Data-Centric Trust in Ephemeral Networks

THÈSE N^{O} 4423 (2009)

PRÉSENTÉE LE 19 JUIN 2009

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 1

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Maxime RAYA

acceptée sur proposition du jury:

Prof. A. Martinoli, président du jury
Prof. J.-P. Hubaux , directeur de thèse
Prof. B. Faltings, rapporteur
Prof. V. D. Gligor, rapporteur
Dr A. Held, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2009

Abstract

New network types require new security concepts. Surprisingly, trust - the ultimate goal of security - has not evolved as much as other concepts. In particular, the traditional notion of building trust in entities seems inadequate in an ephemeral environment where contacts among nodes are often short-lived and non-recurrent. It is actually the trustworthiness of the data that entities generate that matters most in these ephemeral networks. And what makes things more interesting is the continuous "humanization" of devices, by making them reflect more closely their owners' preferences, including the human sense of costs. Hence, in this thesis we study the notion of data-centric trust in an ephemeral network of rational nodes.

The definition of a new notion requires specifying the corresponding basis, measures, and raison d'être. In the following chapters, we address these issues. We begin by defining the system and security models of an example ephemeral network, namely a vehicular network. Next, we delve into the subject of revocation in vehicular networks, before creating and analyzing a game-theoretic model of revocation, where the notion of cost-aware devices makes its first appearance in this thesis. This model not only makes possible the comparison of different revocation mechanisms in the literature, but also leads to the design of an optimal solution, the RevoGame protocol. With the security architecture in place, we formally define data-centric trust and compare several mechanisms for evaluating it. Notably, we apply the Dempster-Shafer Theory to cases of high uncertainty. Last but not least, we show that data-centric trust can reduce the privacy loss resulting from the need to establish trust. We first create a model of the trust-privacy tradeoff and then analyze it with game theory, in an environment of privacy-preserving entities. Our analysis shows that proper incentives can achieve this elusive tradeoff.

Keywords: trust, data-centric, ephemeral network, VANET, rationality, game theory, revocation, privacy.

Résumé

Les nouveaux types de réseaux requièrent de nouveaux concepts de sécurité informatique. Étonnamment, la confiance - l'objectif ultime de la sécurité - n'a pas évolué autant que les autres concepts. Surtout, la notion traditionnelle de l'établissement de la confiance dans les entités semble insuffisante dans un milieu éphémère où les contacts entre les nœuds sont souvent de courte durée et non récurrents. Ce qui importe le plus dans ces réseaux éphémères est la fiabilité des données que les entités génèrent. Et ce qui rend le problème plus intéressant est l'"humanisation" continue des appareils, en les rendant plus configurables pour refléter les préférences de leurs propriétaires, y compris le sens humain des coûts. Par conséquent, dans cette thèse nous étudions la notion de la confiance dans les données dans un réseau éphémère avec des nœuds rationnels.

La définition d'une nouvelle notion requiert la spécification d'une base, de mesures, et d'une raison d 'être. Dans les chapitres suivants, nous nous penchons sur ces questions. Nous commençons par la définition des modèles du système et de la sécurité informatique dans un réseau éphémère exemple, un réseau véhiculaire. Ensuite, nous étudions le sujet de la révocation dans les réseaux véhiculaires avant de créer et d'analyser un modèle de la révocation en utilisant la théorie des jeux. Dans ce modèle, la notion d'appareils conscients des coûts fait sa première apparition dans cette thèse. Ce modèle rend possible la comparaison de différents mécanismes de révocation existants et conduit à la conception d'une solution optimale, le protocole RevoGame. Avec l'architecture de sécurité en place, nous définissons formellement la confiance dans les données et nous comparons plusieurs mécanismes d'évaluation de cette confiance. En particulier, nous appliquons la Théorie Dempster-Shafer à des cas de grande incertitude. Enfin, nous démontrons que la confiance dans les données peut réduire la perte de confidentialité, rendue inévitable à cause du mécanisme de l'établissement de la confiance. Nous créons d'abord un modèle du compromis entre la confiance et la confidentialité avant de l'analyser, avec l'aide de la théorie des jeux, dans un milieu contenant des entités qui tentent de préserver leur confidentialité. Notre analyse démontre que des récompenses adéquates peuvent atteindre cet insaisissable compromis.

Mots-clés: confiance dans les données, réseau éphémère, VANET, rationalité, théorie des jeux, révocation, confidentialité.

Acknowledgments

First and foremost, I am grateful to Prof. Jean-Pierre, my supervisor, for his guidance, support and patience during the whole PhD process. I have learned from him that PhD is not merely about research, it is a professional and personal development process.

I would like to thank my thesis committee members Prof. Boi Faltings, Prof. Virgil D. Gligor, Dr. Albert Held and Prof. Alcherio Martinoli for their time and effort spent reviewing this document. Special thanks to Prof. Gligor during whose stay at EPFL we elaborated the idea of data-centric trust.

My previous and present colleagues at LCA1 have also contributed to this work by co-authoring my papers, reviewing them, discussing ideas, and generally providing a stimulating work environment. And in the background, our system administrators and secretaries have made many complicated things look simple. I appreciate all their help.

My PhD life would not be a life without the many great moments that I have spent with my friends from the doctoral school. They have made the time spent here one of the best periods of my life. Thank you!

In security, Mallory is the attacker. In my life, she is a safe heaven. I was also lucky to be able to rely on my family's support all along the way. My father was proud each time I went to present a paper. Lena and Iyad lifted my spirit each time I was down. I wish my mother had lived to enjoy seeing me receive my diploma, but life has never been known for being fair. They deserve my eternal gratitude and love.

Contents

Chapter 1

Introduction

Never trust the teller. Trust the tale.

D. H. Lawrence

While writing this thesis, I received a call from a travel agent who said I had won a discounted vacation. I had indeed registered my name for a drawing, so the offer seemed possible and I played along. The travel agent described the offer, which sounded great until she asked for my credit card number. Sounding incredulous, I asked whether she could prove her identity; she affirmed, gave me her name and employee identification and proposed to call back in 20 mins. I phoned her employer and "authenticated" her. The next step was to verify what she was actually saying about her great offer and this is where Internet search came into the picture. Effortlessly I found a few forums that discussed the agency's marketing methods. The opinions ranged from "best of lifetime experiences" to outright accusations of fraud. The trouble with online forums is that it is impossible to distinguish previous or current company employees from disgruntled customers.[1] Sifting through the various data sources with different levels of evidence support and credibility, backed by what I knew about scams, I finally witnessed a real one, and pretty good at that. Making a decision before the agent called back was straightforward. The more difficult and interesting question to answer was: How would a computer behave? I suspect that, with a successful authentication and no previous interaction, a protocol-following computer agent would have properly given away my credit card number. Why? Because thus far, trust establishment in computer interactions has focused on building reputation in specific entities over a series of interactions. But in the scam incident, I had to make a quick decision based on input from people I had never interacted with before. In addition, my decision was not anymore about the legitimacy of my interlocutor but about the correctness of what she was saying. This was a problem of *data-centric trust* establishment in an *ephemeral* environment and this thesis provides a solution for the protocol-following computer. But this thesis is not about phone scams. The basis of this work lies in two future technological visions: VANETs (Vehicular Ad Hoc NETworks) and rational software agents.

In VANETs, vehicles will be able to communicate with each other via a WiFi-like interface installed in each vehicle. This will enhance a vehicle's awareness of its environment and thus reduce road accidents and optimize traffic flows. Allowing each and every vehicle to

[1] This is best illustrated by the famous cartoon, from The New Yorker, picturing two dogs in front of a computer and saying: "On the Internet, nobody knows you're a dog".

influence the behavior of other vehicles, however, has raised concerns in the security community: Launching a Denial-of-Service (DoS) attack on a VANET would be possible from the command line of a properly equipped computer. This is why we began working on the topic of VANET security. There were many problems to solve, including key management, authentication, and privacy. But many of them have been already extensively addressed, albeit with somewhat different requirements, in the context of ad hoc and sensor networks. What seemed to set VANETs apart was the issue of trust: As VANETs are mainly about distributing data on road conditions, it is crucial to establish the trustworthiness of this data. Yet, the existing notions of trust that consistently build the reputation of nodes do not fit the new environment. The millions of vehicles moving around a continent and encountering each other perhaps for only a brief moment create *ephemeral networks*, characterized by their large scale and the high mobility of nodes. This means that building a reputation system for such a large-scale system is excessively complex, if not outright impossible. This is why we defined data-centric trust, the notion that trust should be established in the data itself rather than in its source.

Rational software agents are the translation of a person's preferences to a software agent acting on her behalf. Nowadays, computers and especially smartphones are highly personalized and could well perform transactions on the behalf of their owners. For example, auction sites already allow bidders to configure automatic increments to their bids. And in many cases, especially those involving monetary tradeoffs, people, and hence their configured agents or cars, try to rationally "optimize their utilities".[2]

Based on the above, we invite the reader to look at this work through two lenses: ephemeral networks and rational agents. This is basically the system model. The problem we solve is: **How to establish data-centric trust in an ephemeral network with rational nodes.** The answers, provided in the chapters of this thesis, dissect this problem and provide solutions to its constituents. Our contributions can be summarized as follows:

1. We propose a security architecture for VANETs. This implies the need for being able to authenticate nodes and revoke them if they misbehave. And although the first problem has straightforward solutions derived from existing systems, such as ad hoc networks, the problem of revocation requires a closer look. Hence, we propose a set of revocation protocols and, in particular, a distributed solution - RevoGame - that takes both ephemerality and rationality into account. We designed RevoGame based on a game-theoretic framework that we developed to select the optimal revocation strategies in an ephemeral network. This framework uses cost as a criterion for choosing the appropriate strategies and enables the analytical comparison of existing revocation solutions.

2. We define and formalize data-centric trust by providing a closed-form trust computation function. In addition to using traditional metrics of entity-centric trust, such as authenticated identifiers and reputation, as its input, this function operates on variables related to the data itself. These data-specific inputs include the consistency of data over all its sources and the relevance (e.g., in time or space) of the respective data elements to the decision in question. For example, a vehicle receiving an accident warning will rely on several reports recently generated by vehicles in the vicinity of the accident.

3. We analyze the costs of data-centric trust establishment. Thus far, we have talked about rational utility-optimizing agents without specifying the nature of this utility.

[2] Although we can also be "predictably irrational" [13].

An obvious option is *privacy*, something that many people try to protect. In this case, utility optimization means the minimization of privacy loss: Although it is hard for a human to reveal the optimal amount of private information, a properly configured software agent is more capable of doing so. Hence, we study the trust-privacy tradeoff problem and design mechanisms for solving it. We show that rational agents release private information, needed for trust establishment, only when compensated with proper incentives.

Outline of the Thesis

In Chapter 2, we set the ground for the remainder of the thesis by describing a security architecture for VANETs, our running example of ephemeral networks. This architecture is among the first, especially in academia, to cover topics including key management, secure hardware, and privacy. We also elaborate several attack scenarios, both basic and sophisticated. And even though four years have elapsed since we first published this architecture, it remains up-to-date, with several components being reused or actively discussed in more recent publications.

In Chapter 3, we cover in more detail the issue of revocation in VANETs. We describe three revocation protocols that make use of infrastructure (RTC and RC^{2}RL) or compensate the lack thereof (LEAVE). The latter protocol addresses the most interesting, and perhaps probable, VANET scenario, specifically a network with sparse infrastructure. At the time we developed LEAVE, which is based on weighted voting, Tyler Moore et al. at the University of Cambridge elaborated an alternative solution, based on self-sacrifice, for revocation in ad hoc networks. We set to compare the two and obtained, using simulations, mixed results, i.e., none of the two solutions was consistently better than the other one [72]. This led us to the next investigation.

In Chapter 4, we look at the problem of distributed revocation from an analytical point of view. When beginning our analysis, we discovered that there were no analytical frameworks for distributed revocation, but rather a set of competing solutions, often supported by simulation studies. We address this need by introducing an economic model, which will be the reader's first encounter with *game theory* in this thesis. The model we develop allowed us to create the RevoGame protocol that outperforms the solutions mentioned in the previous paragraph.

In Chapter 5, we introduce data-centric trust, assuming the security architecture described in the previous chapters is in place. We propose a trust computation function and instantiate it in the VANET scenario. We also evaluate the capability of several decision logics to combine trust values from different sources. One of our novel contributions in this chapter is the use of the *Dempster-Shafer Theory* (DST) to cope with high uncertainty scenarios.

In Chapter 6, we investigate the costs of trust establishment. Hence, we put on again the economics lens and analyze the tradeoff between trust and privacy, the most probable resource to exchange for trust. We mathematically model this tradeoff, albeit on a simplified example, and show that data-centric trust can actually improve privacy by requiring less information to be revealed than the traditional entity-centric trust. In addition, keeping in mind the rationality of agents trying to minimize their privacy loss, we derive the conditions under which such agents are willing to contribute to trust establishment. We actually find that without proper incentives, agents will simply restrain from contributing.

Chapter 2

The Security of Vehicular Networks

2.1 Introduction

Until recently, road vehicles were the realm of mechanical engineers. But with the plummeting costs of electronic components and the permanent willingness of the manufacturers to increase road safety and to differentiate themselves from their competitors, vehicles are becoming "computers on wheels", or rather "computer networks on wheels". For example, a modern car typically contains several tens of interconnected processors; it usually has a central computer as well as an EDR (*Event Data Recorder*), reminiscent of the "black boxes" used in avionics. Optionally, it also has a GPS (*Global Positioning System*) receiver, a navigation system, and one or several radars.

Manufacturers are about to make a quantum step in terms of vehicular IT, by letting vehicles communicate with each other and with roadside infrastructure, thus creating VANETs (Vehicular Ad-hoc NETworks); in this way, vehicles will dramatically increase their *awareness* of their environment, thereby increasing safety and optimizing traffic. Researchers have investigated many aspects of VANETs [16, 34, 35, 43, 44, 51, 65, 96, 98, 100].

Considering the tremendous benefits expected from VANETs and the huge number of vehicles (hundreds of millions worldwide), it is clear that VANETs are likely to become the most relevant realization of mobile ad hoc networks. The appropriate integration of onboard computers and positioning devices, such as GPS receivers along with communication capabilities, opens tremendous business opportunities, but also raises formidable research challenges.

One of these challenges is security; limited attention [16, 43, 51, 83, 100] has been devoted prior to this work to the security of VANETs. Yet, security is crucial. For example, it is essential to make sure that life-critical information cannot be inserted or modified by an attacker; likewise, the system should be able to help establishing the liability of drivers; but at the same time, it should protect as far as possible the privacy of the drivers and passengers.

These concerns may look similar to those encountered in other communication networks, but they are not. Indeed, the size of the network, the speed of the vehicles, the relevance of their geographic position, the very sporadic connectivity between them, and the unavoidably slow deployment make the problem very novel and challenging. The purpose of this chapter is to describe a comprehensive solution framework to this challenge.

It should be noted that the first applications of VANETs will probably be commercial, such as infotainment services provided by the infrastructure. Yet, it is very important to

establish the foundations of security for the next steps (vehicle-to-vehicle communications) because industrial consortia are already working on the standards.

2.2 Related Work

The research on VANET security has recently started to gain momentum. When we started working on this topic, there were only few pioneer papers in the academic literature and one proposal in the industry.

El Zarki et al. [100] describe an infrastructure for VANETs and briefly mention some related security issues and possible solutions. In [16], Blum and Eskandarian elaborate a security architecture for VANETs intended mainly to counter the so-called "intelligent collisions" (meaning that they are intentionally caused). They propose the use of a PKI and a virtual infrastructure where cluster-heads are responsible for reliably disseminating messages (by a sequential unicast instead of broadcast) after digitally signing them. Parno and Perrig [79] discuss the challenges, adversary types and some attacks encountered in VANETs; they also describe several security mechanisms that can be useful in securing these networks. Other papers focus on particular VANET security subjects, notably the detection of incorrect data. Golle et al. [44] propose comparing received data to a *model of the VANET* and accepting their validity if both agree. Leinmüller et al. describe several heuristic tests to verify the position information provided by other vehicles. Shen et al. propose security architectures based on group signatures [64] and identity-based batch verification [25]. A broader coverage of security in automotive IT systems can be found in the survey by Wolf et al. [93].

On the industry side, the IEEE P1609.2 standard [5] proposes using asymmetric cryptography to sign safety messages with frequently changing keys so that privacy is preserved. There is no mechanism proposed for certificate revocation. Instead, certificates have short lifetimes and are periodically requested by vehicles through roadside base stations, implying the need for a pervasive infrastructure. The NoW (Network on Wheels) projects partially addresses VANET security. Gerlach et al. describe the resulting security architecture in [41]; among other mechanisms, they propose using plausibility checks and confidence values for data verification, although without providing further details. Last but not least, VANET security is fully addressed by the European project SeVeCom (Secure Vehicular Communications) that provides a definition and implementation of security mechanisms for VANETs [37, 76].

Table I summarizes the mechanisms used to provide security features in VANETs and compares them with other network types that are broadly addressed in the literature. We can see that the distinctive properties of VANETs, notably scale and high mobility, justify the need for, as well as the opportunity of, using novel solutions compared to other network types.

2.3 System Model

In this section, we present the distinguishing properties of VANETs (Fig. 2.1) in order to express later the problem statement. Further, we describe a basic safety messaging protocol to be used as a reference in later sections.

Features	Network type			
	Cellular, WLAN	*Sensor Networks*	*P2P (PGP)*	*VANET*
Key Management	symmetric, centralized	symmetric, centralized	asymmetric, decentralized	asymmetric, multiple authorities
Authentication	authentication server	pairwise symmetric	digital signatures, web of trust	digital signatures, CA certificates
Revocation	directly by the operator	distributed voting	counter-certificates	short-lived certificates; CRLs
Privacy	temporary identifiers	NA	anonymizing services	preloaded keys
Positioning	triangulation with base stations	triangulation with beacons	NA	open problem

Table 2.1: Comparison of different network types with respect to security problems. It should be noted here that there exist several mechanisms proposed for some network types, but we consider the most widely adopted of these. Thus, for example, we took Pretty Good Privacy (PGP) as a representative example of peer-to-peer (P2P) security in the Internet.

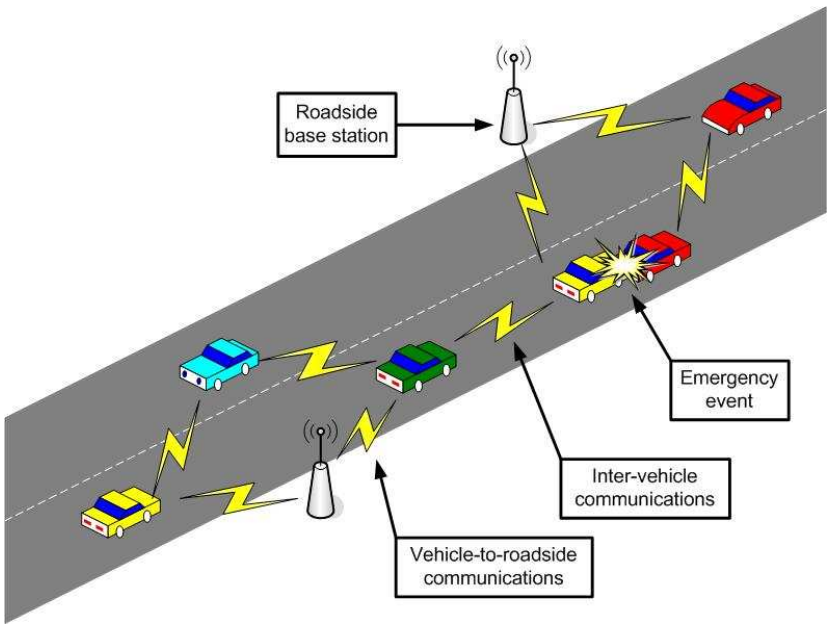


Figure 2.1: A VANET consists of vehicles and roadside base stations that exchange primarily safety messages to give the drivers the time to react to life-endangering events.

2.3.1 System Assumptions

In the rest of this thesis, we will use the following assumptions on VANETs.

Network Model

The communicating nodes in VANETs are either vehicles or base stations. Vehicles can be private (belonging to individuals or private companies) or public (i.e., public transportation means, e.g., buses, and public services such as police cars). Base stations can belong to the government or to private service providers. We assume a communication channel supported by an IEEE 802.11-like technology.

Given that the majority of the network nodes will consist of vehicles, the network dynamics will be characterized by quasi-permanent mobility, high speeds, and (in most cases) very short connection times between neighbors (e.g., in the case of crossing vehicles). For example, on highways vehicle speeds are usually higher than 80km/h (with relative speeds equal to twice these values), and in some countries (e.g., Germany) are not even upper bounded. Another aspect of network dynamics is that vehicle trajectories are mostly well defined by the roads, which incurs some advantages (for message dissemination) and disadvantages (for privacy).

The scale of VANETs is another feature that sets them apart. With hundreds of millions of nodes distributed everywhere, VANETs are likely to be the largest real-world mobile ad hoc network. But communication in this network will be mainly local, thus partitioning the network and making it scalable.

An advantage of VANETs over "usual" ad hoc networks is that vehicles provide substantial computational and power resources, especially taking into account Moore's law and the related improvement of computing platforms in the next few years. As mentioned in the Introduction, a typical vehicle in a VANET will host several tens or even hundreds of microprocessors, an EDR that can be used for crash reconstruction, and a GPS receiver (or a similar system, such as Differential GPS or Galileo) that will provide position and a clock.

VANETs should become partially operational with the release of first products in the next few years. This means that the basic functions of VANETs and the related security mechanisms should be available even with low market penetration, and especially without relying on the existence of an omnipresent infrastructure supporting safety features (which will take a longer time to deploy due to administrative and installation costs).

Application Categories

There are many applications envisioned for VANETs, most of which are proposed by the vehicle manufacturers. Although the spectrum of these applications is very wide ranging (from the realistic to the futuristic), we have divided the applications into two major categories:

1. Safety-related applications, such as collision avoidance and cooperative driving (e.g., for lane merging). The common characteristic of this category is the relevance to life-critical situations where the existence of a service may prevent life-endangering accidents. Hence the security of this category is mandatory, since the proper operation of any of these applications should be guaranteed even in the presence of attackers.

2. Other applications, including traffic optimization, payment services (e.g., toll collection), location-based services (e.g., finding the closest fuel station), infotainment (e.g., Internet

access). Obviously, security is also required in this application category, especially in the case of payment services. But in this work we focus on the security aspects of safety-related applications because they are the most specific to the automotive domain and because they raise the most challenging problems.

