint arXiv:1312.7152*, December 2013.

[46] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, November 2011.

[47] Christina Garman, Ian Miers, and Matthew Green. Decentralized Anonymous Credentials. In *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS)*. USENIX, February 2014.

[48] Mario Gerla, Jui-Ting Weng, and Giovanni Pau. Pics-on-wheels: Photo surveillance in the vehicular cloud. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 1123–1127. IEEE, January 2013.

[49] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in VANETs. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37. ACM, September 2004.

[50] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In *Pervasive Computing*, pages 390–397. Springer, May 2009.

[51] Christian Gross, Dominik Stingl, Björn Richerzhagen, Andreas Hemel, Ralf Steinmetz, and David Hausheer. Geodemlia: A Robust Peer-to-Peer Overlay Supporting Location-Based Search. In *Proceedings of the 12th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P'12)*, pages 25–36. IEEE, September 2012.

[52] Sebastian Gräfling, Petri Mähönen, and Janne Riihijärvi. Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications. In *Proceedings of the 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 344–348, June 2010.

[53] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[54] Mike Hearn. Creating Bitcoin passports using sacrifices. Bitcoin Forum, February 2013. https://bitcointalk.org/index.php?topic=140711.0.

[55] Bernhard Heep. R/Kademlia: Recursive and Topology-aware Overlay Routing. In *Proceedings of 2010 Australasian Telecommunication Networks and Applications Conference (ATNAC 2010)*, pages 102–107. IEEE, November 2010.

[56] Bernhard Heep, Martin Florian, Johann Volz, and Ingmar Baumgart. OverDrive: An Overlay-based Geocast Service for Smart Traffic Applications. In *Proceedings of the 10th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, pages 1–8. IEEE, March 2013.

[57] Ryan Henry and Ian Goldberg. Formalizing anonymous blacklisting systems. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy (SP)*, pages 81–95. IEEE, May 2011.

[58] Philipp Hertweck. Konnektivitätseigenschaften in deutschen Mobilfunknetzen. Bachelor thesis, Institute of Telematics, Karlsruhe Institute für Technology (KIT), Supervisors: Martin Florian, Ingmar Baumgart and Martina Zitterbart, September 2014.

[59] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work, Juan-Carlos Herrera, Alexandre M Bayen, Murali Annavaram, and Quinn Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys'08)*, pages 15–28. ACM, June 2008.

[60] Jason E Holt and Kent E Seamons. Nym: Practical pseudonymity for anonymous networks. *Internet Security Research Lab Technical Report*, 4:1–12, June 2006.

[61] Jean-Pierre Hubaux, Srdjan Capkun, and Jun Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy Magazine*, 2(LCA-ARTICLE-2004-007):49–55, June 2004.

[62] John Peters Humphrey. Universal Declaration of Human Rights, December 1948.

[63] Cullen Jennings, Bruce B Lowekamp, Eric Rescorla, Salman A Baset, and Henning Schulzrinne. REsource LOcation And Discovery (RELOAD) Base Protocol. RFC 6940 (Proposed Standard), January 2014.

[64] Tobias Jeske. Floating Car Data from Smartphones: What Google and Waze Know About You and How Hackers Can Control Traffic. *Proceedings of the BlackHat Europe*, March 2013.

[65] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web (WWW'03)*, pages 640–651. ACM, May 2003.

[66] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE Communications Surveys & Tutorials*, 13(4):584–616, July 2011.

[67] Sunny King and Scott Nadal. PPCoin: Peer-to-peer crypto-currency with proof-of-stake, August 2012. https://archive.org/details/PPCoinPaper.

[68] John Krumm. Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing (PERVASIVE 2007)*, pages 127–143. Springer, May 2007.

[69] Ben Laurie and Richard Clayton. "Proof-of-Work" proves not to work. In *Workshop on Economics and Information Security*, July 2004.

[70] Chris Lesniewski-Lass and M Frans Kaashoek. Whanau: A sybil-proof distributed hash table. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation (NSDI)*. USENIX, April 2010.