Safety Messages

As explained in the previous section, we consider only safety applications. A common property of all the messages is that they are geocast and mainly standalone (i.e., there is no content dependency among them like in media streams). The content of a typical safety message includes position, speed, direction, and acceleration of the vehicle, in addition to data specific to traffic events (e.g., congestion notification or accident). If the sender faces an abnormal situation (e.g., an accident), these data help receivers compute their positions with respect to the sender and determine if they are in danger. The message does not necessarily contain explicit ID information.

An important feature of ad hoc networks is multihopping. But according to the DSRC (Dedicated Short Range Communications) specifications [4] and because of their broadcast nature, safety messages are transmitted over a single-hop with a sufficient transmission power to warn vehicles in a range equal to the distance travelled in 10 seconds at the sender's speed, thus eliminating the need for multihop. Nevertheless, some form of multihop still exists: vehicles that receive warning messages estimate whether the reported problems can also affect their followers; in this case, they forward the messages to them.

Default Trust

A key element in a security system is trust. This is particularly emphasized in VANETs because of the high liability required from safety applications and consequently the nodes running these applications. Due to the large number of independent network members (i.e., they do not belong to the same organization) and the presence of the human factor, it is highly probable that misbehavior will arise. In addition, consumers are becoming increasingly concerned about their privacy. Drivers do not make an exception, especially because the lack of privacy and the related potential of tracking may result in fines on the drivers (e.g., due to occasional overspeeding). As a result, we assume a low level of trust in vehicles, as well as service provider base stations. Beside drivers and service providers, there will be a considerable presence of governmental authorities in VANETs. But due to the reasons stated above, trust in any of these authorities will be limited (e.g., a given police officer may abuse his authority if given full trust). To gain full trust, several authorities will have to cooperate as will be sketched in Section 2.5.

2.3.2 Basic Safety Messaging Protocol

To better describe the security solutions introduced in this chapter, we describe in the following a simple protocol inspired from [98] for safety messaging to use as an example reference in later sections.

- In compliance with the DSRC specifications, we assume that each vehicle V periodically sends messages over a single hop every 300 ms within a variable range that depends on the vehicle's speed (the minimum range is 110 m and the maximum is 300 m).

- The inter-message interval drops to 100 ms and the range to 15 m if the vehicles are very slow or stopped (i.e., their speed is less than 10 miles/h or $\approx$ 16 km/h).

- Vehicles take decisions based on the received messages and may transmit new ones. For example, if a vehicle V receives an emergency warning from another vehicle W and, based on their mutual positions, estimates that it is also in danger, it sends out its own warning messages.

2.4 Attacks on Vehicular Networks

In this section we describe the security threats facing VANETs. Since we cannot envision all the possible attacks that will be mounted in the future on VANETs, we will provide a general classification of attacks substantiated by a list of attacks that we have identified so far. But before describing the attacks, it is important to define the attacker, which we do in the following section.

2.4.1 Attacker Model

To classify the capacities of an attacker, we define four dimensions:

1. *Insider* vs. *Outsider*. The insider is an authenticated member of the network that can communicate with other members. As will be explained later, this means that it possesses a certified public key. The outsider is considered by the network members as an intruder and hence is limited in the diversity of attacks it can mount.

2. *Malicious* vs. *Rational*. A malicious attacker seeks no personal benefits from the attacks and aims to harm the members or the functionality of the network. Hence, it may employ any means disregarding corresponding costs and consequences. On the contrary, a rational attacker seeks personal profit and hence is more predictable in terms of the attack means and the attack target.

3. *Active* vs. *Passive*. An active attacker can generate packets or signals, whereas a passive attacker contents itself with eavesdropping on the wireless channel.

4. *Local* vs. *Extended*. An attacker can be limited in scope, even if it controls several entities (vehicles or base stations), which makes him local. An extended attacker controls several entities that are scattered across the network, thus extending its scope. This distinction is especially important in privacy-violating and wormhole attacks that we will describe shortly.

Inspired by [49], we characterize an attacker by *Membership.Motivation.Method.Scope* where *Membership* stands for *Insider* (I_m) or *Outsider* (O_n), *Motivation* for *Malicious* (M) or *Rational* (R), *Method* for *Active* (A) or *Passive* (P), and *Scope* for *Local* (L) or *Extended* (E); m and n indicate the numbers of I and O nodes that the attacker controls, respectively. These two numbers also cover the notion of collusion. For example, an attacker $I_2.R.A.L$ controls two networks members, behaves rationally, and mounts active attacks in restricted areas. A star ("*") indicates that the corresponding field can take any value.

2.4.2 Basic Attacks

As this work is concerned with vehicular *networks*, we consider only the attacks perpetrated against messages rather than vehicles, as the physical security of vehicle electronics (e.g., against hardware tampering) is out of the scope of this work.

1. *False information* (Fig. 2.2): Attackers are $I_m.R.A.*$ (m indicates any positive integer) and diffuse wrong information in the network to affect the behavior of other drivers (e.g., to divert traffic from a given road and thus free it for themselves).

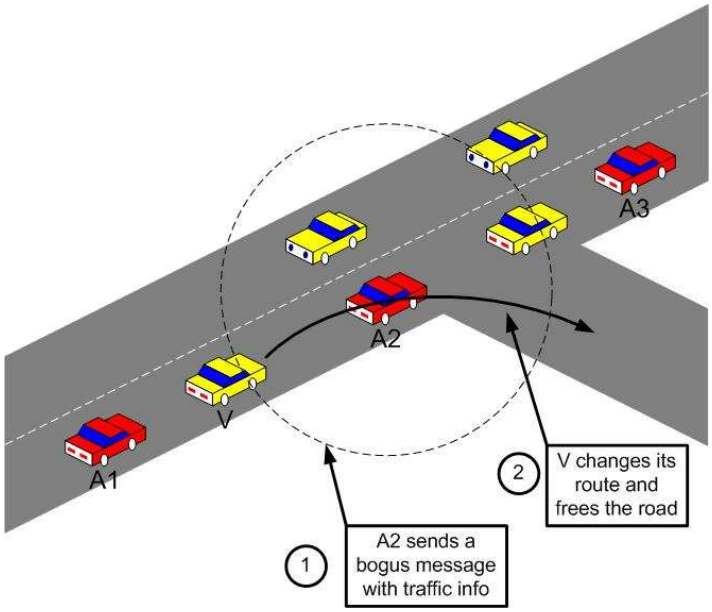


Figure 2.2: In this example *false information* attack, colluding attackers (A2 and A3) disseminate false information to affect the decisions of other vehicles (V) and thus clear the way of attacker A1.

2. *Cheating with sensor information*: Attackers in this case are also $I_m.R.A.L$, and use this attack to alter their perceived position, speed, direction, etc. in order to escape liability, notably in the case of an accident. In the worst case, colluding attackers can clone each other, but this would require retrieving the security material (which should be stored in tamper-proof hardware as discussed in Section 2.5.3) and having full trust between the attackers.

3. *ID disclosure* of other vehicles in order to track their location. This is the Big Brother scenario, where a global observer can monitor trajectories of targeted vehicles and use this data for a range of purposes (e.g., the way some car rental companies track their own cars). To monitor, the global observer can leverage on the roadside infrastructure or the vehicles around its target (e.g., by using a virus that infects neighbors of the target and collects the required data). The attacker is passive. We assume that the attacker does not make use of cameras, physical pursuit, or onboard tracking devices to uncover the identity of its target; otherwise, the tracking problem becomes simpler

but also more expensive and tied to few specific targets, and it can be done anyhow based on existing license plates. In addition, we assume that physical-layer attacks (e.g., using radio fingerprinting [89]) are solved by appropriate physical layer techniques such as radio transmitters that randomize fingerprints.

4. *Denial of Service*: The attacker is $*.M.A.L$ and may want to bring down the VANET or even cause an accident. Example attacks include channel jamming and aggressive injection of dummy messages.

5. *Masquerading*: The attacker actively pretends to be another vehicle by using false identities and can be motivated by malicious or rational objectives.

2.4.3 Sophisticated Attacks

The attacks in this section are more elaborated variants or combinations of the above attacks. They are examples of what an adversary can do. In the context of VANETs, this is the first time these attacks are presented.

1. *Hidden vehicle*: This is a concrete example of cheating with positioning information. It refers to a variation of the basic safety messaging protocol described in Section 2.3.2. In this version of the protocol, a vehicle broadcasting warnings will listen for feedback from its neighbors and stop its broadcasts if it realizes that at least one of these neighbors is better positioned for warning other vehicles. This reduces congestion on the wireless channel. As Fig. 2.3 illustrates, the hidden vehicle attack consists in deceiving vehicle A into believing that the attacker is better placed for forwarding the warning message, thus leading to silencing A and making it hidden, in DSRC terms, to other vehicles. This is equivalent to disabling the system.

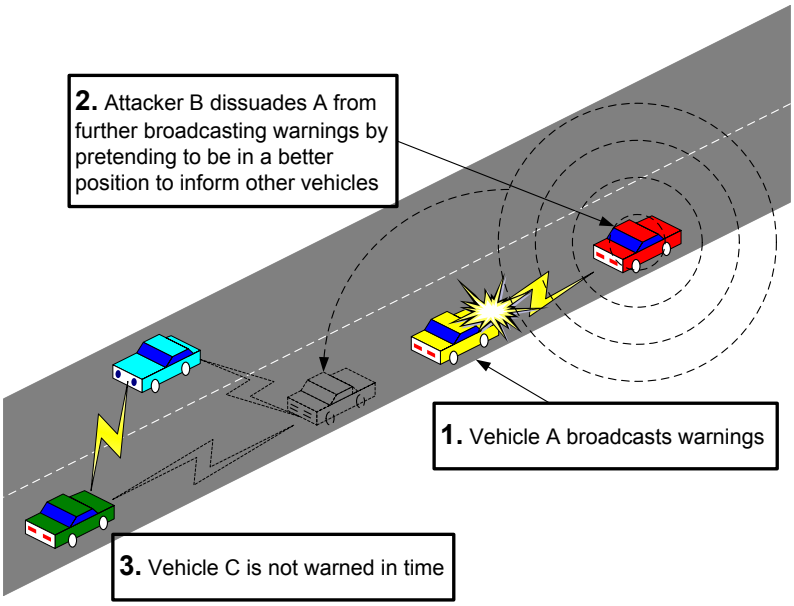


Figure 2.3: Hidden vehicle attack.

2. *Tunnel*: Since GPS signals disappear in tunnels, an attacker may exploit this temporary loss of positioning information to inject false data once the vehicle leaves the tunnel and before it receives an authentic position update as Fig. 2.4 illustrates. The physical tunnel in this example can also be replaced by an area jammed by the attacker, which results in the same effects.

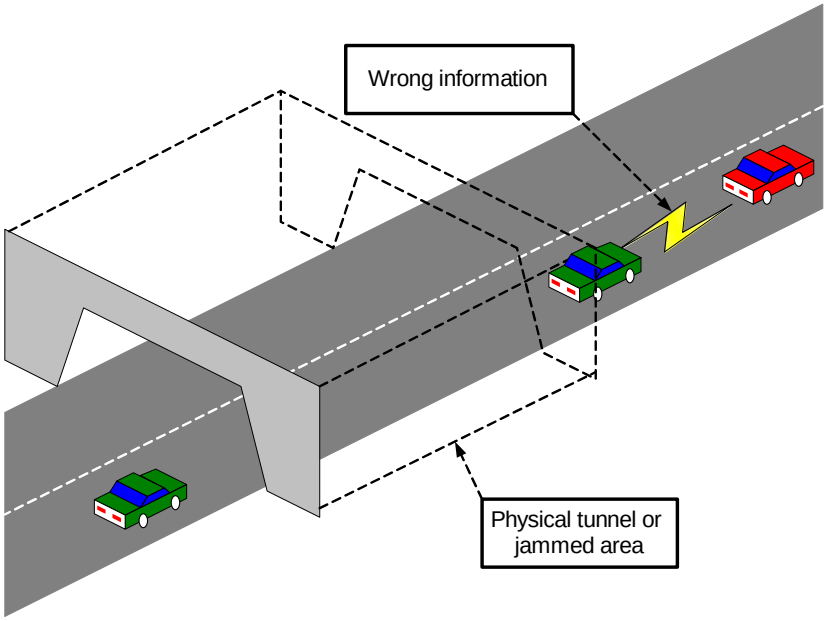


Figure 2.4: Tunnel attack.

3. *Wormhole*: In wireless networking, the wormhole attack [50] consists in tunneling packets between two remote nodes. Similarly, in VANETs, an attacker that controls at least two entities remote from each other and a high-speed communication link between them can tunnel packets broadcasted in one location to another, thus disseminating erroneous (but correctly signed) messages in the destination area.

4. *Bush telegraph*[1]: This is a developed form of the false information attack. The difference is that in this case the attacker controls several entities spread over several wireless hops. Similarly to the social phenomenon of information spreading and its en-route modification, this attack consists in adding incremental errors to the information at each hop. While the errors are small enough to be considered within tolerance margins at each hop and hence accepted by the neighbors, the intentional accumulation of these errors may yield to a false information at the last hop.

[1]Bush telegraph stands for the rapid spreading of information, rumors, etc., usually by word of mouth. As this information is propagated along a human chain, it is frequently modified by each person in the chain. The result may sometimes be completely different from the original.

2.5 Security Architecture

In this section, we present the components needed to protect VANETs against a wide range of threats, some of which are described in the previous section. Fig. 2.5 depicts the general architecture, the requirements and components of which are described next.

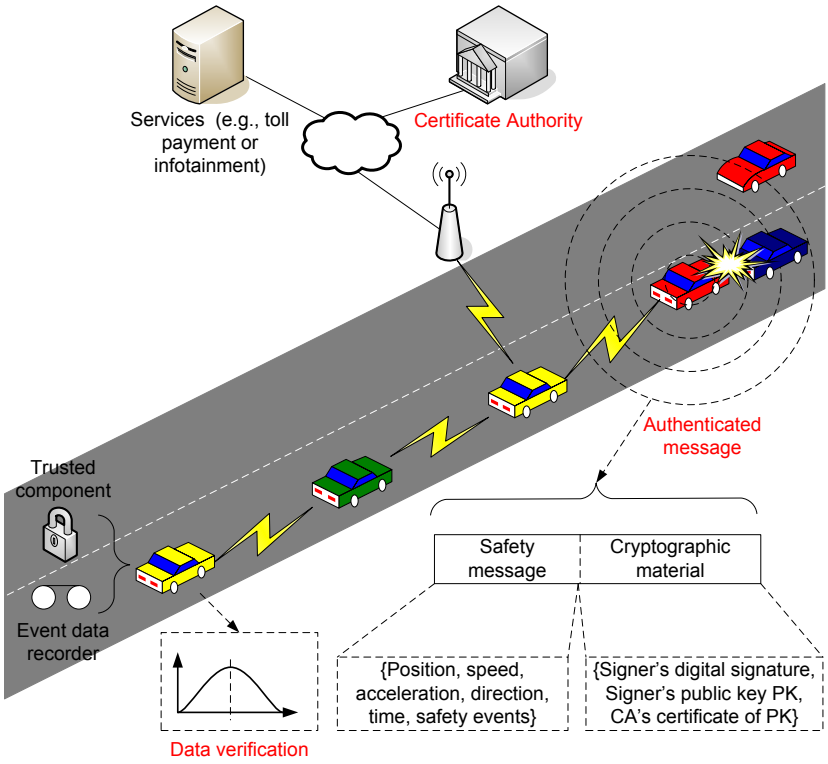


Figure 2.5: Overview of the security architecture

2.5.1 Requirements

A security system for safety messaging in a VANET should satisfy the following requirements:

- *Authentication*: Vehicle reactions to events should be based on legitimate messages (i.e., generated by legitimate senders). Therefore we need to authenticate the senders of these messages.

- *Verification of data consistency*: The legitimacy of messages also encompasses their consistency with similar ones (those generated in close space and time), because the sender can be legitimate while the message contains false data. This requirement is sometimes called "plausibility".

- *Availability*: Even assuming a robust communication channel, some attacks (e.g., DoS by jamming) can bring down the network. Therefore, availability should be also supported by alternative means.

- *Non-repudiation*: Drivers causing accidents should be reliably identified; a sender should not be able to deny the transmission of a message (it may be crucial for investigation to determine the correct sequence and content of messages exchanged before the accident).
- *Privacy*: People are increasingly wary of Big Brother enabling technologies. Hence, the privacy of drivers against unauthorized observers should be guaranteed.
- *Real-time constraints*: At the very high speeds typical in VANETs, strict time constraints should be respected.

2.5.2 Vehicular Public Key Infrastructure

The huge number of vehicles registered in different countries and travelling long distances, well beyond their registration regions, requires a robust and scalable key management scheme. The involvement of authorities in vehicle registration implies the need for a certain level of centralization. Communication via base stations (as in cellular networks) is not enough for VANETs, mainly because vehicles need to authenticate themselves not only to base stations but also to each other (without invoking any server), which creates a problem of scalability. In addition, symmetric cryptography does not provide the non-repudiation property that allows the accountability of drivers' actions (e.g., in the case of accident reconstruction or finding the originators of *false information* attacks). Hence, the use of public key cryptography is a more, if not the only, suitable option for deploying VANET security. But a self-organized trust management approach such as the one in PGP (Pretty Good Privacy) is not satisfactory because of scalability issues.

This implies the need for a *Vehicular Public Key Infrastructure* (VPKI) where Certificate Authorities (CAs) will issue certified public/private key pairs to vehicles (with many pairs per vehicle for privacy reasons as will be explained in Section 2.5.4). Similarly to current vehicle registration authorities, there will be several CAs, each corresponding to a given region (e.g., country, state, metropolitan area, etc.). Other candidates for taking the role of CAs are car manufacturers. In any of the two cases, the different CAs will have to be cross-certified so that vehicles from different regions or different manufacturers can authenticate each other. This will require each vehicle to store the public keys of all the CAs whose certificates it may need to verify. Alternately, in the case where CAs are regional authorities, vehicles may request new public/private key pairs delivered by the foreign region they enter.[2]

Under the PKI solution, before a vehicle sends a safety message, it signs it with its private key and includes the CA's certificate as follows:

$$V \rightarrow * : msg, Sig_V[msg|T], Cert_V$$

where V designates the sending vehicle, $*$ represents all the message receivers, msg is the message, $|$ is the concatenation operator, and T is the timestamp to ensure message freshness (it can be obtained from the security device introduced in Section 2.5.3). $Cert_V$ is the public key certificate of V.

For example, assuming keys are certified by a certain CA, a certificate $Cert_V[PuK_i]$ of the i^{th} anonymous public key PuK_i of a vehicle V should include at least the following:

$$Cert_V[PuK_i] = PuK_i|Sig_{CA}[PuK_i|ID_{CA}]$$

[2]In this context, "foreign" means a region different from a vehicle's home region.

where Sig_{CA} is CA's signature and ID_{CA} is the unique ID of CA.

The receivers of the message have to extract and verify the public key of V using the certificate and then verify V's signature using its certified public key.

If the message is sent in an emergency context, this message should be stored (including the signature and the certificate) in the EDR for further potential investigations in the emergency.

2.5.3 Security Hardware

Among the vehicle onboard equipment, there should be two hardware modules needed for security, namely the *Event Data Recorder* (EDR) and the *Trusted Component* (TC). Whereas the EDR only provides tamper-proof storage, the TC also possesses cryptographic processing capabilities.

The EDR will be responsible for recording the vehicle's critical data, such as position, speed, time, etc., during emergency events, similar to an airplane's black box. These data will help in accident reconstruction and the attribution of liability. EDRs are already installed in many road vehicles, especially trucks. These can be extended to record also the safety messages received during critical events.

The use of secret information such as private keys incurs the need for a TC in each vehicle. In addition to storing the secret information, this device will be also responsible for signing outgoing messages and running security protocols. To reduce the risk of its compromise by attackers, the device should have its own battery, which can be recharged from the vehicle, and clock, which can be securely resynchronized, when passing by a trusted roadside base station. The access to this device should be restricted to authorized people. For example, cryptographic keys can be renewed at the periodic technical checkup of the vehicle. The TC may be a tamper-resistant device containing a set of sensors that can detect hardware tampering and erase all the stored keys to prevent them from being compromised. But tamper-resistance would make the TC too sensitive for VANET conditions (for example, the device will be exposed to extreme temperatures that may not be unusual for vehicles) and too expensive (current commercial products such as the IBM 4764 card [1] cost several thousands of dollars). But recent advances in the design of such devices [10] suggest that sufficiently performing, yet reasonably priced, products will be available on the market. An alternative option to a tamper-resistant device could be a TPM (Trusted Platform Module [2]) that can resist software attacks but not sophisticated hardware tampering. Such units are gaining wide use in personal computers, can have internal clock and battery, and cost only a few tens of dollars. Hence, the actual implementation of the TC will depend mainly on economic and technical factors. This component may actually be a compromise between a tamper-resistant device and a TPM. Our goal in this work is to set the operational requirements that can guide later the choice or design of such a device.

2.5.4 Key Management

We will address below the issues of cryptographic key distribution and certification. The next chapter will be devoted to key revocation.

Cryptographic Information Types

To be part of a VANET, each vehicle has to store the following cryptographic information:

1. An *Electronic License Plate* (*ELP*) [51] issued by a government, or alternatively an *Electronic Chassis Number* (*ECN*) issued by the vehicle manufacturer. These ELPs should be unique and cryptographically verifiable (this can be achieved by attaching a certificate issued by the CA to the ELP) in order to identify vehicles to the police in case this is required (usually, ELPs are hidden from the police). Similarly to the physical license plates, the ELP should be changed (i.e., reloaded in the vehicle) when the owner changes or moves, e.g., to a different region or country.

2. *Anonymous key pairs* that are used to preserve privacy. An *anonymous key pair* is a public/private key pair that is certified by the CA but contains neither information about nor public relationship with (i.e., this relationship cannot be discovered by an observer without a special authorization) the ELP of the vehicle. Yet this privacy is conditional for liability purposes as will be explained later. Normally, a vehicle will possess a set of anonymous keys to prevent tracking. In addition, the vehicle will have a set of temporary identities, or *pseudonyms*, each corresponding to an anonymous key. A pseudonym can be derived from the corresponding anonymous key, for example using a hash function, or can be the key itself if identity-based cryptography is used.

Key Bootstrapping and Rekeying

Since the ELP is the electronic equivalent of the physical license plate, it should be "installed" in the vehicle using a similar procedure, which means that the governmental transportation authority will preload the ELP at the time of vehicle registration (in the case of the ECN, the manufacturer is responsible for its installation at production time).

Anonymous keys are preloaded by the transportation authority or the manufacturer, but with different consequences as discussed in the next section. Moreover, while ELPs are fixed and should accompany the vehicle for a long duration (potentially its lifetime), anonymous key sets have to be periodically renewed after all the keys have been used or their lifetimes have expired. This renewal can be done during the periodic vehicle checkup (typically yearly) or by similar procedures.

In addition to the ELP and anonymous keys, each vehicle should be preloaded with the CA's public key.

Certificate Lifetime

The *certificate lifetime* should be short to reduce the vulnerability window of the system in case an anonymous public/private key pair is compromised. Each anonymous key should be used only with a sequence of consecutive messages as described in Section 2.6.2; otherwise a global attacker can track a public key that is reused, even on different days, as public keys are sent with messages to allow the verification of digital signatures. As the driving duration changes from day to day (e.g., a long trip on vacation compared to a daily home-work-home trajectory), there should be enough keys to be used on any day. To account for this, the lifetime of a key certificate should be stretched over several days (this is distinct from the usage duration of a key, further explained in Section 2.6.2, which is only several seconds or minutes and aims at protecting the privacy of the key holder). The overlap of certificate lifetimes means that a vehicle can use multiple anonymous keys in a very short period, thus effectively creating a Sybil attack. To limit the effects of such a strategy, the TC invalidates anonymous keys once they are used and can impose an upper bound on the key-changing

frequency. In addition, an attacker mounting a Sybil attack quickly depletes its available keys, requiring a costly renewal.

2.5.5 Anonymous Public Keys

There are several types of privacy. Vehicle owners will be only concerned about identity and location privacy. To respond to these concerns, we propose the use of anonymous public keys that we detail in this section.

Identity and Location Privacy

All vehicle identifiers, in particular MAC and IP addresses, must change over time. And even though anonymous keys do not contain any publicly known relationship to the true identity of the vehicle owners, privacy can still be hijacked by logging the messages containing a given key (as keys are part of the certificates that accompany messages) and thus tracking the corresponding vehicle owner until discovering her identity (e.g., by associating her with her place of living).

Therefore, anonymous keys should be changed in such a way that a pervasive observer cannot track the owner of the keys. The downside of this approach is that a vehicle will have to store a large key and certificate set (depending on the key-changing frequency). In Section 2.6.2 we will propose a variable-frequency key-changing algorithm that can preserve privacy while minimizing the key storage space.

Conditional Privacy

Privacy preservation is a requirement for deploying vehicular safety applications. But safety and the implied liability requirement have higher priority. Hence, privacy should be conditional on the scenario (e.g., if there are issues of law enforcement or national security, privacy should be overriden). But if police (or other law enforcement entities) are given full control over the ID disclosure process, abuse can occur. Hence, the ID disclosure capability should be distributed among multiple authorities (in the same way it is done with other legal issues, such as bank account disclosure). For example, police should not be able to retrieve the identity corresponding to an anonymous key without the permission of a judge. Secret sharing [88] can be used to technically reinforce the distribution of authorizing material among authorities, whereby authorities share the secret needed to access the database that matches true vehicle identities (ELPs) with the set of their anonymous public keys.

2.6 Security Analysis

In the following we analyze how the previously proposed solutions address the requirements stated in Section 2.5.1.

2.6.1 Compliance with the Security Requirements

Authentication of messages is provided by the digital signature of the sender and the corresponding CA certificate. The only guarantee that this provides is that the message comes from a vehicle that was trusted, at least when its cryptographic keys were issued. Nevertheless, these mechanisms ensure that outsiders are not able to send messages to network members.

Availability is hard to guarantee, notably due to the large scale of ephemeral VANETs. Yet, the ways in which an attacker can disrupt the network service are limited: outsiders can only mount jamming attacks. Even in this case, channel or communication technology switching can reduce the impact of such attacks.

Non-repudiation is achieved as follows:

- Vehicles cannot claim to be other vehicles (*masquerade* attack) since all the messages they transmit are signed by their (anonymous) public keys.
- A vehicle cannot deny having sent a message because it is signed by an anonymous key that belongs exclusively to the sender; likewise, the vehicle cannot claim that the message was replayed because a timestamp is included in each message.
- Vehicles cannot cheat about their position and related parameters if a secure positioning solution is used. This is a rather strong assumption as the problem of secure positioning system in VANETs is not solved yet, but position verification solutions [62] are a first step in this direction.

The satisfaction of the privacy requirement is addressed in the next section.

2.6.2 Privacy

In order to preserve the driver's privacy and minimize the storage costs of public keys, we propose a key-changing algorithm that adapts to the vehicle speed and takes into account key correlation by the attacker as described below.

Let us consider a typical tracking scenario where the attacker controls stationary base stations separated by a distance d_{att} and captures all the received safety messages; it can later use these data (including the public keys) to illegally track vehicles. In addition, we assume that the attacker can correlate two keys if the sender moves at a constant speed in the same direction and on the same lane between two observation points (e.g., given the initial position of the target, the attacker can predict its position in the future and confirm this prediction if a message is received at the next observation point with correct predicted speed and position); this is typical of a highway scenario. It should be noted that the following algorithm and analysis apply when there are at least two neighboring targets under observation; otherwise, the tracking of a single target becomes trivial despite the usage of any privacy measures.

Assume the speed of target V is v_t, its transmission range is d_r, and d_v is the distance over which a vehicle does not change its speed and lane (the vulnerability window with respect to the correlation of keys). As Fig. 2.6 illustrates, the vehicle's privacy is vulnerable over a distance equal to $d_v + 2d_r$. This means that it is not worth changing the key over smaller distances because an observer can correlate keys with high probability. This defines the lower bound on the key changing interval T_{key}:

$$min(T_{key}) = \frac{d_v + 2d_r}{v_t} \text{seconds}$$

But if $d_{att} > d_v + 2d_r$, V can avoid being tracked (by changing its key) as long as it does not use the same key for a distance equal to or longer than d_{att}. This in turn defines the upper bound on the key changing interval:

$$max(T_{key}) = \frac{d_{att-2d_r}}{v_t} \text{seconds}$$

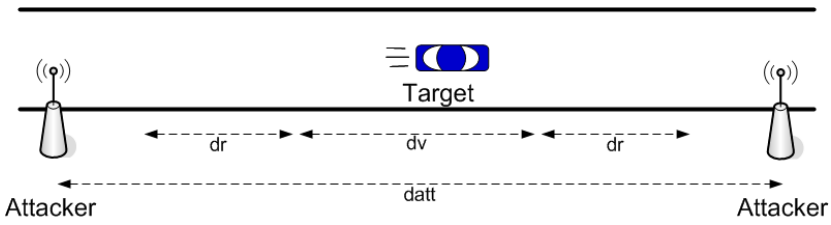


Figure 2.6: To uncover the identity of its targets, the attacker leverages on key correlation and the target's transmission range.

Since V does not know d_{att}, but knows d_r and d_v, it can choose a value of T_{key} that is a little larger than $min(T_{key})$. If we denote by r_m the message rate, one key should be used for at most:

$$N_{msg} = \lceil r_m \times T_{key} \rceil \text{messages}$$

For example, assume $d_{att} = 2$ km, $r_m = 3.33$ msg/s (1 message every 300 ms), $d_v = 30$ s $\times v_t$ (i.e., V does not change its lane and speed during 30 s), $d_r = 10$ s $\times v_t$ (according to DSRC, the transmission range is equal to the distance travelled in 10 s at the current speed), and $v_t = 100$ km/h. Then $min(T_{key}) = 50$ s and $max(T_{key}) = 52$ s. V can choose T_{key} to be 51 s; as a result, $N_{msg} = 170$ messages.

Leveraging on the above analysis, a vehicle should change its anonymous key only after having used it for a certain number of messages. Reusing the example in 2.6.2, a vehicle should change its key within an interval of around 1 min. If we assume that an average driver uses her car 2 hours per day, the number of required keys per year is approximately 43800, which amounts to around 4.2 Mbytes (assuming a storage space of 100 bytes per key, including its certificate).

2.7 Summary

In this chapter, we cast light on the main problems of VANET security. Starting from a description of several attacks scenarios, we developed a set of requirements and a corresponding security architecture. The main components of this architecture are a security hardware, for storing cryptographic material, and a vehicular public key infrastructure, for enabling scalable offline authentication of vehicles. In addition, we proposed a simple privacy mechanism based on anonymous frequently changing public keys. With these components in place, this chapter sets up the general framework for the mechanisms that we will introduce in the next chapters.

Chapter 3

Revocation of Misbehaving Nodes in Vehicular Networks

3.1 Introduction

In this chapter, we develop revocation mechanisms for VANETs. In fact, the possession of a certificate does not guarantee that its holder will provide correct information: a node can simply inject faulty data (e.g., alerts, warnings, coordinates) while complying with the implemented protocols. Safeguarding the system against such faulty or compromised nodes is crucial for its robustness. A typical approach for achieving this is the revocation of node certificates; once this is done, messages from these nodes will be ignored.

Timely access to revocation information is a particularly hard problem in VANETs. The road-side infrastructure can act as the gateway of the Certification Authority (CA) to the network, distributing the latest Certificate Revocation Lists (CRLs) [48]. The lack of an omnipresent road-side infrastructure, especially in the early deployment stages, and the huge scale of the VANETs are obstacles to the application of traditional certificate revocation schemes. Moreover, unless a node is revoked for administrative reasons (e.g., the vehicle owner did not renew its registration), how can the authority obtain and validate sufficient evidence that a node is faulty or compromised? Thus, an additional challenge is how non-misbehaving nodes can be protected until they obtain the revocation information regarding misbehaving nodes.

Our contributions in this chapter address these problems. We propose the combination of (i) infrastructure-based revocation protocols, Revocation of the Trusted Component (RTC) and Revocation using Compressed Certificate Revocation Lists (RC^{2}RL), and (ii) a Local rEvocation of Attackers by Voting Evaluators (LEAVE) protocol to safeguard the system operation, until the attacker is revoked by the CA, partially or fully based on the evidence LEAVE provides.

We emphasize however that no group of nodes has the power to revoke another node. The CA is the sole entity with the right to initiate a revocation protocol. This design choice ensures resilience to collusion attacks, retains accountability, and yet equips nodes with a rapid reaction and self-protection tool.

3.2 Related Work

Revocation has been considered mostly in the context of the wireline Internet and the design of Public Key Infrastructure (PKI) services [48]. Nevertheless, the design of mechanisms to disseminate the revocation information across systems similar to VANETs has not been considered in the wireline Internet context (for a survey and discussion of tradeoffs see [94, 101]). Due to the network volatility and scale, the overhead of querying a server to obtain timely revocation status, assuming the server is reachable, could be impractically high. For the same reasons, schemes that distribute the load of a server to a set of participating clients [95] (to redundantly forward revocation information) would not be practical for deployment within the VANET, but only meaningful behind the fixed infrastructure.

Existing works on VANET security [79, 100] propose the use of a PKI and digital signatures but do not provide any mechanisms for certificate revocation, even though it is a required component of any PKI-based solution. Different aspects of revocation were discussed in [77, 26] without a complete solution provided. In the context of VANETs, the IEEE 1609.2 Draft Standard [5] is the only reference on certificate revocation. It proposes the distribution of CRLs and short-lived certificates, but does not elaborate how to achieve this. Short-lived certificates are also proposed in [52]. Short lifetimes are essentially a means of revocation that achieves efficiency but opens a vulnerability window if there is no mechanism to revoke a certificate before its expiration; such an approach is not appropriate for a life-critical VANET environment.[1] Moreover, certificates have to be refreshed frequently to keep the vulnerability window very small. This could create high loads both on the CA and the network.

Instantiating a CA in the context of mobile ad hoc networks was investigated, with the distribution of its functionality to a number of servers [102]. However, this scheme does not consider the problem of revocation, especially in a highly mobile environment like a VANET. Instantiation of the CA functionality (or part thereof) by impromptu coalitions of network nodes (e.g., [31, 60]) cannot be applicable in VANET systems. Allowing any ad hoc and, in general, small subset of adversarial nodes to maliciously accuse and revoke legitimate nodes would be an unacceptable breach of the VANET system security where accountability and liability are mandatory.

3.3 System Model

The system model of this chapter is based on the architecture of the previous chapter, in addition to several new elements described below.

Safety messages that need to propagate across multiple hops (and perhaps have the originator's signature, coordinates and time intact as they propagate) are signed and include the coordinates and timestamp of the last relaying node. This ensures the freshness of the information and limits the propagation of false information.

We assume that the DSRC protocol [4], being standardized as IEEE 802.11p, is used, unless noted otherwise. Beyond DSRC, VANETs can leverage on other wireless communication technologies, such as the (licensed-frequency) existing cellular networks, broadband wireless (e.g., WiMax), or low-speed radio broadcast systems used today for traffic information.

[1] An exception can be context-specific credentials, allocated, for example, to a vehicle entering a highway segment and "purchasing" access to a service. However, this is orthogonal to the problem we are considering here.

We denote a subset of the network nodes as the infrastructure, comprising the short-range DSRC base stations and mobile units. The latter include public safety vehicles (e.g., highway assistance and fire-fighting vehicles), police vehicles, aerial vehicles (e.g., police helicopters), and public transport vehicles (e.g., buses, trams). In our context, these nodes can be used, for example, to disseminate CRLs. Infrastructure nodes serve as the gateway of the CA to/from the VANET; the connection of the CA to the static infrastructure nodes is over wireline secure links. We note however that accessibility of the CA from the VANET is not assumed to be guaranteed at all times.

3.3.1 Adversary Model

We term as an adversary or attacker any node that deviates from the legitimate VANET protocols. Nodes can also be faulty due to failures of their equipment. A detailed discussion of adversary and fault models is given in [78]. Any of these attacks or faults, or combinations thereof, can affect the VANET-enabled applications. We also refer to adversaries as *misbehaving nodes*. As our proposed mechanisms apply to both misbehaving and faulty nodes, we will use both terms interchangeably in the remainder of this chapter without losing the generality of the solutions.

In addition, the information-oriented operation of VANETs, with their diverse data types, makes *false information dissemination* a very effective attack, compared to deviations from the networking protocols. In fact, it would suffice for an adversary to manipulate the sensory inputs rather than compromise the protocol stack and the computing platform [78]. It is also possible that an attacker controls incoming communication, e.g., by selectively erasing messages received by its onboard platform. We emphasize that we are concerned with misbehaving nodes equipped with valid credentials, because they can effectively abuse the system.

3.4 Scheme Overview

Our scheme consists of the following basic components: (i) the centralized revocation of a node by the CA, (ii) the local detection of misbehavior, performed individually by each node and (iii) a distributed, localized protocol for the revocation of an attacker by its neighboring nodes. The scheme with its components is illustrated in Fig. 3.1.

We propose two methods for misbehaving node revocation, initiated by the CA. The first one, RTC (Revocation of the TC, described in Section 3.5.1), leverages on the presence of a TC unit onboard the vehicle. The CA determines that a vehicle V must be revoked and, with the help of the road-side infrastructure, initiates a two-party end-to-end protocol with TC_V, the trusted component of V. The CA instructs the TC to erase all cryptographic material (e.g., keys) it stores and halt its operation upon completion of the protocol. Essentially, this protocol "kills" the TC, depriving the misbehaving node from its cryptographic keys, and thus ensuring that all its messages are ignored by all other correct nodes.

However, RTC is not robust against a sophisticated adversary that controls the communication link between the CA and the TC. If the CA fails in executing RTC (detected by the lack of an acknowledgment), it will revert to the distribution of the revocation information, namely, a CRL, to the VANET (more specifically, to the neighborhood of the revoked vehicle V). This way, the CA invalidates credentials before the end of their lifetime. But the size of CRLs will grow with the size of the VANET and hence is not scalable. To adapt this approach

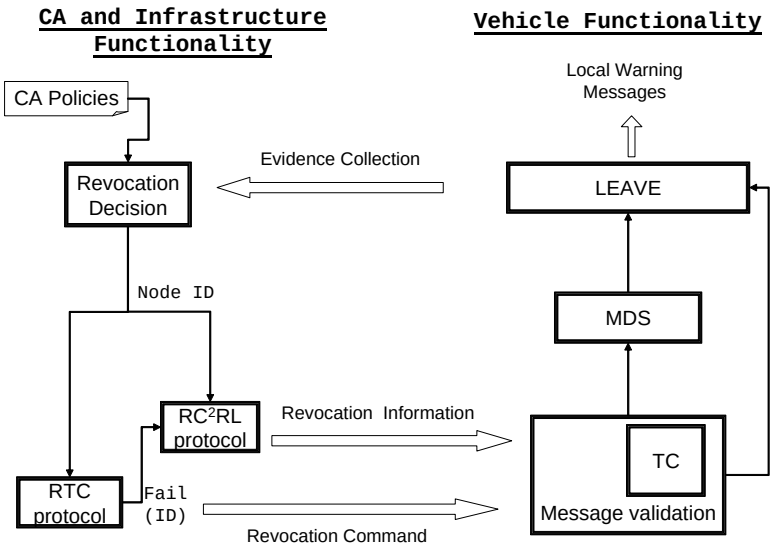


Figure 3.1: Overview of the revocation scheme.

to the VANET scale, we propose the RC^{2}RL (Revocation using Compressed Certificate Revocation Lists) protocol (Section 3.5.2), with Compressed CRLs (C^{2}RLs) being shorter than traditional CRLs by means of Bloom filter compression.

The timely and efficient distribution of revocation information across the VANET is the primary means of revoking misbehaving nodes. However, to design a robust and efficient system capable of progressively isolating misbehaving nodes before this information becomes available, we propose the use of the LEAVE (Local Eviction of Attackers by Voting Evaluators) protocol that runs on each vehicle.

MDS (Misbehavior Detection System) is an essential enabler of LEAVE. Each node uses its own sensory inputs (including time and location), messages received from its neighbors, and a set of evaluation rules, to classify safety messages received from a given node as faulty or correct. Messages that are outdated (aged), received beyond their expected area of propagation, or contradictory to the node's own state[2] are considered false. Their senders, as long as they are neighbors of the node running MDS, are also tagged as misbehaving. Then, their identity is passed to LEAVE. The literature on VANETs already contains methods for adversary detection. For example, threshold-based tests to verify positioning information in VANETs were proposed in [62]. In [44], a more general framework for malicious data detection compares the received data to a *model of the VANET*. We do not elaborate on the MDS any further in this thesis, due to the lack of real data about VANET misbehaviors.

The main principle of LEAVE, detailed in Section 3.5.3, is simple: the neighbors of the misbehaving vehicle temporarily "evict" it. In contrast to RTC and RC^{2}RL, LEAVE is not a revocation protocol, but rather a collective warning system against misbehaving nodes. Upon detecting an attacker, vehicles broadcast *warning* messages to all vehicles in range, so that

[2]For example, a traffic jam message received when the node's velocity in the allegedly jammed area is well above the velocity expected for a traffic jam.

the sharing of information improves the effectiveness of the stand-alone detection systems. Moreover, such warnings can be invaluable when vehicles receive them before being able to observe the misbehaving node themselves.

The revocation of an attacker by its neighbors is temporally limited to the duration of contact between the attacker and its neighbors running LEAVE. But once enough evidence against the attacker is gathered, the CA can initiate one of the previously described revocation protocols. Recall that the CA is the only system entity entitled to revoke keys (due to all the related administrative responsibilities and costs). In this work, we do not consider the CA decision process for node revocation, as a number of legal and policy aspects are involved. In addition, the reasons for revocation are largely orthogonal to the operation itself and can include administrative procedures (e.g., change of registration domain), cryptographic material compromise (e.g., a private key was detectably disclosed) or, as mentioned above, node misbehavior for which the CA obtains sufficient evidence.

3.5 Revocation protocols

3.5.1 RTC

When the CA decides to revoke a vehicle V, it first uses RTC: The CA generates a revocation message that contains V's identity, encrypted with V's public key PuK_V, and a timestamp T; the message is signed by the CA. Thus, TC_V and the RSUs that forward the message can verify its authenticity and freshness. The message format is:

$$CA \overset{RSU}{\rightarrow} TC_V : E_{PuK_V}(V), T, Sig_{CA}[E_{PuK_V}(V), T]$$

where $E_{PuK_V}()$ denotes encryption with public key PuK_V.

There are several options for channeling this message to the TC_V. The first choice would be to route it to the RSU closest to the concerned vehicle, if its location is known to the CA. Otherwise the CA defines a paging area consisting of several RSUs in the region of the vehicle's most recent locations (trajectory extrapolation based on the vehicle's expected speed and acceleration can be useful in determining the paging area). If all else fails, the CA can use other distribution media mentioned in Section 3.3, such as low-speed radio broadcast.

When TC_V receives the RTC message, it immediately erases the cryptographic key and stops signing VANET messages. It sends back a timestamped and signed acknowledgment, as soon as it comes within range of a RSU:

$$TC_V \overset{RSU}{\rightarrow} CA : ACK, T, Sig_{PrK_V}[ACK, T]$$

If the vehicle V is an attacker capable of blocking messages destined to its TC, the CA will receive no acknowledgement and thus will detect the failure of RTC. It will then revert to the RC^{2}RL protocol discussed next.

3.5.2 RC^{2}RL

As CRLs contain very little redundancy, they cannot be efficiently compressed using normal lossless methods. We therefore use Bloom filters [15], a special form of lossy compression, to generate C^{2}RLs (Compressed CRLs) that the CA signs and broadcasts using one of the previously mentioned distribution methods. Bloom filters provide a probabilistic data structure

used to test whether an element is a member of a set. They are characterized by a configurable rate of false positives and no false negatives. This ensures that the CA can efficiently revoke all targeted nodes while keeping false revocations within acceptable error margins. A more detailed explanation of Bloom filters and their application to revocation in VANETs can be found in Appendix 3.A.

3.5.3 LEAVE

As mentioned in Section 3.4, being warned of the misbehaving nodes allows the observing vehicle to ignore any messages sent by these nodes. Warnings can be triggered by the standalone MDSs running on each vehicle. This is the key concept behind LEAVE. More precisely, vehicles that detect an attacker begin broadcasting warning messages to all vehicles in range. The latter can use this information as input to their respective MDSs. In this chapter, we consider the case of vehicles that receive warning messages before they are able to make any observations of the attacker and thus rely entirely on these messages. The final step is to report the attackers or faulty nodes to the CA as soon as possible (i.e., when in reach of a base station or mobile unit as defined in Section 3.3).