[71] Mingyan Li, Krishna Sampigethaya, Leping Huang, and Radha Poovendran. Swing & swap: user-centric approaches towards maximizing location privacy. In *Proceedings of the 5th ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 19–28. ACM, October 2006.

[72] Qinghua Li and Guohong Cao. Providing privacy-aware incentives for mobile sensing. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, volume 18, pages 76–84. IEEE, July 2013.

[73] Zi Lin, Denis Foo Kune, and Nicholas Hopper. Efficient private proximity testing with gsm location sketches. In *Financial Cryptography and Data Security*, pages 73–88. Springer, March 2012.

[74] Rongxing Lu, Xiaodong Li, Tom H Luan, Xiaohui Liang, and Xuemin Shen. Pseudonym changing at social spots: An effective strategy for location privacy in VANETs. *IEEE Transactions on Vehicular Technology*, 61(1):86–96, January 2012.

[75] Christian Maihofer. A survey of geocast routing protocols. *IEEE Communications Surveys & Tutorials*, 6(2):32–42, 2004.

[76] Lauri Makinen and Jukka K Nurminen. Measurements on the feasibility of TCP NAT traversal in cellular networks. In *Next Generation Internet Networks (NGI)*, pages 261–267. IEEE, April 2008.

[77] Greg Maxwell. CoinJoin: Bitcoin privacy for the real world. Bitcoin Forum, August 2013. https://bitcointalk.org/index.php?topic=279249.0.

[78] Petar Maymounkov and David Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[79] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP)*, pages 397–411. IEEE, May 2013.

[80] Andrew Miller. Storing UTXOs in a Balanced Merkle Tree (zero-trust nodes with O(1)-storage). Bitcoin Forum, August 2012. https://bitcointalk.org/index.php?topic=101734.0.

[81] Prateek Mittal, Matthew Caesar, and Nikita Borisov. X-Vine: Secure and Pseudonymous Routing in DHTs Using Social Networks. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS)*. USENIX, February 2012.

[82] Steven J Murdoch and George Danezis. Low-cost traffic analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (SP)*, pages 183–195. IEEE, May 2005.

[83] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. TrafficView: traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(3):6–19, January 2004.

[84] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. http://nakamotoinstitute.org/bitcoin/, October 2008.

[85] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP)*, pages 111–125. IEEE, June 2008.

[86] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (SP)*, pages 173–187. IEEE, April 2009.

[87] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, and Dan Boneh. Location Privacy via Private Proximity Testing. In *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*. USENIX, February 2011.

[88] Julio C Navas and Tomasz Imielinski. GeoCast—geographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–76. ACM, September 1997.

[89] Kay Noyen, Dirk Volland, Dominic Wörner, and Elgar Fleisch. When Money Learns to Fly: Towards Sensing as a Service Applications Using Bitcoin. *arXiv preprint arXiv:1409.5841*, September 2014.

[90] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In *Proceedings of the third conference on Theory of Cryptography*, pages 80–99. Springer, March 2006.

[91] Balaji Palanisamy and Ling Liu. MobiMix: Protecting location privacy with mix-zones over road networks. In *Proceedings of the 27th IEEE International Conference on Data Engineering (ICDE)*, pages 494–505. IEEE, April 2011.

[92] Panagiotis Papadimitratos, Levente Buttyan, Tamás Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and J-P Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, November 2008.

[93] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym schemes in vehicular networks: a survey. *IEEE Communications Surveys & Tutorials*, 17(1):228–255, March 2015.

[94] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, August 2010.

[95] Marcio Picone, Michele Amoretti, and Francesco Zanichelli. GeoKad: A P2P Distributed Localization Protocol. In *Proceedings of the 8th IEEE International Pervasive Computing and Communications Conference (PERCOM 2010) Workshops*, pages 800–803. IEEE, March 2010.

[96] Raluca A Popa, Hari Balakrishnan, and Andrew J Blumberg. VPriv: Protecting Privacy in Location-Based Vehicular Services. In *Proceedings of the 18th USENIX Security Symposium*, pages 335–350. USENIX, August 2009.

[97] Veena Pureswaran and Paul Brody. Device democracy: Saving the future of the Internet of Things. IBM Global Business Services Executive Report, IBM, July 2015.