LEAVE relies on the collective information gathered from a vehicle's neighborhood. As the vehicle's neighbors can be attackers, the warning messages may contain correct or wrong *accusations*. Hence, it is important to aggregate this warning information while accounting for the presence of attackers. The most popular approach is to use a fixed number of votes [28, 66, 99] or a fixed fraction of nodes (e.g., a majority) [12]. In these cases, all nodes contribute equally to the aggregate warning, leading to the need for assumptions such as honest majority. In addition, the equal contributions miss the additional and useful information that can be retrieved about each node from its context. One example of this information is the reputation of nodes [22]. More generally, we propose attributing to an accusation by node i a *weight* $w_{i,a}$ computed as follows:

$$w_{i,a} = m_a \times \frac{t_i}{t_a} \tag{3.1}$$

where t_i and t_a are the trust levels of node i and the accused node, respectively. These can be the default trust levels of the nodes (e.g., police cars are more trustworthy than passenger vehicles) or computed in real-time based on input from other vehicles [21, 31]. m_a is the observed amount of misbehavior of the accused node; more precisely, if we compute how much the observed behavior of the accused node deviates from the normal behavior (e.g., the model of a VANET [44]), m_a is the ratio of the two (observed to normal). The rationale behind this weighting function is to allow highly trustworthy vehicles, such as police cars, to thwart coalitions of untrustworthy attackers while tailoring the accusation to the amount of observed misbehavior to tolerate small errors in the sensory inputs of vehicles. The accused node is locally revoked if the sum of weights exceeds the *revocation threshold* RT:

$$W_a = \sum_{i=1}^{n} w_{i,a} \geq RT \tag{3.2}$$

where n is the number of accusers. Local revocation means that warning messages are transformed into *disregard* messages that instruct all the neighbors of the detected attacker to ignore its messages. The difference between warning and disregard messages is that a specific number of *supporting signatures* is included by the sender in the disregard message, compared

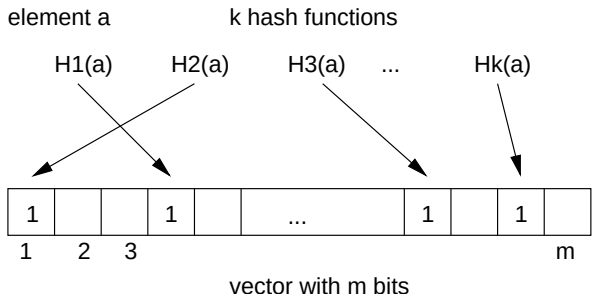


Figure 3.2: Bloom filter concept.

to only one signature in the warning message. This increases the credibility of the message and maximizes channel efficiency by message aggregation. In the next chapter, we describe a vote-based instantiation of the above mechanism.

3.6 Summary

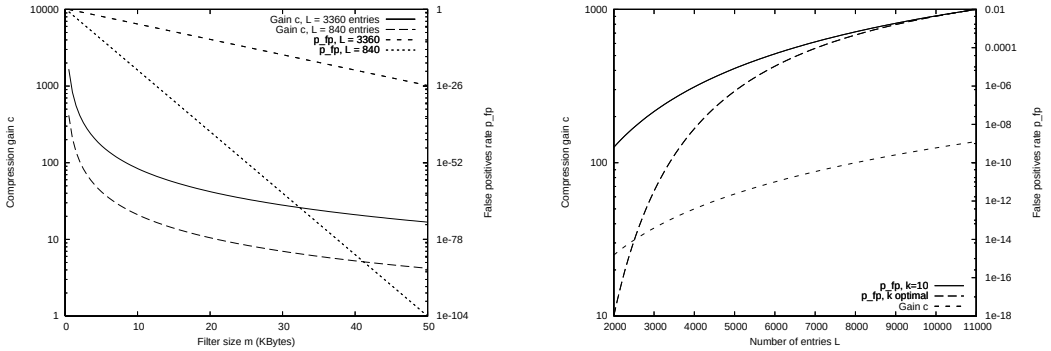
In this chapter, we propose a framework to thwart internal attackers in vehicular networks. The revocation of faulty or attacking nodes is crucial to the robustness of vehicular communication systems. As revocation is the primary means to achieve this, we designed two protocols tailored to the characteristics of the VANET environment. To eliminate the vulnerability window, due to the latency for the authority to identify faulty or misbehaving nodes and distribute revocation information, we designed a scheme that can robustly and efficiently achieve their isolation, as well as contribute to their eventual revocation. This is done with the help of a distributed revocation protocol.

Appendix

3.A Bloom Filters

A Bloom filter, illustrated in Fig. 3.2, consists of a m-bit vector with all its bits initially set to zero. An element (a public key in our context) can be included in the filter by (i) hashing it with k independent hash functions that output numbers in the range $1, \ldots, m$, and (ii) setting to 1 the vector bit each hash function points to. It is possible that one bit is set to 1 multiple times due to the addition of several elements. To check if a given element is contained in the filter, the element is hashed and the corresponding filter bits are checked: If at least one of those bits is zero, the element is not contained in the filter. Otherwise, if all necessary k bits are set, the element is included with high probability. The corresponding bit could have been set also due to multiple additions of other elements. The more elements added, the larger the probability of false positives. In the context of revocation by C^{2}RLs, the nodes validate the certificates included in received messages by checking the Bloom filter.

Bloom filters provide a compression tool with configurable compression gain (c) and false positives rate (p_{fp}). The configurable parameters of the filter are: (i) the filter vector size (m) and (ii) the number of hash functions (k). An additional input, however not configurable, is the number (L) of list entries (certificate IDs), each considered to be l bits long.



(a) Compression gain and false positives rate vs. filter size, using optimal k

(b) Compression gain and false positives rate vs. number of entries L, using k=10 hash functions

Figure 3.3: Bloom filter performance.

A large value of m considerably reduces the false positives rate $p_{fp} = (1-(1-1/m)^{kL})^k \approx (1-e^{-kL/m})^k$, at the cost of decreasing the compression gain $c = L \times l/m$. The choice of m can be derived from Fig. 3.3(a), where the number of hash functions k is chosen to be optimal ($ln(2) \times m/L$, when $dp_{fp}/dk = 0$). Taking also into consideration the range of the number of list entries L, we choose $m = 20$ KBytes, transmittable over the considered radio channels within short time limits (e.g., around 27 ms over a 6 Mbps DRSC channel).

The choice of the optimal number of hash functions k improves the efficiency, at the cost of increased system complexity. In fact, to use a variable number of hash functions, the CA must transmit the used value of k together with the filter. At the receiving side, the verifier must learn k and use k entries of a pre-established list of hash functions.

To avoid this complexity, we use a fixed number of hash functions k. Fig. 3.3(b) shows the case where $k = 10$ (for $m = 20$ KBytes), a compromise between computation complexity and false positives rate. We can see that the resulting false positives rate is reasonably low for small L and converges to the performance provided by optimal values of k when L increases. As for the considerably high compression gain c ($10 < c < 138$), it is independent of the number of hash functions k (whether fixed or variable/optimal).

Chapter 4

Revocation Games in Ephemeral Networks

4.1 Introduction

The introduction of mobile ad hoc networks has brought numerous challenges at every layer of the protocol stack. Like others, security researchers had to solve a new set of problems such as self-organized key management [27] and misbehavior detection [70] in a decentralized, mobile, and infrastructureless environment. As ad hoc networks are typically associated with multihop routing, attacks on the routing and packet forwarding primitives have received particular attention [97]. Reputation systems, with their inherent adaptivity to a constantly changing environment, have been typically proposed as a good cure to many ad hoc networking ills [22].

Since the early days of ad hoc networks, the research landscape has seriously changed. Mostly static sensor networks, with a single owner and limited network and security infrastructure, became a focal topic of research. But packet forwarding remained an important networking component and reputation systems proved again to be effective [40]. Now, the landscape is changing again with the advancement of new types of networks, such as vehicular [96] and delay-tolerant [38] networks. As mentioned in the Introduction of this thesis, we refer to such networks as ephemeral networks due to the shortness of the interactions between the participating wireless devices. In ephemeral networks, mobility makes the monitoring of neighbors' misbehavior infeasible [23]. In addition, the range of misbehavior types has extended beyond routing and packet forwarding to more diverse problems such as malicious data in vehicular networks [44]. Because traditional reputation systems are tightly coupled to a specific misbehavior type (as in packet forwarding), they cannot be merely transposed to these new types of networks. This leads us to the conclusion that new primitives are needed to replace reputation systems, where the latter cannot be efficient anymore.

In this chapter, we advocate a different approach to handling misbehavior. First, the detection system should be decoupled from the reaction system. Second, the reaction should be fast and clear-cut to ensure that misbehaving nodes are denounced to their neighbors and punished despite the changing environment. Where reputation systems cannot again be the magic pill, another primitive - *local revocation* - comes to the rescue. In fact, revocation has long been part of the key management bundle of protocols [27, 28, 66, 99] and was used to cope with node compromise and misbehavior, sometimes similarly to reputation systems [12].

And although both can achieve the same goal of removing attackers from a system for certain misbehavior types, several differences exist between them.

Reputation systems, on one hand, limit the effect of attackers without specifically removing them. This means that attackers with bad reputations are naturally excluded from the network but they can redeem their good reputation by behaving correctly and benefiting from the short memory of their neighbors [21]. This also implies that monitoring nodes have to continuously run a watchdog-like mechanism [68] and keep state of their neighbors for the duration of their mutual interactions. This duration is often too short in ephemeral networks to correctly and timely react to attackers. On the other hand, revocation systems aim to remove attackers from the system and the only way revoked nodes can rejoin the system is by using new identities or credentials. Hence, the duration of the interaction and amount of state corresponding to a given attacker are limited by the time it takes to make a revocation decision.

It is worth noting that revocation typically refers to the annulment of credentials, corresponding to a compromised key, by the key issuer. The key issuer can be a Certificate Authority (CA) in managed systems [101] or the key owner herself in self-organized systems [27]. In this work, we consider revocation as a means for nodes to cope with their misbehaving neighbors, like in reputation systems. This kind of local revocation can be particularly useful when the key issuer is not available to revoke the misbehaving node (e.g., when the CA is offline) or when it is unaware of the misbehavior, especially when the latter does not involve key compromise (e.g., sending bogus information). The node that carries out the local revocation reports its result to the key issuer once the latter is reachable. A node is completely denied participation in the network only when the key issuer revokes it, based on a decision process that is out of the scope of this chapter. This means that local revocation is only temporary and can be repudiated, thus preventing false revocations, due to abusers or errors, from permanently removing benign nodes from the network.

Several papers propose local revocation mechanisms as defined above. There are three major techniques that emerge: (i) voting with a fixed number of votes [28, 66, 99] or a fixed fraction of nodes (e.g., a majority) [12], (ii) key expiration and update [27], (iii) and suicide [71] whereby an accusing node can revoke an accused node by invalidating the credentials of both nodes (the high cost of revocation is meant to deter the abuse of this mechanism).[1] But what is clearly lacking is a unifying framework for comparing the various techniques and consequently defining optimum strategies for given scenarios. For example, the choice of the number of votes varies significantly among different proposals (it is 5 in [66] and sometimes more than 50 in [12]).

An extensive comparison by simulations of a voting scheme based on [12] and a suicide scheme based on [71] was carried out in [72] and gave valuable insights into the strengths and weaknesses of these two techniques in terms of security and networking performance; in addition, a heuristic hybrid protocol was proposed but not evaluated. In this chapter, we make the first attempt to define an analytical framework, based on game theory, that takes into account the economic considerations of individual nodes (actually their owners, as explained below) and prove the conditions under which each strategy (namely, voting or suicide) performs best. Hence, whereas the study in [72] looks at the optimal conditions for using each strategy from a network perspective, our work focuses on the individual preferences of the nodes and then

[1] A similar strategy of expensive punishment was first introduced as a reputation system for networks of mix cascades in [32]; in this case, a single node can declare the failure of its whole mix cascade.

optimizes the network accordingly. In addition, our analytical framework helped us to design a protocol, RevoGame, that outperforms the other strategies.

We chose game theory to be our modeling tool because nodes in ephemeral networks, as defined here, will belong to individuals and hence should represent the rational nature of their owners. As we will further explain in Section 4.2.1, the identities of the nodes participating in revocation are a costly resource and should not be wasted (otherwise, the node owners may have to pay the logistic costs of renewing these identities). Thus, although the mechanism of choice will not be controlled by the device operator (i.e., the user), it will be implemented using a design policy palatable to users. In fact, most users will prefer avoiding the contribution to the system (by revoking attackers) while still benefiting from its services (removing attackers benefits everyone), notably due to the potential costs of the revocation procedure. This situation is famously known as the *free rider problem* in game theory [39]. And as game theory is known to be hard to apply "as is" to networks [67], we complement our theoretical analysis with a protocol that takes practical considerations into account and we back up the resulting design using simulations on an ephemeral network.

Other Related Work

The bounds on the number of voters in local majority voting have been analyzed using graph theory [81]. We do not make any explicit assumptions on the percentage of attackers. In fact, our work tackles the revocation problem from an economic perspective, a direction that is gaining momentum in the security community [11].

Recently, cryptographers started applying game theory to multi-party computation (e.g., secret sharing). A survey of the latest results, in addition to a brief tutorial on game theory, can be found in [56]. A treatment of malicious and selfish behavior in wireless networks can be found in [24].

4.2 System Model

4.2.1 Network Model

We reuse the system model and security architecture in Chapter 2. Hence, as each device changes its pseudonym periodically to avoid being tracked, a node identity is only temporary and can be renewed if the node is revoked. But, given the cost of pseudonym generation (in the logistic and not computational sense) and management by the CA, each vehicle has a limited number of pseudonyms that can quickly become a scarce resource if frequently changed.

4.2.2 Adversary Model

We assume that the adversaries have the same communication capabilities as the mobile nodes. Hence, an adversary possesses the same credentials as any benign node. In this chapter, we consider adversaries that disseminate *false information* in the system. For example, in VANETs this false information can be a warning of the presence of ice on the road, although there is none; this may cause vehicles to make a detour to an ice-free road, thus clearing the main road for an adversary. The adversaries can be misbehaving nodes themselves or compromised benign nodes (e.g., the sensors of a vehicle are not protected from tampering and thus are exposed to attacks). We also assume that adversaries can collude, for example,

by disseminating the same false information in order to increase its credibility. Last but not least, the adversaries can renew their pseudonyms, as any benign device, at the end of a pre-defined time interval (e.g., during the periodic vehicle control) and, if previously revoked, potentially restart their malicious activity.

4.2.3 Detection System

We assume that some of the nodes are equipped with a detection mechanism to identify false information. In VANETs, vehicles could rely on neighborhood information to verify whether there is indeed ice on the road [84]. Let p_d be the average probability with which a benign node detects an attacker. One can also interpret p_d as the fraction of nodes that possess the detection capabilities (e.g., luxury vehicles). In reality, this probability of detection depends on the nature of the false information and hence is specific to each attack. The estimation of p_d can be done in several ways. For example, a bit can be set in packets to indicate the presence of the detection equipment (e.g., a GPS device to correctly detect attacks related to position). p_d can also be the fraction of nodes observing the same event (this can be deduced based on their messages) or it can be the amount of misbehavior. Each node estimates p_d independently, but as explained in Section 4.5.2, our model applies even in the presence of only one detector in the system.

4.3 Revocation Game

In this section, we introduce our game-theoretic model. The key point of the game-theoretic analysis is to consider costs when making a revocation decision. In fact, many security protocols proposed in the literature are often evaluated by their capability to cope with or to completely remove attackers. In addition, the effects of compromise and false positives corresponding to a security protocol are frequently taken into account. In this work, we choose a different metric, namely cost, to design and evaluate a security protocol. After all, if an attack is mild (e.g., the attacker infrequently broadcasts false information), there may be no need to revoke the attacker, given the effort required to carry out the revocation. We also take into account the cost of abuse of the revocation scheme by attackers. In Section 4.3.1, we describe the different revocation strategies that nodes can follow. We introduce the game-theoretic model in Section 4.3.2 and the costs in Section 4.3.3. Section 4.3.4 is a brief overview of the game-theoretic concepts that we use in this chapter.

4.3.1 Revocation Strategies

We consider three revocation strategies for each player (i.e., node) based on the existing protocols. First, player i can *abstain* from the local revocation procedure by playing A (abstain). This strategy assumes that player i (the index i indicates the sequence of play, which is in turn determined by the contention on the wireless channel) is not willing to contribute to the local revocation procedure and instead expects other players or eventually the CA to revoke the attacker. Second, player i can participate in a local *voting* procedure by casting a vote V against a detected attacker [28]. We assume that n votes are required to revoke an attacker locally. The choice of the value of n is a key issue in voting mechanisms and hence we optimize it in Section 4.4.3. Finally, following the protocol suggested in [71],

we allow player i to *self-sacrifice* (denoted by the decision S), i.e., to declare the invalidity of both its current identity (the pseudonym it currently uses) and the identity of the attacker.

4.3.2 Game-Theoretic Model

We model the revocation problem using a finite *dynamic (sequential) game* $\mathbf{G}$ with wireless devices as players. Our choice of dynamic games [39] is based on the sequential nature of the wireless channel access where the action of one player is conditioned by the action of the preceding player (i.e., the second player observes, before making its decision, the action of the first player). We can represent dynamic games by their extensive form, similar to a tree where branches represent the available actions for a given player. Each level of the tree represents a *stage* of the game. We define a *revocation game* for each accused node and we assume that if several nodes are accused then there exist as many revocation games running in parallel. For simplicity and without loss of generality, we consider a single revocation game. Nonetheless, we take the parallel revocation games into account by calculating the collusion cost of the attackers playing revocation games against benign nodes.

There are N benign nodes in one game and M attackers in total; we describe how N and M are estimated in practice in Section 4.5.2. We assume that the number of nodes with detection capabilities is defined by the probability of detection p_d introduced in Section 4.2.3, i.e., there are $p_d N$ detectors. Hence, we consider these detectors as players. Note that we take the non-detecting nodes also into account when calculating the social cost introduced in the next section.

As mentioned earlier, we can represent a dynamic revocation game in an extensive-form tree. We show an example in Fig. 4.1 for $p_d N = 3$ players that want to revoke one attacker, all in power range of each other. We assume that $n = 2$ votes or a single self-sacrifice are enough to locally revoke the attacker. Due to the random nature of the medium access protocol, the sequence of the players' moves is not defined in advance; hence, we refer to the players by their respective order of successful transmission on the shared channel (i.e., index i).

4.3.3 Costs

We represent the costs of the players on the leaves of the extensive-form tree. The cost for any player i has two components: the cost induced by the attack and the cost of participation in revoking the attacker. All costs are represented in terms of *keys* (more precisely, pseudonyms) because they are a scarce resource, as mentioned in Section 4.2.1.

As shown in Fig. 4.1, we first assume that the attack-induced cost is a constant (e.g., the attacker disables, for a limited duration, a road section by disseminating false information about an accident on that section) and denote it by c. The value of this cost depends on the effect of the false information distributed by the attacker, hence we consider various values for it in Section 4.4. Later, we make the assumption that the attack-induced cost is variable and actually increases with time (e.g., the longer the attacker stays in the system, the more road sections it manages to disable by disseminating false information about roadwork on these sections). As for the revocation costs, it is straightforward to assume that abstaining from the game does not cost the players anything. If player i sacrifices itself, then this action implies a cost of 1 key (the player totally loses its ability to use the sacrificed pseudonym). We assume that casting a vote imposes a cost v on any player i; for example, each player may have a voting quota that is decremented each time a player votes. Although voting costs in

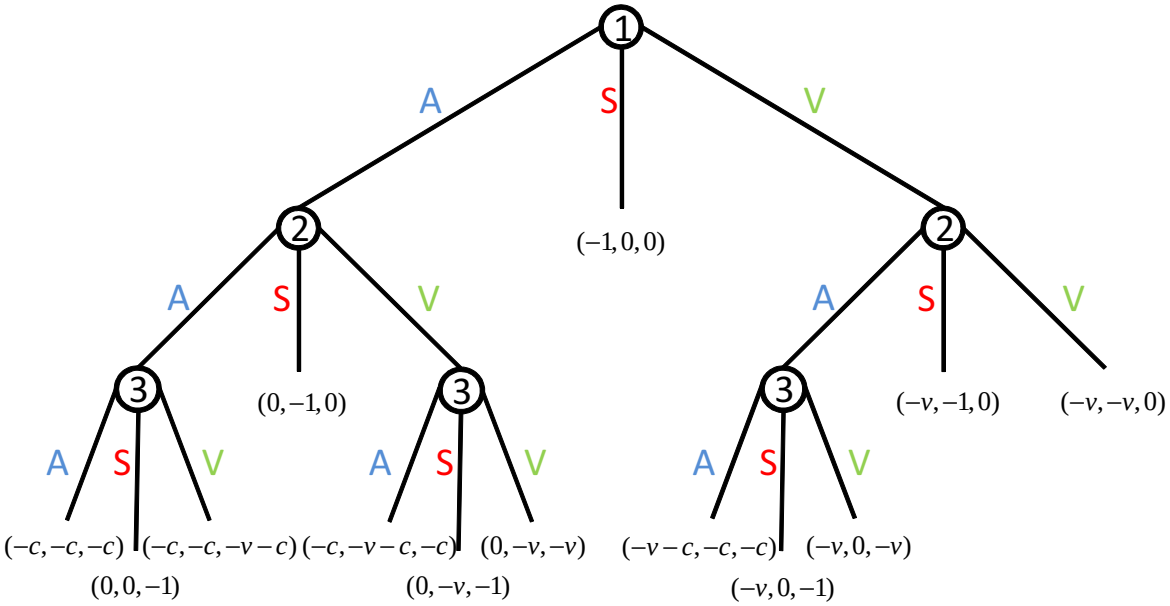


Figure 4.1: The extensive form of the revocation game model when the cost induced by the attack is fixed, i.e., c. The game is represented by a tree and node 1 plays the first action. The game has three stages corresponding to the moves of the three players. The actions (abstain A, self-sacrifice S, and voting V) are represented on each branch of the tree. The leaves of the tree represent the costs of the game for all players. v and 1 are the costs of voting and self-sacrifice, respectively.

terms of computation and communication overhead, we ignore these costs in our model as we assume that nodes (vehicles in the case of VANETs) are powerful enough to justify this assumption. In contrast, the cost of pseudonym management is largely independent of the computational capabilities of the nodes for which the pseudonyms are used. We also make the reasonable assumption that $v < 1$.

For example, in Fig. 4.1, if the three players play (A, V, V), then they successfully revoke the attacker and the second and third players bear the cost of this revocation, resulting in the costs (0, $-v$, $-v$). The objective of the players is to play the strategy that minimizes their individual costs.

In addition, we introduce the notion of the *social cost* C that represents the total cost induced by attackers for all benign nodes in the system; C includes all the costs that are not captured by the sum of individual costs. Hence, given that their individual costs are minimized, nodes should also minimize the social cost, if possible (i.e., as long as minimizing the social cost does not increase the minimized individual costs). C can be expressed by:

$$C = C_{rev} + C_{abuse} + C_{fp} + C_{fn} \tag{4.1}$$

C_{rev} is the cost of revocation games and is equal to the sum of individual costs, C_{abuse} expresses the effect of attackers abusing the revocation system, C_{fp} is the cost of false positives, caused by incorrectly detecting benign nodes as attackers, and C_{fn} quantifies the cost of undetected attackers (false negatives).

4.3.4 Preliminaries

Here, we introduce some basic game-theoretic notions we use to solve the revocation problem. For the sake of simplicity, we only give an intuition for each concept. For the precise mathematical definitions we refer the reader to [39]. A brief tutorial on game theory can also be found in [24].

To predict the outcome of the extensive-form game $\mathbf{G}$, one can use the well-known concept of Nash equilibrium: A strategy profile constitutes a Nash equilibrium if none of the players can increase her payoff by unilaterally changing her strategy. Unfortunately, the Nash equilibrium concept is somewhat limited when it is applied to extensive-form games because it sometimes predicts outcomes that are not *credible* for some players (i.e., these outcomes are unreachable because the players will not play, out of self-interest, according to the incredible Nash equilibrium path). Hence, we use the stronger concept of *subgame-perfect equilibrium*. The strategy profile s is a subgame-perfect equilibrium of a finite extensive-form game $\mathbf{G}$ if it is a Nash equilibrium of any subgame $\mathbf{G}'$ (defined by the appropriate subtree) of the original game $\mathbf{G}$. "Finite game" means that the game has a finite number of stages.

One can check the existence of subgame-perfect equilibria by applying the *one-deviation property*. This property requires that there exists no single stage in the game, in which a player i can gain by deviating from her subgame-perfect equilibrium strategy while conforming to it in other stages. Hence, we can state that strategy profile s is a subgame-perfect equilibrium of a finite extensive-form game $\mathbf{G}$ if the one-deviation property holds.

We will check the existence of subgame-perfect equilibria in the revocation game by the technique of *backward induction* (also called Zermelo's algorithm in dynamic programming). Backward induction works by eliminating sub-optimal actions (i.e., yielding higher costs than the other actions in the same subtree and at the same stage of the game tree), beginning at the leaves of the extensive-form tree. The obtained path (sequence of actions) in the game tree defines the backward induction solution and any strategy profile that realizes this solution is a subgame-perfect equilibrium [39].

4.4 Analysis

In this section, we study two types of revocation games. First, we consider a version where the attack-induced cost is fixed. We show that in this model, the players tend to delegate the revocation decision to the nodes playing in the last stages. This behavior is very risky if the number of players is not precisely known or if attackers can collude to remove some players before they can vote or sacrifice themselves. We then study a game with the assumption of increasing costs. We show that this game gives incentives to the game participants to play in the early stages of the revocation procedure.

4.4.1 Game with Fixed Costs

We assume first that the attacker causes a fixed cost if not revoked and we model this in a *revocation game with fixed costs* $\mathbf{G}^f$. We assume that $\mathbf{G}^f$ is a game of perfect information, meaning that the players know the history of play (in practice, this history is relayed by the players, as we show in Section 4.5.3). Figure 4.1 shows a simple example of $\mathbf{G}^f$. Let us recall that in this example we have $p_d N = 3$ players that want to revoke an attacker. They need $n = 2$ votes or a self-sacrifice to succeed (in Section 4.4.3 we describe how to compute n such

that $n \leq p_d N$). Let $n_i = p_d N - i$ be the number of remaining nodes that can participate in the revocation game after node i that plays in the i^{th} stage of the game. We also assume that n_h is the number of votes that have already been cast (i.e., history of voting). Hence $n_r = n - n_h$ is the number of remaining votes that is required to revoke the attacker by voting. Theorem 4.4.1 identifies the strategies that a player i should follow to achieve a subgame-perfect equilibrium in $\mathbf{G}^f$. The proof is provided in Appendix 4.A.

Theorem 4.4.1. *For any given values of n_i, n_r, v, and c, the strategy of player i that results in a subgame-perfect equilibrium is:*

$$s_i = \begin{cases} A \quad if \quad [c < v] \vee [(c > 1) \wedge (n_i \geq 1)] \\ \qquad\qquad \vee [(v < c < 1) \wedge (n_i \geq n_r)] \\ V \quad if \quad (v < c < 1) \wedge (n_i = n_r - 1) \\ S \quad if \quad (c > 1) \wedge (n_i = 0) \end{cases}$$

Essentially, Theorem 4.4.1 says that as the attacker cost is fixed, the only objective of the players is to remove the attacker, but they do not care in which stage. Thus, the revocation decision is left to the last players, either by voting or by self-sacrifice, whichever induces less cost. For example, a node plays S only if the attack-induced cost is higher than the cost of self-sacrifice and this node is the last player in the game.

The solution in Theorem 4.4.1 is not robust to estimation errors that are due to the high mobility in ephemeral networks. In fact, some of the last players may move out of radio range, thus leaving the game before their turn to play; hence, a revocation decision cannot be reached in this case. To overcome this limitation, we propose a modified version of the revocation game considering variable costs in Section 4.4.2. We also consider these errors when designing our practical revocation protocol in Section 4.5.1 by allowing the players to reassess the game conditions in each stage.

4.4.2 Game with Variable Costs

As explained in Section 4.3.3, the attack-induced cost can increase in each stage where the attacker is not revoked. We model this situation in a revocation game with variable costs $\mathbf{G}^v$. For simplicity, let us assume that the cost at stage j can be represented by $c_j = j \cdot \delta$, where δ is equal to the cost in a single stage. In our model, c_j is a linearly increasing cost, but we can derive similar results for any increasing cost function. Figure 4.2 shows an example of $\mathbf{G}^v$ for three players.

We also assume that the cost after the final stage of the revocation game grows infinitely, i.e., $\lim_{j \to \infty} c_j = \infty$ if the attacker is not revoked. In addition, we assume that $v < \delta$. Figure 4.3 shows the simplified version of this revocation game with the above assumptions. Theorem 4.4.2 identifies the strategy profile that achieves a subgame-perfect equilibrium in $\mathbf{G}^v$. The proof is provided in Appendix 4.B.

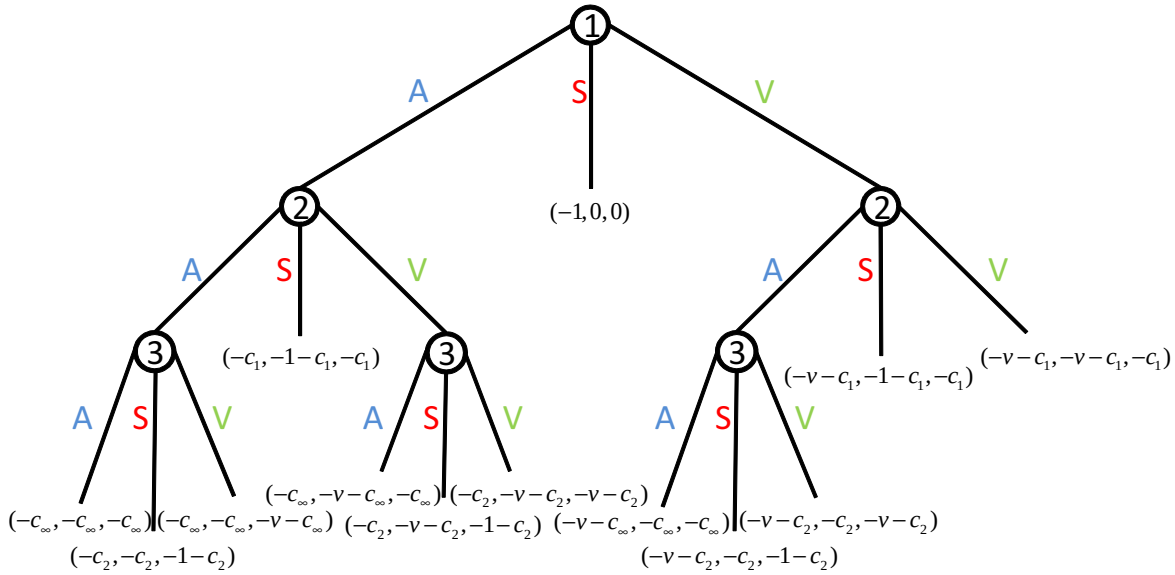


Figure 4.2: Game model with variable costs for three users. Attack-induced costs increase in each stage j: $c_j = j \cdot \delta$.

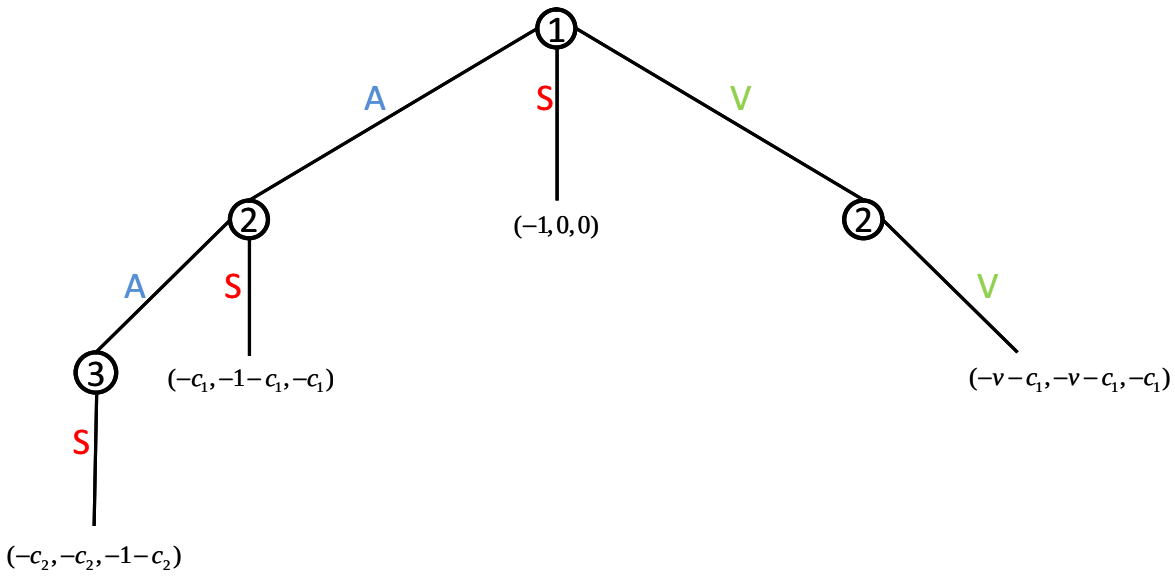


Figure 4.3: Game model with variable costs for three users. Simplified version where $\lim_{j \to \infty} c_j = \infty$ and $v < \delta$. The branches corresponding to the incredible threats have been eliminated, considering the above assumptions.

Theorem 4.4.2. *For any given values of n_i, n_r, v, and δ, the strategy of player i that results in a subgame-perfect equilibrium is:*

$$s_i = \begin{cases} A & \text{if } \begin{array}{l}[(1 \le n_i < \min\{n_r - 1, \frac{1}{\delta}\}) \\ \wedge(v + (n_r - 1)\delta < 1)] \vee [(1 \le n_i < \frac{1}{\delta}) \\ \wedge(v + (n_r - 1)\delta > 1)]\end{array} \\ V & \text{if } (n_i \ge n_r - 1) \wedge (v + (n_r - 1)\delta < 1) \\ S & \text{otherwise} \end{cases}$$

The intuition of Theorem 4.4.2 is that, in contrast to the game with fixed costs, players are more concerned about quickly revoking the attacker because its cost increases with time. Hence, under some conditions, they will begin the revocation process (by voting or self-sacrifice) in the early stages of the game.

4.4.3 Optimal Number of Voters

According to Theorem 4.4.1, if $v < c < 1$ and $n_i = n_r - 1$, revocation takes place by voting in $\mathbf{G}^f$. Similarly, revocation will be performed by voting in $\mathbf{G}^v$ when $n_i \ge n_r - 1$ and $v + (n_r - 1)\delta < 1$. In this section we compute the optimal number of voters n_{opt} by minimizing a simplified version of the social cost C (Eq. 4.1). We derive n_{opt} only for variable cost games (as mentioned in Section 4.4.1 and confirmed by simulations, $\mathbf{G}^f$ cannot cope with high mobility in ephemeral networks). The simplified social cost is:

$$C = n + \frac{M}{n} \tag{4.2}$$

The first term, n, is the amount of time, in stages, during which the attacker can damage the network and it represents the cost of revocation C_{rev}. The second term, $\frac{M}{n}$, represents the cost of abuse C_{abuse} of the revocation scheme by M adversaries (assuming the worst case scenario when M adversaries can revoke at most $\frac{M}{n}$ benign nodes). We do not include C_{fp} and C_{fn} in the equation because their influence on the estimation of n is less significant; instead, we show by simulations their effect on the social cost. Taking into account that the number of votes should not exceed the number of players, the optimal number of votes is:

$$n_{opt} = \min\{p_d N, \sqrt{M}\} \tag{4.3}$$

where $n = \sqrt{M}$ is the value that minimizes C in Eq. 4.2. The estimation of the parameters used in the computation of n_{opt} will be explained in the next section.

4.5 Protocols

In this section, we describe a set of protocols that implement revocation games. Section 4.5.1 introduces the RevoGame protocol that selects strategies according to Theorem 4.4.2. We do not show the protocol for games with fixed costs because it does not work in ephemeral networks due to high mobility. Section 4.5.2 is a detailed evaluation of RevoGame by realistic simulations; it also compares the protocol with the voting and self-sacrifice strategies. Last but not least, section 4.5.3 shows how to cryptographically aggregate the votes of several players in an efficient way.

Protocol 1 RevoGame.

Require: $v < 1$

1: **if** misbehavior detected **then**
2: $\delta \Leftarrow estimate(\delta)$
3: **if** $\delta > 1$ **then**
4: **play** S
5: **else**
6: $p_d \Leftarrow estimate(p_d)$
7: $p_a \Leftarrow estimate(p_a)$
8: $N \Leftarrow estimate(N)$
9: $M \Leftarrow estimate(M)$
10: $n \Leftarrow \min\{p_a p_d N, \sqrt{M}\}$
11: $n_r \Leftarrow n - n_h$
12: $n_i \Leftarrow \lfloor p_a p_d N - i \rfloor$
13: **if** $v + (n_r - 1)\delta < 1$ **then**
14: **if** $n_i \geq n_r - 1$ **then**
15: **if** $n_r = 1$ **then**
16: send revocation message with vote
17: **else**
18: **play** V
19: **else if** $1 \leq n_i < \min\{n_r - 1, \frac{1}{\delta}\}$ **then**
20: **play** A
21: **else**
22: **play** S
23: **else if** $1 \leq n_i < \frac{1}{\delta}$ **then**
24: **play** A
25: **else**
26: **play** S

4.5.1 The RevoGame Protocol

The game-theoretic model presented in Section 4.4 allowed us to determine the optimal strategies and parameters, given some assumptions that should hold during the game. These assumptions are necessary to gain some theoretical insight into the problem, but they have to be carefully assessed when we apply the theoretical results in practical settings. For example, we derived the subgame-perfect equilibria in Section 4.4 by using backward induction. One of the main criticisms of backward induction in the game-theoretic community is that it relies on a long chain of assumptions on other players' actions, especially when the number of players is large. Hence, basing a strategy on the assumption that a given number of following nodes will play their predicted actions implicitly assumes that these nodes will be still in the attacker's neighborhood. But this assumption may not hold in a highly mobile network. In addition, nodes that revoke an attacker stop playing, even if they stay in the attacker's neighborhood, and hence the effective number of players decreases. The solution to these problems is twofold. First, we scale down the number of detectors $p_d N$ by a factor p_a that represents the estimated fraction of *active* players (i.e., nodes that continue playing) among the detectors. Second, the parameters used in the derivation of the optimal strategy should be

re-estimated before each action. The data-centric trust establishment mechanism described in the next chapter helps making these locally estimated parameters consistent by relying on inputs from multiple sources.

The protocol RevoGame integrates the game-theoretic analysis in Section 4.4 with the practical considerations above for $\mathbf{G}^v$. In a nutshell, RevoGame estimates the different parameters, defined in Section 4.4, and checks the play conditions in Theorem 4.4.2. Based on the selected condition, RevoGame plays the corresponding strategy.

4.5.2 Evaluation

To evaluate our theoretical results in a practical context, we simulated the above protocol in an ephemeral network. More specifically, we focused on the re-estimation of n, given the number of attackers and detectors. To simulate the ephemeral network, we use a city scenario with 303 vehicles moving at an average speed of 50 km/h. Each vehicle broadcasts periodic messages every 300 ms over a range of approximately 150 m, conforming to the DSRC specification [4] (i.e., the communication range is the distance the vehicle can cross in 10 sec at its current speed). The scenario, illustrated in Fig. 4.4, has an area of 6.8 km $\times$ 5.5 km in a cartesian coordinate system, is located in Manhattan, and was generated using the VANET simulation framework TraNS [3]; the underlying network simulator is ns-2. The results were averaged over 50 runs with 95 % confidence intervals. All attackers collude against benign nodes, including detectors, trying to revoke as many as possible before being revoked themselves (the attackers also cause damage by disseminating false information). We chose $p_d = 0.8$ (equivalently, 80 % of benign nodes are detectors, as explained in Section 4.2.3) and $\delta = 0.1$ key/message, i.e., the cost of one attacker is 0.1 key for each message sent in the network (in this case, the sequence of messages represents the time scale). Messages sent by attackers are tagged as "bad"; p_d is applied to each received message and thus represents the fact that detection is imperfect. The cost of voting v is set to 0.02. The sequential nature of the game is realized by making each node backoff a random duration between 0 and 20 ms before sending its decision, thus allowing the node to receive messages from players that have accessed the channel earlier.

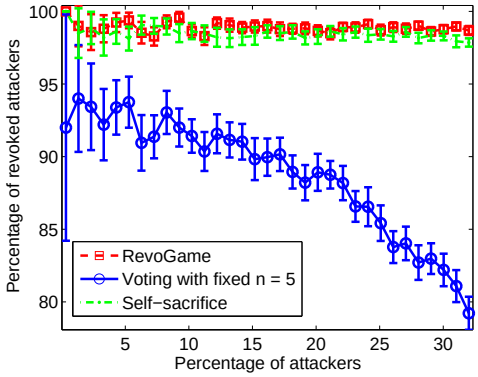
To avoid any consensus-building overhead, each node has to estimate the parameters of the game independently. Hence, a node sets N to be the number of its neighbors that can hear the attacker (this can be estimated knowing the transmission ranges of nodes, in turn derived from their speeds as explained above) at the beginning of each game whereas M is the total number of attackers it has seen so far (worst-case scenario where all attackers collude). After doing a set of initial simulations, we realized that a very important parameter to properly set is p_{fp}, the probability of false positives of the detection mechanism. In our implementation, p_{fp} is the probability of identifying a message received from a benign node as "bad". Relatively high values of p_{fp} (e.g., 10^{-2}, i.e., 1 % of all messages) may result in revoking many benign nodes. Therefore, we selected a much smaller value of p_{fp}, namely 10^{-4}, which can be realized by requiring nodes to receive several "bad" messages before identifying their sender as an attacker. Last but not least, each node estimates the fraction of active players p_a by using its own history of play; more precisely, p_a is the fraction of times the node has participated in revocation games.

When we first tried to simulate the game with fixed costs, we realized that none of the attackers nor the benign nodes was revoked. This can be easily explained by the fact that in the subgame-perfect equilibrium of this game, the revocation was left to the very last players.

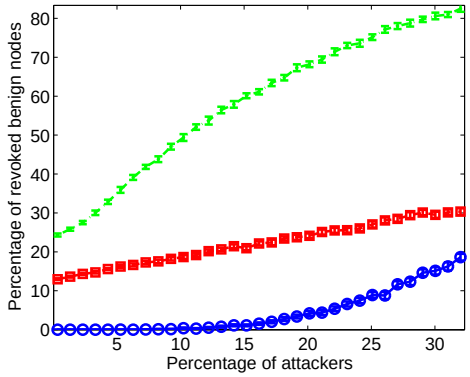
Figure 4.4: The simulation scenario, taken from Manhattan and rendered in Google Earth using TraNS.

In ephemeral networks, the assumption that these players will still be available to play does not hold because nodes move out of range very fast. However, this problem is alleviated in the case of continuously increasing attack-induced costs as nodes play in the early stages of the game with variable costs. Figure 4.5 compares the performance of the RevoGame protocol, self-sacrifice, and voting with a fixed number of votes $n = 5$ (this is the value of n proposed in [66], the only related work where we found a specific small value for n). We can see how the game-theoretic approach adapts to the number of detectors and attackers in the system and thus performs better than the other two protocols. In fact, as this scenario is of medium density, the optimal number of votes is $n_{opt} = 1$ most of the time. Although this value may seem counterintuitive at first (it makes abuse easier), the plots actually show in detail how it surpasses the other options. It is worth noting here that a strategic adversary trying to manipulate RevoGame has little advantage as voting with 1 vote is presumably the strategy that the adversary can exploit best. In addition, $n_{opt} = 1$ shows that the algorithm for estimating n_{opt} is conservative and successfully manages to cope with varying numbers of detectors and attackers.