[98] Alan Reiner. Creating Bitcoin passports using sacrifices. Bitcoin Forum, June 2012. https://bitcointalk.org/index.php?topic=88208.0.

[99] Meni Rosenfeld. Overview of colored coins. https://bitcoil.co.il/BitcoinX.pdf, December 2012.

[100] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In *Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS'14)*, volume 8713 of *Lecture Notes in Computer Science*, pages 345–364. Springer, September 2014.

[101] Jedrzej Rybicki, Björn Scheuermann, Markus Koegel, and Martin Mauve. PeerTIS: a peer-to-peer traffic information system. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking (VANET'09)*, pages 23–32. ACM, September 2009.

[102] Jochen Schiller. *Mobile Communications*. Addisson-Wesley, April 2000.

[103] Donald C Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

[104] Daniel J Solove. *Understanding privacy*. Harvard University Press, Cambridge, MA, USA, May 2008.

[105] Pyda Srisuresh, Bryan Ford, and Dan Kegel. State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs). RFC 5128 (Proposed Standard), March 2008.

[106] Stuart G Stubblebine, Paul F Syverson, and David M Goldschlag. Unlinkable serial transactions: protocols and applications. *ACM Transactions on Information and System Security (TISSEC)*, 2(4):354–389, October 1999.

[107] Keen Sung, Brian Neil Levine, and Marc Liberatore. Location Privacy without Carrier Cooperation. In *Proceedings of the IEEE Workshop on Mobile System Technologies (MoST)*. IEEE, May 2014.

[108] Sami Tabbane. *Handbook of Mobile Radio Networks*. Artech House Mobile Communications Library. Artech House, January 2000.

[109] Anthony Tockar. Riding with the Stars: Passenger Privacy in the NYC Taxicab Dataset. http://research.neustar.biz/2014/09/15/riding-with-the-stars-passenger-privacy-in-the-nyc-taxicab-dataset/, September 2014.

[110] Florian Tschorsch and Björn Scheuermann. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IACR Cryptology ePrint Archive*, 2015(464), May 2015.

[111] Guido Urdaneta, Guillaume Pierre, and Maarten Van Steen. A survey of DHT security techniques. *ACM Computing Surveys (CSUR)*, 43(2):8, January 2011.

[112] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. In *Proceedings of the ACM SIGCOMM 2010 conference*, pages 363–374. ACM, August 2011.

[113] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. CAPTCHA: Using hard AI problems for security. In *Advances in Cryptology - EUROCRYPT 2003*, pages 294–311. Springer, May 2003.

[114] Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Zhuoqing Mao, and Ming Zhang. An untold story of middleboxes in cellular networks. In *Proceedings of the ACM SIGCOMM 2011 conference*, pages 374–385. ACM, August 2011.

[115] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, January 2014.

[116] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, April 2014.

[117] Björn Wiedersheim, Zhendong Ma, Frank Kargl, and Panos Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *Proceedings of the 7th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 176–183. IEEE, February 2010.

[118] Lars Wischoff, André Ebner, Hermann Rohling, Matthias Lott, and Rudiger Halfmann. SOTIS - a self-organizing traffic information system. In *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC)*, volume 4, pages 2442–2446. IEEE, April 2003.

[119] Kim Wuyts and Wouter Joosen. LINDDUN privacy threat modeling: a tutorial. CW Reports CW685, Department of Computer Science, KU Leuven, July 2015.

[120] Bin Xiao, Bo Yu, and Chuanshan Gao. Detection and localization of sybil nodes in VANETs. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pages 1–8. ACM, September 2006.

[121] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP)*, pages 3–17. IEEE, June 2008.

[122] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2006)*, pages 267–278. ACM, August 2006.

[123] Hui Zang and Jean Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom'11)*, pages 145–156. ACM, September 2011.

[124] Jan Henrik Ziegeldorf, Fred Grossmann, Martin Henze, Nicolas Inden, and Klaus Wehrle. CoinParty: Secure Multi-Party Mixing of Bitcoins. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pages 75–86. ACM, March 2015.