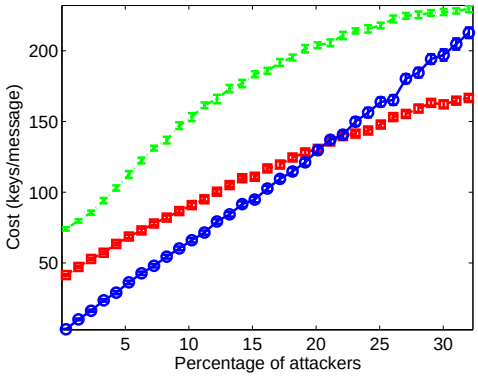
Figure 4.5(a) shows the percentage of revoked attackers. We can see that both RevoGame and self-sacrifice are efficient at revoking attackers, obviously because it is easy to revoke attackers when a revocation decision is taken by a single node; RevoGame slightly outperforms self-sacrifice because fewer detectors sacrifice themselves in the first case. The effect of the three protocols on the percentage of revoked benign nodes can be seen in Fig. 4.5(b) where self-sacrifice removes many more benign nodes, including detectors, than RevoGame. Clearly, voting with a fixed $n = 5$ adapts poorly to a large number of attackers (because there are not enough non-revoked detectors) but avoids the revocation of benign nodes when the percentage of attackers is small. To see whether this is beneficial or detrimental to the system, we plotted



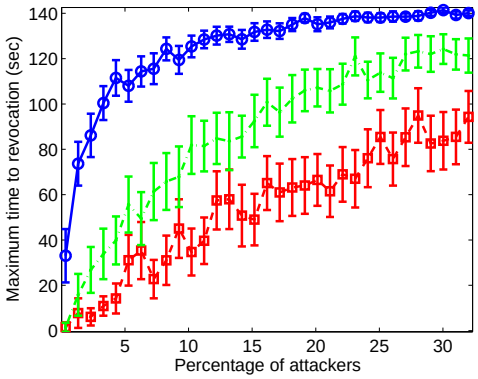
(a) Revoked attackers



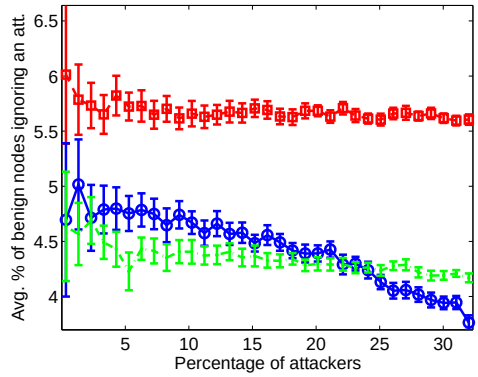
(b) Revoked benign nodes



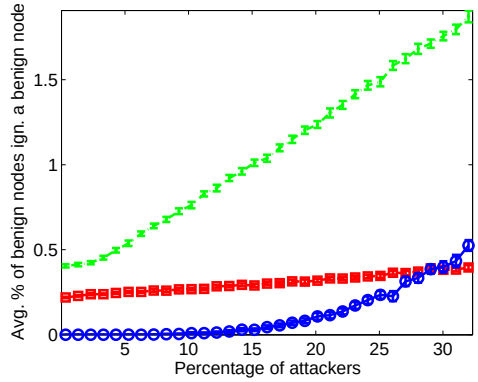
(c) Social cost



(d) Maximum time to revocation



(e) Global effect of local revocations



(f) False positives and abuse

Figure 4.5: Performance of local revocation schemes.

the social cost of each scheme in Fig. 4.5(c) where we can see that self-sacrifice is costlier than the other two schemes. Although RevoGame is costlier than voting for small numbers of attackers, the cost of the conservative voting scheme increases faster with the number of attackers. This is because the costs increase while the attackers remain in the system and voting takes longer to revoke attackers. Figure 4.5(d) shows the maximum delay between the appearance of an attacker in the system and its revocation. RevoGame performs best because it revokes fewer benign nodes than the other two schemes (put differently, it keeps more detectors in the system); as expected, voting is the slowest among the three.

Figures 4.5(e) and 4.5(f) show how the local revocation information is spread in its neighborhood; it reflects the global effect of the local mechanisms. We can see that RevoGame manages to keep the average percentage of benign nodes ignoring an attacker higher than that of the other two protocols because fewer benign nodes are revoked by RevoGame. The same reason explains why a smaller percentage of benign nodes ignore a benign node. Figure 4.5(f) actually gives us an interesting insight into the effect of false positives and attacker abuse of each protocol. Self-sacrifice can result in many benign nodes being ignored by other benign nodes. Voting is obviously the most resilient to false positives and abuse for relatively small percentages of attackers. But as the percentage of attackers increases, the effect of their collusion increases fast, resulting in more benign nodes being ignored. The positive aspect of RevoGame is that it keeps the average percentage of ignored benign nodes relatively stable.

In summary, RevoGame is as efficient, in terms of success rate and speed, as self-sacrifice in removing attackers but revokes fewer benign nodes, thus limiting the effect of abuse of the revocation scheme by attackers. Voting with a fixed number of votes is less efficient and slower than the other two schemes. Voting is also more resilient to the abuse of the revocation scheme when the number of attackers is relatively small, but RevoGame performs better for larger percentages of attackers. These observations justify the need for economic models of security problems, taking into account metrics (e.g., attack-induced costs) that are not captured by merely computing the number of revoked attackers and benign nodes. The protocols, such as RevoGame, built on such a model can be both efficient and adaptive to the actual security costs of a system.

4.5.3 Protocols for Vote Aggregation

In the case of voting, a revocation message should include the signed votes of all the voters that contributed to the revocation decision in order to limit the abuse of the revocation scheme by attackers. A straightforward approach to do this is to concatenate all the votes in the revocation message. But this would inflate both the size and the verification time of the revocation message, which is highly undesirable in ephemeral networks. Fortunately, savings in space, and sometimes time, can be achieved by using a special cryptographic construction - *aggregate signatures* - that allows for the compression of multiple signatures into a single one. Two possible constructions are suitable in our case: general and sequential aggregate signatures. In the following, we describe how these constructions can be applied to vote aggregation and compare them.

General Aggregate Signatures

General aggregate signatures allow any node to aggregate the signatures of other nodes in any order. For the sake of generality, we define the three algorithms GenSign, GenAggSign,

and GenAggVerify to designate individual signature generation, aggregate signature generation and aggregate signature verification, respectively. Although any general aggregate signature scheme can be used, our scheme of choice in this work is the BGLS signature [19] based on the BLS short signatures [20].

Let u_i be the public key of a node i, u_a be the public key of the attacker being revoked, σ_i be the individual signature of i, σ'_i be the aggregate signature so far, n'_i be the optimal number of voters computed by i, T_i the current timestamp of i, $vote_i$ the message (vote) of i, and $vote'_i$ the aggregate vote so far. Let the field $flag_r = 1$ designate that the revocation decision is reached, otherwise $flag_r = 0$. As general aggregate signatures are unordered, a voter can precompute the signature on its vote prior to the reception of the other votes. If the revocation decision is not yet reached, the aggregate-so-far signature on the other votes σ'_{i-1} can be aggregated with σ_i (note here that the index i does not refer to the signing order but to the transmission order). If the revocation decision is reached, σ_i has to be recomputed with $flag_r$ set to 1. In addition to aggregating the votes, the certificates $cert_i$ of the public keys u_i can be aggregated as well, using the same primitive. In this case, "certificate" specifically means the signature of the CA over u_i, hence certificates can be aggregated with the signatures on votes. We assume that nodes know the public key of the CA and thus are able to verify its certificates. The protocol GenVote illustrates the vote aggregation process.

Protocol 2 GenVote

Ensure: $flag_r = 0$

1: $vote_i \Leftarrow u_a \| flag_r \| u_i \| n'_i \| T_i$
2: $\sigma_i \Leftarrow \mathsf{GenSign}(vote_i)$ // precompute the signature
3: **if** votes received **then**
4: **if** decision = revocation **then**
5: $flag_r \Leftarrow 1$
6: $vote_i \Leftarrow u_a \| flag_r \| u_i \| n'_i \| T_i$
7: $\sigma_i \Leftarrow \mathsf{GenSign}(vote_i)$ // recompute the sig.
8: $\sigma'_i \Leftarrow \mathsf{GenAggSign}(\sigma_i, cert_i)$
9: $vote'_i \Leftarrow u_a \| flag_r \| v_1 \| n'_1 \| t_1 \| \ldots \| u_i \| n'_i \| T_i \| \sigma'_i$

GenVoteVerify illustrates the vote verification process. This is done only if $flag_r = 1$, i.e., a revocation decision has been reached. First, the votes need to be reconstructed and then the aggregate signature is verified.

Protocol 3 GenVoteVerify($vote'_k$)

1: $u_a, flag_r, \sigma'_k \Leftarrow extract(vote'_k)$
Ensure: $flag_r = 1$
2: **for** $i = 1$ to k **do**
3: $u_i, n'_i, T_i \Leftarrow extract(vote'_k)$
4: **if** $i < k$ **then**
5: $vote_i \Leftarrow u_a \| 0 \| u_i \| n'_i \| T_i$ // reconstruct the votes
6: **else**
7: $vote_k \Leftarrow u_a \| 1 \| v_k \| n'_k \| t_k$
8: GenAggVerify($vote_1, \ldots, vote_k, v_1, \ldots, v_k, \sigma'_k$)

Sequential Aggregate Signatures

Although general aggregate signatures sometimes allow for saving time on signature computation, they require both the public key and the certificate of a voter to be sent over the air. A common solution to this problem is to use identity-based signatures (IBS) where the identity of a node serves as its public key and its capability to sign plays the role of the certificate in the previous scheme. The only possible, so far, aggregate constructions using IBS are sequential aggregate signatures where signatures are generated in a given order and aggregation and signing are the same operation [17]. This means that it is not possible to precompute individual signatures. The advantage of identity-based signatures is that no certificates need to be added to the signatures. Given that it is easier to optimize the computation time, as explained in the next section, rather than the message overhead, sequential aggregate signatures based on IBS can be a viable option in our scenario.

We define IBSeqSign and IBSeqVerify to designate the aggregate signature generation and verification algorithms. Let ID_i be the identity of node i and ID_a be the identity of the attacker. We assume that the aggregate signature construction is based on the IBSAS scheme recently proposed in [17]. The protocol IBSeqVote illustrates the vote aggregation process. The same definitions as in the previous section apply.

Protocol 4 IBSeqVote

Ensure: $flag_r = 0$
1: **if** $vote'_{i-1}$ received **then**
2: **for** $j = 1$ to $i - 1$ **do**
3: $ID_j, n'_j, t_j \Leftarrow extract(vote'_{i-1})$
4: **if** decision = revocation **then**
5: $flag_r \Leftarrow 1$
6: $vote_i \Leftarrow ID_a \| flag_r \| ID_1 \| n'_1 \| t_1 \| \ldots \| ID_i \| n'_i \| T_i$
7: $\sigma_i \Leftarrow \mathsf{IBSeqSign}(vote_i)$
8: $vote'_i \Leftarrow vote_i \| \sigma_i$

We omit the description of IBSeqVerify as it is similar to GenVoteVerify.

Comparison

A general aggregate signature can be as short as 171 bits (for 1024-bit security) if the individual signatures are BLS short signatures [20], but requires the certificates of the public keys of all signers. Although certificates can be aggregated as described in the previous section, this may not always be possible (e.g., if certificates are not based on the BLS short signatures). An additional drawback of the general aggregate signatures based on the BGLS scheme is that the verification time requires a linear, in the number of voters, number of pairing computations [19], an expensive cryptographic primitive.

The identity-based aggregate signatures are longer than the BLS signatures but require no certificates; they still require the transmission of the identities of all the signers. An additional advantage of the IBSAS scheme is that verification of an aggregate signature requires three pairing computations independently of the number of signers [17].

In summary, in terms of communication overhead, general aggregate signatures are comparable to sequential aggregate signatures if the certificates in the first scheme are aggregated;

otherwise, the latter scheme generates shorter aggregate signatures. In terms of computation overhead, general aggregate signatures are faster to generate (especially with precomputations) but slower to verify, whereas sequential aggregate signatures are slower to generate but faster to verify. As a last note on the efficiency of pairing computations, recent results have shown the feasibility of pairings on resource-constrained devices such as smart cards [85]. Of course, the platform of choice plays a crucial role in the execution speed, but it is reasonable to expect that efficient bilinear pairings will soon be a common and viable cryptographic primitive [18].

4.6 Summary

New types of networks usually require new security mechanisms. Ephemeral networks are an example where reputation systems may not perform well. In this chapter, we have studied the applicability of local revocation mechanisms to handle misbehavior in these networks. Using a game-theoretic model, we have derived the optimal strategies and parameters for different combinations of detection capability and attacker penetration and impact. In addition, we have designed a protocol, RevoGame, based on the game-theoretic analysis and practical considerations. Realistic simulation results in vehicular networks show that the game-theoretic approach achieves the elusive tradeoff between the approaches found in the literature.

Appendix

4.A Proof of Theorem 4.1

We use the one-stage-deviation principle to prove that deviating from each of the strategies in Theorem 4.4.1 under the corresponding conditions will not result in a gain.

Let us assume that $c < v$, i.e., voting is more expensive than enduring the attack-induced cost. If at any stage, player i deviates from the strategy A, playing V or S would result in a cost of v or 1, respectively. In both cases, the cost is bigger than c (assuming that $v < 1$ as mentioned in Section 4.3.3). Figure 4.6 illustrates this case.

If $v < c < 1$ and $n_i \geq n_r$, i.e., voting is less expensive than the attack-induced cost and the number of remaining detectors is bigger than the required number of voters, then playing S or V would result in a cost of 1 or v, respectively. These costs are greater than 0 (the attacker will be revoked anyway because $v < c$). Hence, i cannot gain by deviating from the action A. This is shown in Fig. 4.7 for $p_d N = 3$.

Another case that makes action A the best response is when the attack-induced cost c is bigger than 1, the cost of self-sacrifice, and the number of remaining players is bigger than 1 ($n_i \geq 1$), i.e., the attacker will be revoked by another player anyway. The proof is similar to the previous cases and is illustrated in Fig. 4.8.

Let us now assume that $v < c < 1$, $n_i = n_r - 1$, and player i that is supposed to play V according to strategy s_i above, deviates in a single stage. If it plays S or A, it loses 1 or c, respectively, both bigger than v. In both cases, i cannot gain by deviating from V.

Finally, if $c > 1$ and $n_i = 0$, if player i deviates from S by playing A or V, the player's cost will be c or $c + v$, respectively. Both costs are greater than 1 and deviation results in a loss.

Based on the above cases, deviation from any action under the corresponding conditions results in a loss for the deviating player and hence strategy s_i leads to a subgame-perfect equilibrium. □

4.B Proof of Theorem 4.2

We will prove this theorem, as before, by showing that the actions in the theorem are the players' best responses under the corresponding conditions.

If $1 \leq n_i < \min\{n_r - 1, \frac{1}{\delta}\}$ (i.e., there are not enough voters but at least another player can self-sacrifice) and $v + (n_r - 1)\delta < 1$ (i.e., self-sacrifice is more expensive than voting), deviating from playing A will cause player i a cost of 1, because playing S is the only possible option (the number of voters is insufficient). Hence, i does not gain by deviation. This is shown in the subtree of player 2 on the left in Fig. 4.9.

If $1 \leq n_i < \frac{1}{\delta}$ and $v + (n_r - 1)\delta > 1$, deviating from playing A will cause player i a cost of 1 if it plays S and a cost of $v + (n_r - 1)\delta$ if it plays V. Hence, i does not gain by deviation from A. This is illustrated in the subtree of player 2 on the left in Fig. 4.10.

If $n_i \geq n_r - 1$ and $v + (n_r - 1)\delta < 1$, i.e., there are enough voters and voting is less expensive than self-sacrifice, and player i deviates from playing V by playing A or S, its cost will be $n_i\delta$ or 1, respectively. In both cases the cost will be greater than $v + (n_r - 1)\delta$, assuming v is negligible. Hence, the player does not gain by one-stage deviation. This is shown in Fig. 4.9.

The expanded condition for playing S is: $[\delta > 1] \vee [(n_i < n_r - 1) \wedge ((n_i = 0) \vee (n_i \geq \frac{1}{\delta})) \wedge (v + (n_r - 1)\delta < 1)] \vee [((n_i = 0) \vee (n_i \geq \frac{1}{\delta})) \wedge (\delta < 1 < v + (n_r - 1)\delta)]$. If $\delta > 1$ (the attack-induced cost is more expensive than self-sacrifice) and player i deviates from strategy S, it can play V or A. If it plays V, it loses $v + (n_r - 1)\delta > 1$ and if it plays A, it loses $n_i\delta > 1$ as shown in Fig. 4.11.

If $n_i = 0$ (the current player is the last one) or $n_i \geq \frac{1}{\delta}$ (alternately $n_i\delta \geq 1$, which means that the cost of abstaining is higher than the cost of self-sacrifice), and $v + (n_r - 1)\delta < 1$ and $n_i < n_r - 1$ (i.e., voting is cheaper than self-sacrifice but there are not enough voters), deviating from S by playing A would result in a cost of $n_i\delta$ if $n_i \geq \frac{1}{\delta}$ or ∞ if $n_i = 0$ (the attacker will not be revoked). Playing V would not lead to revocation (as there are not enough voters) and hence result in a cost of $v + n_i\delta$ if $n_i \geq \frac{1}{\delta}$ or ∞ if $n_i = 0$. Hence, deviation from S would not pay off. This is shown in the subtree of player 3 in Fig. 4.9. A similar proof can be made if $n_i = 0$ or $n_i \geq \frac{1}{\delta}$, and $\delta < 1 < v + (n_r - 1)\delta$: both voting and abstaining are more expensive than self-sacrifice and hence deviation from S would increase the cost (in other words, lower the payoff). This can be seen in the subtree of player 3 in Fig. 4.10. Based on the above, one-stage-deviation from the strategy s_i degrades the deviating player's payoff and hence s_i leads to a subgame-perfect equilibrium. □

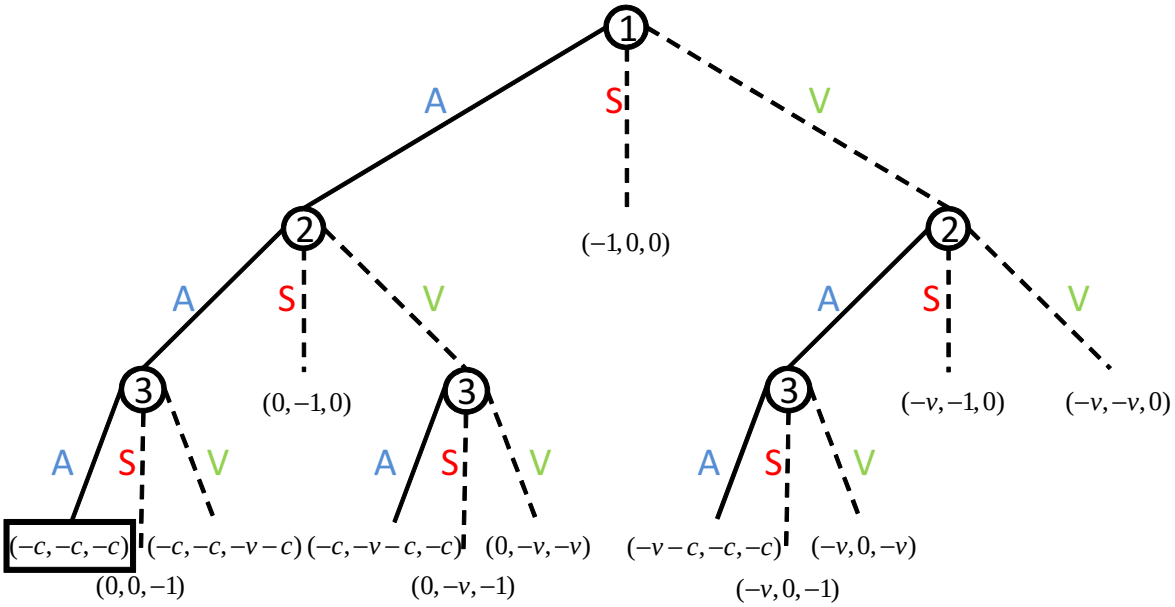


Figure 4.6: Extensive form of $\mathbf{G}^f$ when $c < v$. Action A for all players is the best response. The thick lines show the best response of each player in every stage of the game. This can be obtained by comparing the resulting costs for all strategies of a player moving in a given stage. The continuous thick line represents the subgame-perfect equilibrium.

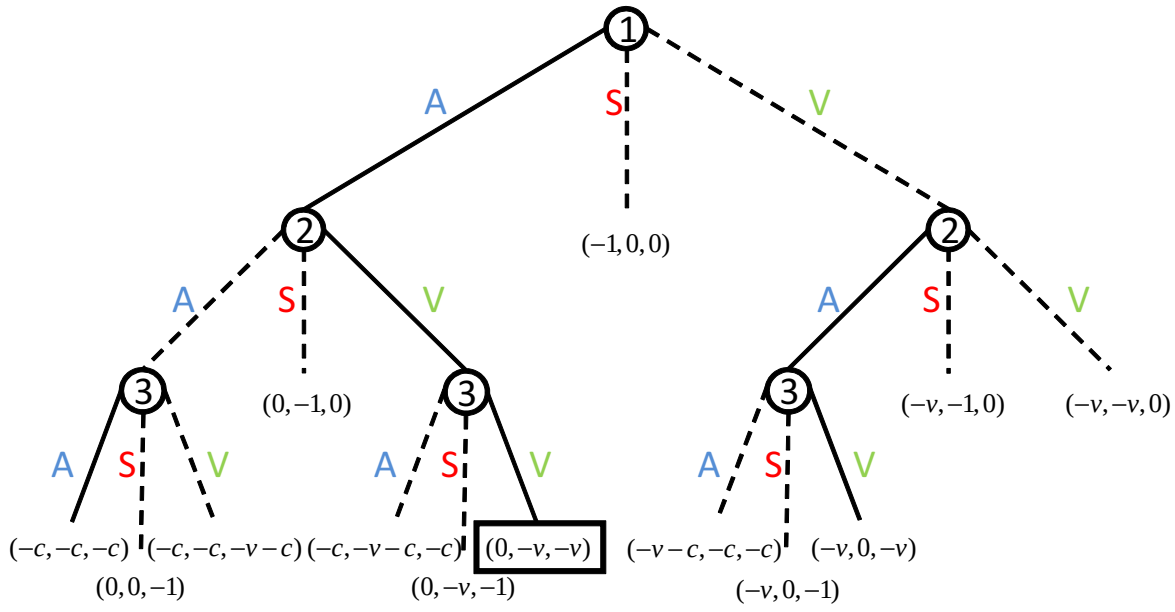


Figure 4.7: Extensive form of $\mathbf{G}^f$ when $v < c < 1$. Action A for the first player is the best response since $n_i \geq n_r$, whereas the best response for nodes 2 and 3 is V since $n_i < n_r$. Hence, the subgame-perfect equilibrium game is achieved if player 1 plays A, 2 plays V, and 3 plays V.

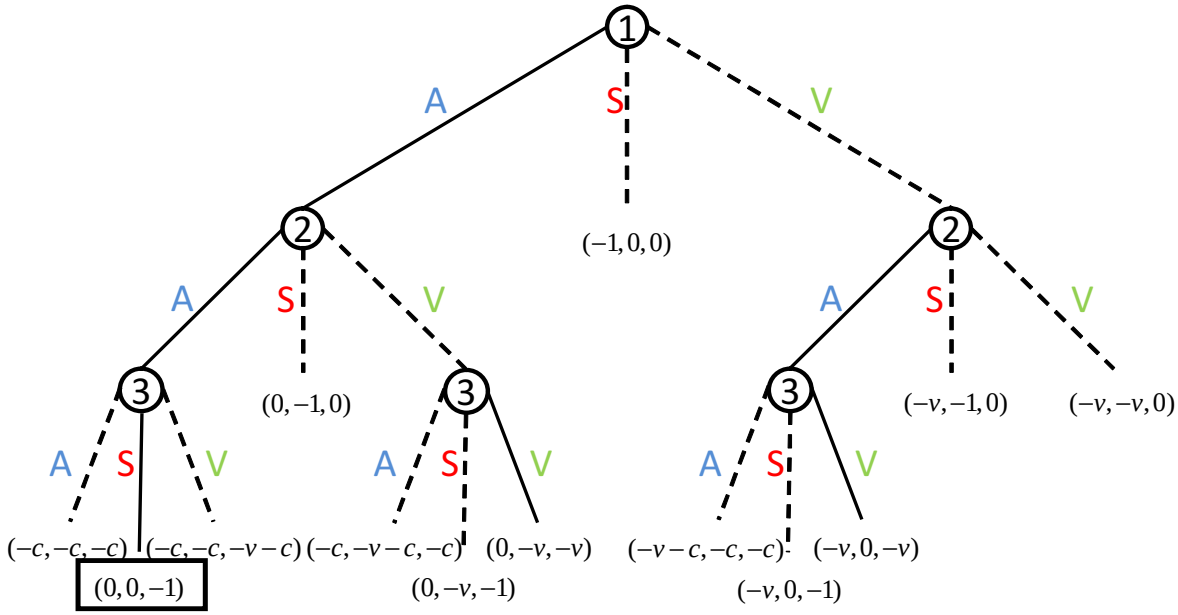


Figure 4.8: Extensive form of $\mathbf{G}^f$ when $c > 1$. Action A is the best response of the first and second players because $n_1 > n_2 \geq 1$, whereas player 3 plays S because $n_3 = 0$.

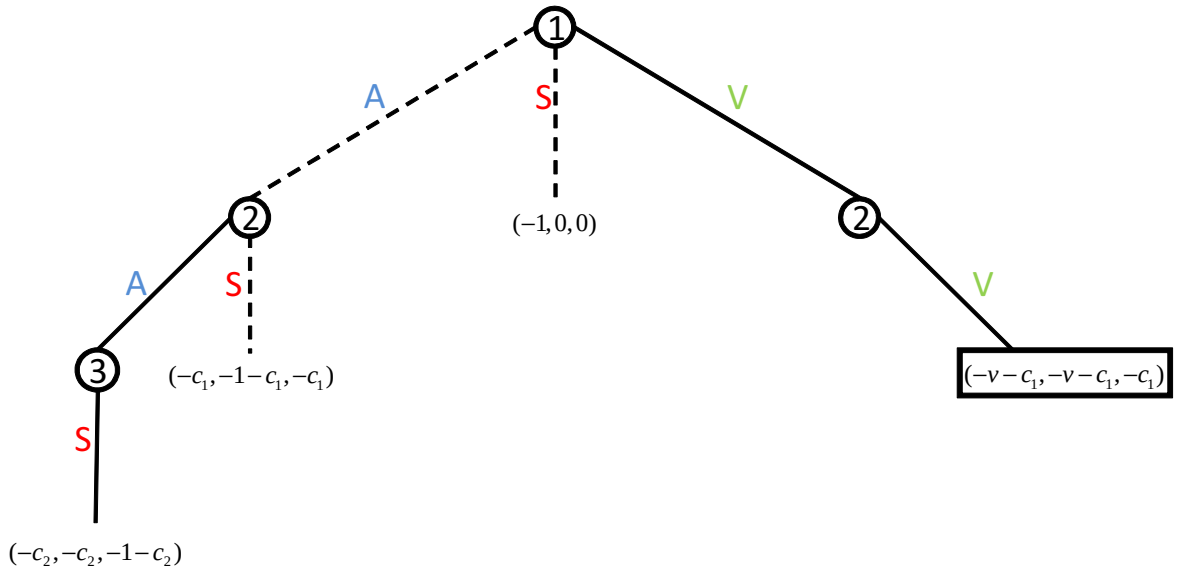


Figure 4.9: Extensive form of $\mathbf{G}^v$ when $v + (n_r - 1)\delta < 1$. The subgame perfect equilibrium is achieved when players 1 and 2 play V.

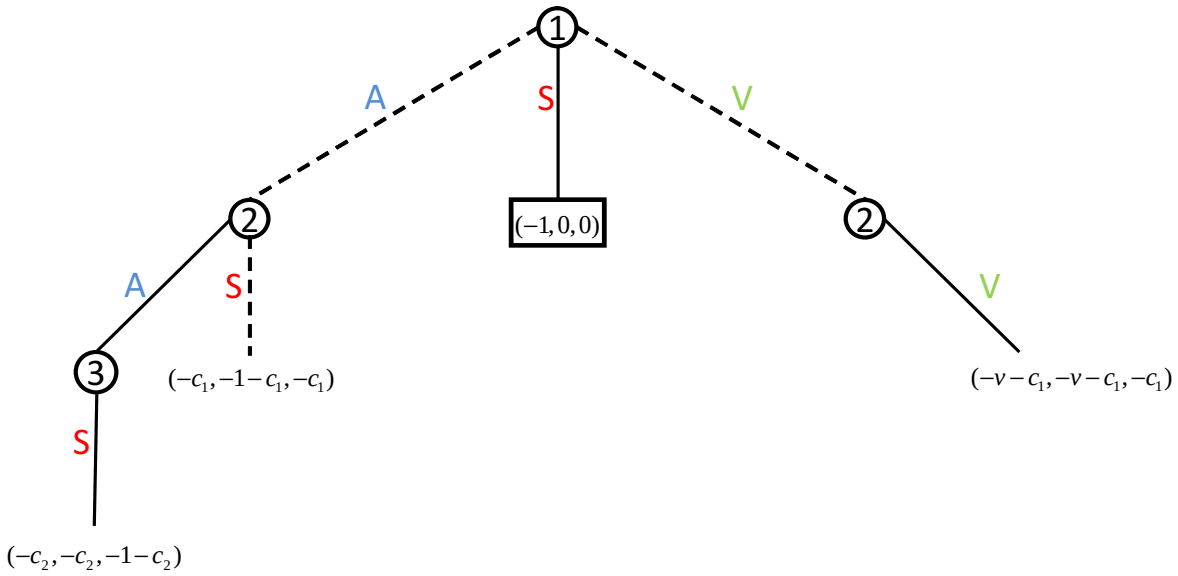


Figure 4.10: Extensive form of $\mathbf{G}^v$ when $v + (n_r - 1)\delta > 1 > \delta$. The subgame-perfect equilibrium is achieved when player 1 plays S.

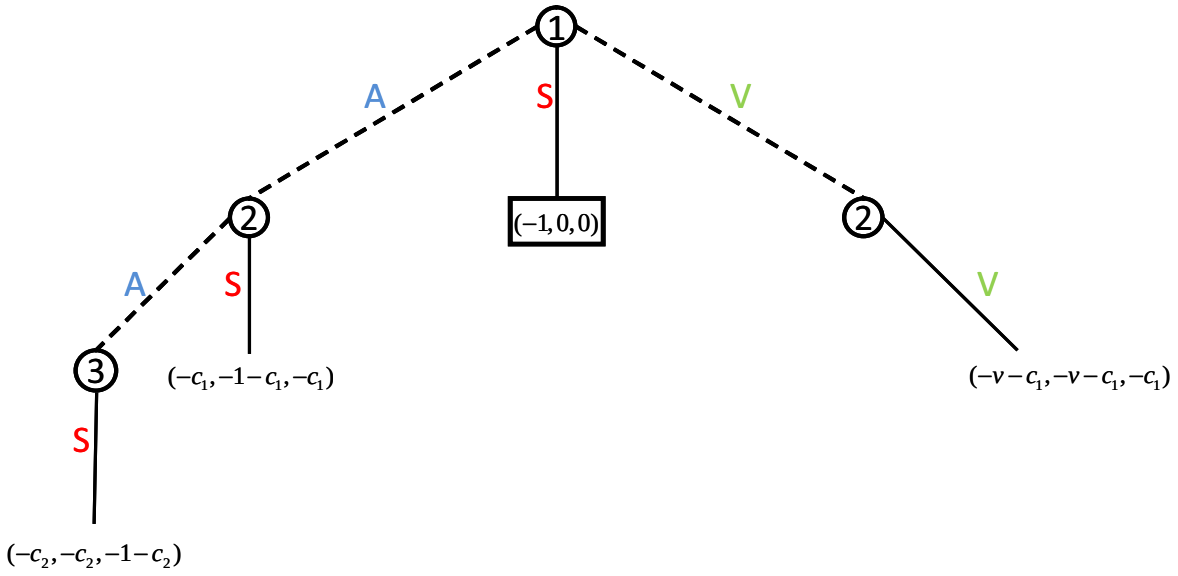


Figure 4.11: Extensive form of $\mathbf{G}^v$ when $\delta > 1$. The subgame-perfect equilibrium is achieved when player 1 plays S.

Chapter 5

On Data-Centric Trust Establishment in Ephemeral Networks

5.1 Introduction

In all traditional notions of trust, data trust (e.g., trust in the identity or access/attribute certificates) was based exclusively on a priori trust relations established with the network entities producing these data (e.g., certification authorities, network nodes) [42, 58, 61]. This was also the case when trust was derived via fairly lengthy interactions among nodes, as in reputation systems [21, 40, 73, 103]. Moreover, any new data trust relationships that needed to be established required only trust in the entity that produced those data. All trust establishment logics proposed to date have been based on entities (e.g., "principals" such as nodes) making statements on data [21, 36, 42, 53, 58, 61, 91, 92]. Furthermore, traditional trust relations evolved generally slowly with time: once established, they lasted a long time and changed only after fairly lengthy operations (e.g., certificate revocation or monitoring and then voting-off of peers).

These observations indicate that existing trust notions are *entity-centric* and *slow to change*. However, several emerging mobile networking systems are heavily, if not entirely, *data-centric* in their functionality and operate in *ephemeral* environments. In such scenarios, it is more useful to establish trust in data rather than in the nodes reporting them. For example, in VANET, node identities are largely irrelevant; rather, safety warnings and traffic information updates, along with their time freshness and location relevance, are valuable. At the same time, interactions with data reporters do not rely on any prior association, and encounters are often short-lived, especially due to high mobility.

In such scenarios, and unlike in traditional trust establishment schemes, the trust level associated with data is not the same as that of the node that generated the data. More specifically, in the VANET example, vehicles will have different preset node trust levels (e.g., police cars are more trustworthy than private vehicles), but (i) different events reported by the same vehicle may have different levels of trust (due to distance to the event, timeliness of the report, vehicle equipment level) that may differ from that of the vehicle itself; (ii) the same event reported by multiple vehicles with different preset node trust levels has to be associated with a single trust level that would, of course, differ from some of the levels of the reporting vehicles; and (iii) an event reported by a vehicle requires corroboration by other

vehicles and hence its level of trust would differ from that of the reporting vehicle. In other words, the following question arises naturally: how can these emerging systems be effective and trustworthy when their basic operational requirements are not satisfied by existing trust notions? To address this challenge, we advocate a clean-slate approach. We propose data-centric trust establishment: *data trustworthiness should be attributed primarily to data per se, rather than being merely a reflection of the trust attributed to data-reporting entities.*

The logic we propose extends the traditional notions of trust and methods of trust establishment in several ways. First, unlike traditional trust, a priori trust relationships in entities (nodes) represent only one of the default parameters for establishing data trust. For example, our logic, while using nodes' statements on data, does not rely exclusively on such statements. Instead, it takes into account dynamic factors, such as location and time, as well as the number and type of the statements on data, to derive data trust relations. Second, beyond the traditional time-invariant or slow-evolving trust notions, data-centric trust relations are by definition ephemeral and have to be established and re-established frequently, based on network and perceived environment changes. For example, an event report (e.g., accident report, weather report) that must be believed by recipient nodes in real-time cannot last longer than the lifetime of the event or of the network that generated this event. Multiple rounds of node interactions are typically not possible in such networks. Third, trust does not stem from a single source of data (e.g., a certification authority) and generally it is application-dependent (in contrast to entity-centric trust when, for example, multiple applications use certificates for their access control and authentication policies).

We derive trust in data (e.g., reported event) from multiple pieces of evidence (e.g., reports from multiple vehicles). Then, our logic weighs each individual piece of evidence according to well-established rules and takes into account various trust metrics, such as time freshness and location relevance, defined specifically in the context of an application. Then, data and their respective weights serve as inputs to a decision logic that outputs the level of trust in these data. We evaluate several techniques, including voting, Bayesian inference, and the Dempster-Shafer Theory of evidence. Notably, Bayesian inference takes into account prior knowledge, whereas the Dempster-Shafer Theory accounts for the *uncertainty* about data. More specifically, while trust establishment mechanisms based on popular decision logics, such as voting and Bayesian inference, consider uncertainty as refutal of evidence, our framework considers uncertainty as either supporting or refuting evidence, thus making the decision process more realistic. We show in this chapter that this distinction affects the flexibility and resilience to attackers in some scenarios.

5.2 Related Work

Work on trust has produced rich literature in conventional, P2P and ad hoc networks. In the latter, most contributions assume that there is no infrastructure and no PKI; trust is a relation among entities; trust is based on observations, with a history of interactions needed to establish trust. To the best of our knowledge, the computation of trust values in the context of ad hoc networks has been considered in only two cases: certification [36, 92] and routing [21, 103]. Otherwise, trust evaluation assumes the prior establishment of trust relations. In both certification and routing, trust values are established in very specific ways that cannot be generalized to other approaches. Eschenauer et. al. [36] introduce the general principles of trust establishment in mobile ad hoc networks and compare them to those in the Internet.

They describe examples of generic evidence generation and distribution in a node-centric authentication process.

Several papers [21, 40, 73, 103] describe the use of modified Bayesian approaches to build reputations systems with secondhand information to establish trust in routing protocols. As mentioned throughout the chapter, reputation systems monitor node actions over several interactions to compute node trust values. In contrast, data trust, as defined in this work, focuses on evaluating data rather than nodes and based on only one message per node (to cope with the ephemerality of the network). In addition, all of these works relied on BI to compute reputation scores, whereas we showed that DST is more resilient to attacks when there is high uncertainty in the network (Section 5.6.3). Data trust can actually complement reputation systems in non-ephemeral networks.

The main approach advanced by Jiang and Baras [53] is based on local voting that is a weighted sum of votes. Conflicting votes are mitigated by each other when summed. They also favor local interactions that we use as well.

The main idea behind the work by Sun et. al. [91] is that trust represents uncertainty that in turn can be computed using entropy. They also introduce the notion of *confidence of belief* to differentiate between long-term and short-term trust. Trust can be established through direct observations or through a third party by recommendations.

Theodorakopoulos and Baras [92] assume the transitivity of trust to establish a relation between two entities without previous interactions. In this context, they model trust evaluation as a path problem on a directed graph. Routing protocols are the main target of this approach.

More closely related to VANETs and thus the case-study instantiation of our framework, Ostermaier et al. [75] analyze the performance of voting schemes for local danger warnings in VANETs. As mentioned earlier, voting schemes cannot properly express composite events. Another paper by Golle et al. [44] proposes a framework for data validation in VANETs; it consists in comparing received data to a model of the VANET and accept their validity if both agree. Building and updating such a model in real-time may not satisfy the requirement of fast data processing in VANETs. Klein [57] describes the application of several data fusion techniques at the traffic management center.

There is little work on applying the Dempster-Shafer Theory to ad hoc networks, the most relevant to our work is the paper by Chen and Venkataramanan [29] that describes how DST can be applied to distributed intrusion detection in ad hoc networks. Siaterlis and Maglaris [90] apply DST to DoS anomaly detection. The notion of belief, disbelief, and uncertainty appears in the work of Jøsang [54]. The paper describes a certification algebra based on a framework for artificial reasoning called *Subjective Logic*.

5.3 General Framework

5.3.1 Preliminaries

We consider systems with an authority responsible for assigning identities and credentials to all system entities that we denote as *nodes*. All legitimate nodes are equipped with credentials (e.g., certified public keys) that the authority can revoke. Specific to the system and applications, we define a set $\Omega = \{\alpha_1, \alpha_2, \ldots, \alpha_I\}$ of mutually exclusive *basic events*. Composite events γ are unions or intersections of multiple basic events. Examples of basic events are "ice on the road" and "traffic jam". If the ice on the road causes a traffic jam, this becomes

the composite event "ice on the road and traffic jam ahead". Each α_i is a perceivable event generated by the environment, network, or an application running on vehicles. There may be multiple applications, each having its own set of relevant events. These sets are overlapping, as their events belong to the pool of basic events.

We consider $\mathcal{V}$, the set of nodes u_k, classified according to a system-specific set of node types, $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$. We define a function $\tau : \mathcal{V} \rightarrow \Theta$ returning the type of node u_k. *Reports* are statements by nodes on events, including related time and geographic coordinates where applicable. For simplicity, we consider reports on basic events, as reports on composite events are straightforward. We do not dwell on the exact method for report generation, as this is specific to the application.

5.3.2 Default Trustworthiness

We define the *default trustworthiness* of a node u_k of type θ_n as a real value that depends on the attributes related to the designated type of node u_k. For all node types, there exists a trustworthiness ranking $0 < \theta_1 < \theta_2 < \ldots < \theta_N < 1$. For example, some nodes are better protected from attacks, more closely monitored and frequently re-enforced, and, overall, more adequately equipped, e.g., with reliable components. As they are less likely to exhibit faulty behavior, they are considered more trustworthy.

We stress here that the data-centric trust establishment framework does not aim to replace or amend source authentication, as in reputation systems, but uses it as an input to the data trust evaluation function. In fact, if a node reputation system were in place, its output scores could also be used as input to the data trust function. Hence, data trust builds on the information provided by source authentication and reputation systems without trying to supplant them. The choice of the entity trust establishment system is orthogonal to the scope of this chapter and has been prolifically addressed in the literature (Section 5.2).

5.3.3 Event- or Task-Specific Trustworthiness

Nodes in general perform multiple tasks that are system-, node- and protocol-specific actions. Let Λ be the set of all relevant system tasks. Then for some nodes u_1 and u_2 with types $\tau(u_1) = \theta_1$ and $\tau(u_2) = \theta_2$ and default trustworthiness rankings $t_{\theta_1} < t_{\theta_2}$, it is possible that u_1 is more trustworthy than u_2 with respect to a task $\lambda \in \Lambda$.

Reporting data on events is clearly one of the node tasks. For the sake of simplicity, we talk here about event-specific trustworthiness implying that it is actually task-specific trustworthiness. Nevertheless, the two can be easily distinguished, when necessary; e.g., when tasks include any other protocol-specific action such as communication.

With the above considerations in mind, we define the event-specific *trustworthiness* function $f : \Theta \times \Lambda \rightarrow [0, 1]$. f has two arguments: the type $\tau(u_k)$ of the reporting node u_k and the task λ_j. f does differentiate among any two or more nodes of the same type, and if $\lambda_j = \emptyset$ (no specific event or task), f is the default trustworthiness $f = t_{\tau(u_k)}$.

5.3.4 Dynamic Trustworthiness Factors

The ability to dynamically update trustworthiness can be valuable, especially for capturing the intricacies of a mobile ad hoc networking environment. For example, nodes can become faulty or compromised by attackers and hence need to be revoked. In addition, the location

and time of report generation change fast and are important in assigning trustworthiness values to events.

To capture this, we define a *security status* function $s : \mathcal{V} \to [0, 1]$. $s(u_k) = 0$ implies node u_k is revoked, and $s(u_k) = 1$ implies that the node is legitimate. Intermediate values can be used by the system designer to denote different trustworthiness levels, if applicable.

Second, we define a set of *dynamic trust metric* functions $\mathcal{M} = \{\mu_l : \mathcal{V} \times \Lambda \to [0, 1]\}$ indexed by a selector l indicating different node attributes (e.g., location) that dynamically change. That is, for each attribute, a different metric μ_l is defined. μ_l takes node $u_k \in \mathcal{V}$ and task $\lambda_j \in \Lambda$ as inputs and returns a real value in $[0, 1]$.

5.3.5 Location and Time

Among the possible values of l for metric μ_l, *proximity* either in *time* or *geographic location* is an attribute of particular importance. Proximity can increase the trustworthiness of a report: The closer the reporter is to the location of an event, the more likely it is to have accurate information on the event. Similarly, the more recent and the closer to the event occurrence time a report is generated, the more likely it is to reflect the system state.

Cryptographic primitives, such as digital signatures, can ensure that location and time information cannot be modified if included in a report. However, the accuracy of such information can vary, due to nodes' differing capabilities or (malicious or benign) faults. This is especially true for reports that depend on fine-grained time and location data. Hence, different types of nodes are more or less trustworthy when reporting such data. In some cases, time- or geo-stamping a report can be a distinct task.

5.3.6 Scheme Overview

We compute the trustworthiness of a report e_k^j, generated by node u_k and providing supporting evidence for event α_j, by using both (i) static or slow-evolving information on trustworthiness, captured by the default values and the event-specific trust f, and (ii) dynamically changing information captured by security status s and more so by metric μ_l. We combine these as arguments to a function

$$F(e_k^j) = G(s(u_k), f(\tau(u_k), \lambda_j), \mu_l(u_k, \lambda_j))$$

that returns values in the $[0, 1]$ interval. If u_k reports no evidence for α_j, $F(e_k^j) = 0$. These values are calculated locally for each report received from another node and are called the *weights* (or *trust levels*) of the reports. Figure 5.1 illustrates our scheme.

Nonetheless, such a per message assessment may often be insufficient. It can be hard to decide whether the reported event took place based on a single message, and it is vulnerable to faults (e.g., equipment failures or compromised nodes). Instead, we propose the collection of multiple reports related to the same event and of their weights, i.e., the accompanying F values, and their combination into a robust decision scheme. Thus, the reports along with their weights are passed to a *Decision Logic* module that outputs an assessment on the event in question. The way to use such decisions and inferences is beyond the scope of this chapter, as it is specific to particular systems.

The above process is tightly related to the multisensor data fusion techniques [45]. In fact, F can be computed using rule-based expert systems; the output of the *Decision Logic* module can be used by another expert system that makes decisions based on reported events.

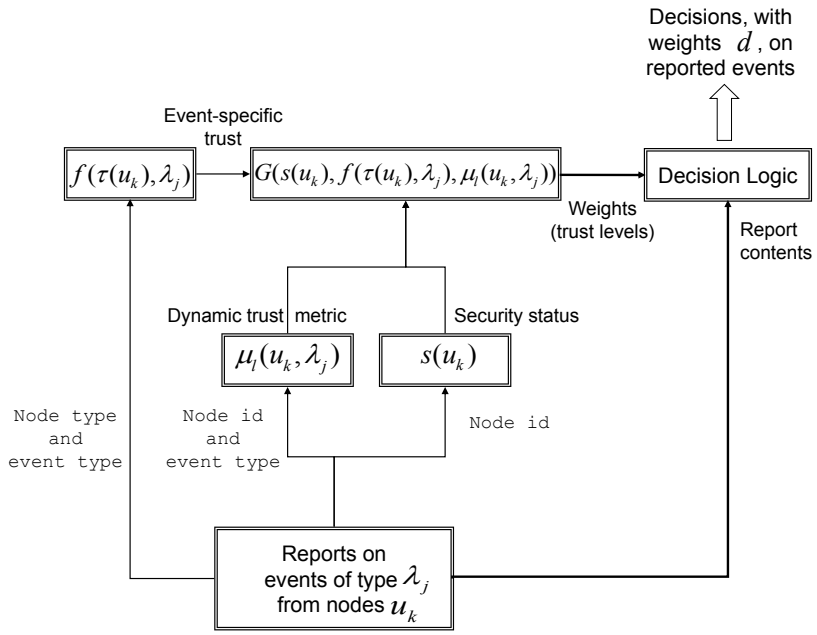


Figure 5.1: Data-centric trust establishment framework.

There are several algorithms to implement the *Decision Logic* module; we will compare next a selected subset of these algorithms in the context of data-centric trust establishment. It should be stressed here that data-centric trust establishment in wireless networks is a new application of data fusion techniques, to the best of our knowledge.

5.4 Evidence Evaluation

The literature on trust in ad hoc networks proposes several approaches for trust establishment, which we survey in Section 5.2. In this chapter, we propose a new technique and compare it to four other existing techniques. These techniques are described below.

To mathematically model our approach, assume a node u_1 has to decide among several basic events $\alpha_i \in \Omega$, based on K pieces of evidence e_k^j (reports from K distinct nodes).

Let d_i denote the combined trust level computed by evaluating evidence corresponding to event α_i. The *Decision Logic* module outputs the event that has the highest combined trust level, i.e., $\max_i(d_i)$.

5.4.1 Basic Techniques

The following two techniques are used for reference and serve as a basis of comparison for the remaining three techniques.

Majority Voting

In this technique, the majority wins (e.g., [75]). The combined trust level corresponding to event α_i is defined by:

$$d_i = \sum_{k=1}^{K} F(e_k^i) \tag{5.1}$$

where $F(e_k^i) = 1$ if u_k reports α_i and it is 0 otherwise.

Most Trusted Report

The Most Trusted Report (MTR) decision logic outputs a trust level equal to the maximum value of trust levels assigned to reports about the event; the point of using MTR is to show the effect of isolated high trust values (in data or entities) on the system. The combined trust level corresponding to event α_i is defined by:

$$d_i = \max_k(F(e_k^i)) \tag{5.2}$$

5.4.2 Weighted Voting

As its names implies, Weighted Voting (WV) sums up all the votes supporting an event with each vote weighted by the corresponding trust level to output the combined trust level:

$$d_i = \sum_{k=1}^{K} F(e_k^i) \tag{5.3}$$

It should be noted here that decisions on composite events are harder to do using the above three techniques since they do not provide formalisms for handling unions and intersections of events. In contrast, the next two techniques provide such formalisms.

5.4.3 Bayesian Inference

Among the data fusion techniques, Bayesian Inference (BI) [80] is the one most frequently used for trust establishment. In BI, the combined trust level corresponding to α_i is the posterior probability of α_i given new evidence $e = \{e_1^j, e_2^j, \ldots, e_K^j\}$; it is expressed in terms of the prior probability $P[\alpha_i]$ using the Bayes' theorem:

$$P[\alpha_i|e] = \frac{P[\alpha_i] \prod_{k=1}^{K} P[e_k^j|\alpha_i]}{\sum_{h=1}^{I} (P[\alpha_h] \prod_{k=1}^{K} P[e_k^j|\alpha_h])} \tag{5.4}$$

where we assume that reports are independent for the sake of mathematical tractability (the receiver cannot sort out the dependencies among reports from distinct vehicles since such information is not provided in the reports).

The computation of posterior probabilities for composite events γ (recall that they are unions or intersections of basic events) follow the rules of probability theory.

$P[e_k^i|\alpha_i]$ is the probability that report k confirms event α_i, given that α_i happened. Using trust levels as weights of reports, this probability is equal to the *trust level*: $P[e_k^i|\alpha_i] = F(e_k^i)$.

For $j \neq i$, $P[e_k^j|\alpha_i]$ is the probability that report k does not confirm α_i (hence, it confirms $\bar{\alpha_i}$, the complement of α_i in Ω), given that α_i happened. This is equivalent to a malfunctioning

or cheating node (ideally, a node would report a real event). Hence, $P[e_k^j|\alpha_i] = 1 - P[e_k^i|\alpha_i] = 1 - F(e_k^i)$.

5.4.4 Dempster-Shafer Theory

In Dempster-Shafer Theory (DST) [87], evidence evaluation is inspired by human reasoning. More specifically, the lack of knowledge about an event is not necessarily a refutal of the event. In addition, if there are two conflicting events, uncertainty about one of them can be considered as supporting evidence for the other. The major difference between BI and DST is that the latter is more suitable for cases with uncertain or no information. More precisely, in DST a node can be uncertain about an event, unlike in BI where a node either confirms or refutes the event. For example, if a node u_1 confirms the presence of an event with probability p, in BI it refutes the existence of the event with probability $1 - p$. In DST, probability is replaced by an uncertainty interval bounded by *belief* and *plausibility*. Belief is the lower bound of this interval and represents supporting evidence. Plausibility is the upper bound of the interval and represents non-refuting evidence. Hence, in this example, node u_1 has p degree of belief in the event and 0 degree of belief in its absence.

In DST, the *frame of discernment* contains all mutually exclusive possibilities related to an observation. Hence, in our context, it is the set Ω defined previously. The belief value corresponding to an event α_i and provided by report k is computed as:

$$bel_k(\alpha_i) = \sum_{q:\alpha_q \subset \alpha_i} m_k(\alpha_q)$$

which means it is the sum of all basic belief assignments $m_k(\alpha_q)$, α_q being all basic events that compose the event α_i. In this case, only $\alpha_i \subset \alpha_i$ and hence $bel_k(\alpha_i) = m_k(\alpha_i)$.

The plausibility value corresponding to event α_i represents the sum of all evidence that does not refute α_i and is computed as:

$$pls_k(\alpha_i) = \sum_{r:\alpha_r \cap \alpha_i \neq \emptyset} m_k(\alpha_r)$$

Belief and plausibility are related by $pls(\alpha_i) = 1 - bel(\bar{\alpha_i})$.

The combined trust level corresponding to event α_i is the belief corresponding to α_i:

$$d_i = bel(\alpha_i) = m(\alpha_i) = \bigoplus_{k=1}^{K} m_k(\alpha_i) \tag{5.5}$$

where pieces of evidence can be combined using Dempster's rule for combination:

$$m_1(\alpha_i) \bigoplus m_2(\alpha_i) = \frac{\sum_{q,r:\alpha_q \cap \alpha_r = \alpha_i} m_1(\alpha_q) m_2(\alpha_r)}{1 - \sum_{q,r:\alpha_q \cap \alpha_r = \emptyset} m_1(\alpha_q) m_2(\alpha_r)}$$

As before, using trust levels as weights of reports, the basic belief assignment that confirms α_i is equal to the *trust level*: $m_k(\alpha_i) = F(e_k^i)$.

For composite events γ, belief can be computed similarly using the above equations.

5.5 Case Study

To illustrate the application and utility of the data trust framework, we present in the following a case study of an ephemeral network instantiation, namely a VANET. We first describe the system and adversary models, then explain through examples how the different components of data trust can be practically derived.

5.5.1 System Model

We assume that the security architecture defined in Chapter 2 is in place. In this example, source authentication identifies the type of the report sender and enables the assignment of default trustworthiness, as explained in Section 5.3.2. It is important to note here that our approach, based exclusively on local processing, does not add any communication overhead and very little computation overhead to a secure VANET where the actual overhead is due to frequent broadcasting and asymmetric cryptography and is inherent in VANETs.

5.5.2 Adversary Model

Nodes either comply with the implemented protocols (i.e., they are correct) or they deviate from the protocol definition intentionally (attackers) or unintentionally (faulty nodes). Both attackers and faulty nodes can cause damage to the network and hence we consider them both as adversaries. The attacks that can be mounted by either internal (equipped with credentials and cryptographic keys) or external adversaries vary greatly. In brief, adversaries can replay any message, jam communications, and modify (yet in a detectable manner due to the digital signatures) messages. More importantly, they can inject faulty data and reports, or control the inputs to otherwise benign nodes and induce them to generate faulty reports.

We assume that at most a small fraction of the nodes are adversaries, and consequently the fraction of the network area affected by them is bounded. This bound on the presence of adversaries could be further refined by distinct values for different node types. But this assumption does not preclude that a few adversarial nodes surround a correct node at some point in time.

5.5.3 Framework Instantiation

We focus on the use of our scheme onboard a vehicle. Clearly, it could be run on RSUs; nonetheless, the challenge is to design a scheme practical for nodes that are not part of the system infrastructure.

The forms of the f (event-specific trust), s (security status), μ_l (dynamic trust metric), and G (trust level) functions are determined by the secure VANET: They are either preloaded at the time the node is bootstrapped, or updated after the node joined the system. Their values are either provided by the authorities or distributed by the infrastructure.

To illustrate our instantiation, we consider an example scenario: a highway accident in which vehicle u_2 is involved. Now, let us consider a vehicle u_1, several communication hops away from the accident location. u_1 receives safety messages indicating that there is an accident on its route and has to decide whether to trust this information. In this case, we assume the event α_1: "There is an accident at location L_{u_2}". The granularity of the event location should be properly defined to avoid having reports on several different events while, actually, all these reports refer to the same event but with slightly different locations. Now

assume that one or more attackers generate safety messages supporting the null event $\alpha_2 = \emptyset$: "There is no accident at location L_{u_2}". If there are several events (e.g., several distinct locations, given the defined granularity), the data trust is computed for each of them. The resulting values can be used by the application to decide the consequent action; the specific use of these values by the application is beyond the scope of this chapter.

Two important system parameters are the set of reports and the time needed to make a correct judgment. The reports considered valid for making decisions should be sent by vehicles that are on the communication path between the accident location and u_1. The time to correct judgment should be equal to the time needed by the *Decision Logic* module to converge to a stable output value; this time depends on the frequency of message reception. It is also constrained by the tolerable decision delay (e.g., in critical situations, decisions should be made very fast, given the available data) that depends on the event in question. The convergence in a typical VANET scenario is illustrated in Section 5.6.4.

5.6 Performance Evaluation

In this section, we examine the performance of the decision logics described in Section 5.4. Recall that a vehicle computes the combined trust in an event based on the reports it receives from distinct vehicles. We compare the four decision logics: MTR, WV, BI, and DST against the basic majority voting scheme. We use the example scenario with two events α_1 and α_2 described in Section 5.5.3.

As noted earlier, the exact choice of the actual values of the f, s, μ_l, and G functions depends on the system designer and hence is out of scope of this chapter. In order to provide an analysis that is independent of administrative decisions, we studied the effects of several general but representative parameters, namely the *percentage of false reports*, *prior knowledge*, *uncertainty*, and *evolution in time*; this allows us to draw conclusions that are independent from the specific choice of default trustworthiness values. We study the effect of these parameters on the *probability of attack success*, which is very important in a security context. In the case of basic majority voting, this probability is equal to 1 if the percentage of attackers is larger than 50%; basic majority voting is represented in the following figures by a vertical or horizontal dashed line corresponding to a percentage of false reports equal to 50 or a probability of attack success equal to 0.5, respectively.

We use a Beta distribution, with its mean equal to an average trust level (defined for each scenario), to assign the trust levels to the reports received by a vehicle u_1. We chose the Beta distribution because it approximates the Normal distribution, a common choice in statistics, but with bounds (0 and 1). We simulate scenarios with 10 or 50 valid reports (i.e., sent by vehicles on the communication path between an accident location and u_1, as described in Section 5.5.3).[1] This means that u_1 includes all these reports in its decision process; each report confirms either event α_1 or α_2. Table 5.1 lists the parameters used in the following simulation scenarios.

Simulations were performed in MATLAB (Sections 5.6.1 to 5.6.3) and ns-2 (Section 5.6.4) with results averaged over 100 randomly seeded runs and plotted with 95% confidence intervals.

[1] We do not simulate the wireless medium in this case since it is orthogonal to our evaluation. Section 5.6.4 simulates a VANET, including the wireless communication.

Scenario number	Parameter		
	$E[F(false)]$	$E[F(correct)]$	K
1	0.6	0.8	50
2	0.8	0.6	50
3	0.6	0.8	10
4	0.2	0.4	50
5,6	0.6	0.8	17

Table 5.1: Simulation scenario parameters. $E[F(false)]$ denotes the average trust level of false reports and $E[F(correct)]$ is the average trust level of correct reports. K is the number of reports. In scenarios 5 and 6, the value of K is determined by the ns-2 simulations as further explained in Section 5.6.4.

The results show that: First, trust decisions based on MTR are the most sensitive to different parameters since the MTR is not corroborated by other vehicles in this case. Second, under realistic conditions, the other three decision logics outperform both majority voting and MTR. Third, there is no clear winner among these decision logics as each performs best in certain scenarios. The details follow.

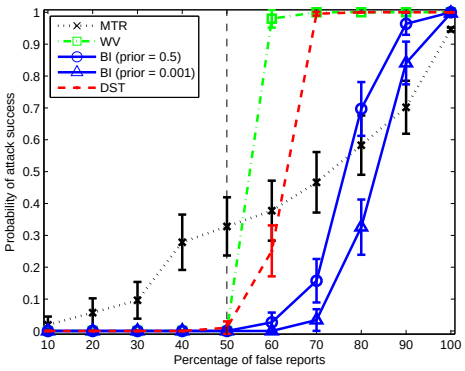
5.6.1 Effect of Data Trust

To see the effect of data trust on the resilience of the decision logic, we compare the different decision logics to majority voting. The graphs in Figs. 5.2(a) and 5.2(b) provide insight into the effect of the percentage of false reports in order to disturb the perception of the observing vehicles (Section 5.5.2). There are two different pieces of information, the false one (originating from colluding attackers or malfunctioning honest vehicles) and the correct one from functional honest vehicles, that are conflicting in their content. Collusion in this case means that all attackers report the same false information. In addition, the trust distributions of the reports generated by honest nodes and by attackers follow Beta distributions with different means. We examine two scenarios: in Scenario 1 (Fig. 5.2(a)), the average trust of false reports is lower than that of correct reports; Scenario 2 (Fig. 5.2(b)) illustrates the opposite situation (e.g., because attackers are positioned closer to the event). The mean values corresponding to both scenarios are listed in Table 5.1.

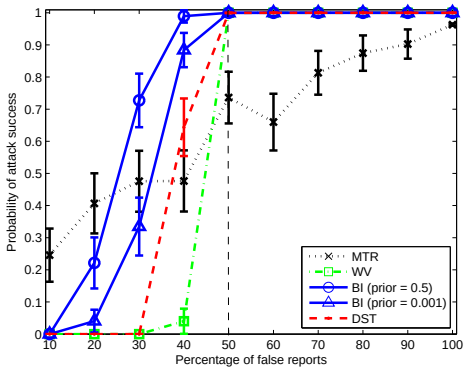
In Figs. 5.2(a) and 5.2(b), we observe that MTR is both little resilient to small percentages of attackers and highly resilient (on average) to high percentages of attackers. This can be explained by the fact that MTR relies on the trust value of only one report, which can differ significantly from the average trust value. The other three decision logics are more resilient to attacks than majority voting when correct reports are more trustworthy than false ones (this is a realistic situation). BI is the most resilient of all three methods. When false reports are more trustworthy than correct ones, the situation is reversed and weighted voting becomes the most resilient technique. There are two curves for BI, each corresponding to a different prior probability; these plots are discussed next.

5.6.2 Effect of Prior Knowledge

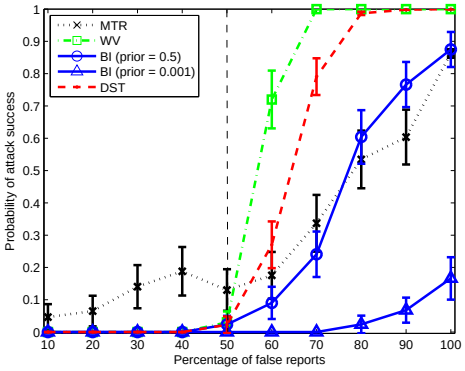
One of the properties of BI is that it uses a prior probability to compute the posterior probability of an event (Section 5.4.3). The prior probability represents the amount of knowledge



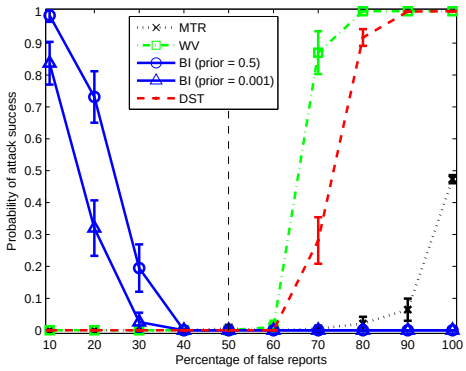
(a) Correct reports are more trustworthy than false ones.



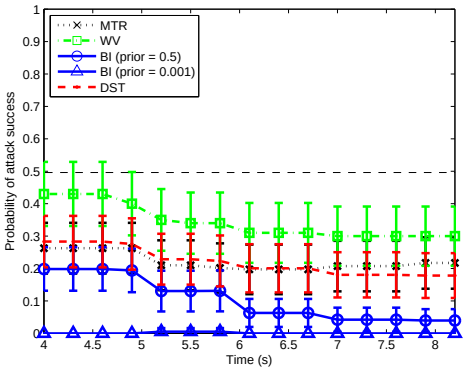
(b) False reports are more trustworthy than correct ones.



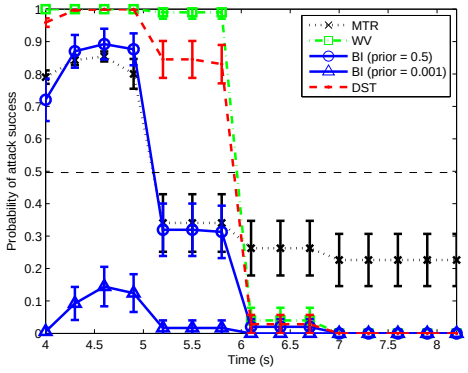
(c) Effect of prior knowledge.



(d) Effect of uncertainty.



(e) Evolution in time; evenly distributed false reports.



(f) Evolution in time; false reports received first.

Figure 5.2: Performance of the decision logics with respect to the percentage of false reports, prior knowledge, uncertainty, and time.

about the event prior to the reception of new evidence; in our example, this is the probability of the presence of an accident. In this section, we study the effect of prior probabilities on the performance of BI. In VANETs, prior probability can be derived from ITS (Intelligent Transportation Systems) studies developed to estimate the probabilities of crash occurrence in different scenarios (e.g., [7]). Since these estimates depend on many parameters, notably the human factors, and cannot be computed in a generic case, we chose a rather conservative prior probability of 0.001 for our simulations. We also simulate the effect of the neutral prior probability 0.5 (i.e., lack of prior knowledge).

Figs. 5.2(a) and 5.2(b) show to some extent that the availability of prior knowledge increases the resilience of BI to false data attacks. In Fig. 5.2(c), there are fewer reports (only 10 compared to 50 in the previous two scenarios) and we can clearly see the benefit of prior knowledge. The reason for this increase in resilience when the number of reports decreases is that large numbers of reports damp the effect of prior probability in the calculation of the posterior probability (Section 5.4.3).

5.6.3 Effect of Uncertainty

In a nutshell, BI does not take uncertainty into account whereas DST does (Section 5.4.4). To simulate the effect of uncertainty (Fig. 5.2(d)) on the decision logics, we use low mean data trust levels for both false and correct reports; the exact values are listed in Table 5.1 (such values can result from low values of the security status s, e.g., due to the discovery of a virus in the network). In this case, DST is indeed the most resilient of the decision logics.

An interesting observation is related to the behavior of BI. At high trust levels (Fig. 5.2(a)), it exhibits behavior similar to that of WV and DST. But at low trust levels (Fig. 5.2(d)), it behaves opposite to the other two methods. This is so because BI deals with only the probabilities of true and false hypotheses and if a report is assigned a 0.2 trust level (i.e., 0.2 probability of being correct), it is assumed to have a 0.8 mistrust level (i.e., 0.8 probability of being false). In fact, BI requires that hypotheses be mutually exclusive and hence does not support general uncertainty that may overlap with either hypothesis. Thus, given a small percentage of supporting reports with low trust levels, there is a high percentage of refuting reports with low trust levels also. In BI, this high percentage is transformed into a high percentage of supporting reports with high trust levels (i.e., the opposite). Similar reasoning applies to high percentages of supporting reports.

It is important to note here that we applied BI in the typical way widely used in both research and industry (e.g., [57]). There are efforts to transform belief function models to probability models, thus enabling BI to handle uncertainty [30]. The use of such methods may reduce the adverse effects of uncertainty on the performance of BI shown in this section.

5.6.4 Evolution in Time

In ephemeral networks, it is important to evaluate data trust rapidly in order to permit an application logic to use the resulting values. Hence, a decision logic should be able to output the final result as fast as possible, based on the freshly received reports. This property distinguishes the mechanisms explored in this work from other approaches that rely on a longer history of available reports (e.g., reputation systems [21, 73, 103]). In this section, we are only interested in the decision delay as reports arrive. The total event detection delay by the observing vehicle depends also on how fast the reporting vehicles detect the event, which

in turn depends on the particular detection sensors and hence we do not consider it in this work.

To simulate ephemeral networks, we use VANETs with mobile vehicles. Our scenario is a 2 km-long highway with 3 lanes in each direction. There are 300 vehicles moving at speeds between 90 km/h and 150 km/h; the average distance between two vehicles on the same lane is 40 m. Vehicles periodically broadcast safety messages every 300 ms within a radius of 300 m (single hop), according to the DSRC specification [4]; the broadcast start times are uniformly distributed between 0 and 2 seconds, approximately. In our simulations, we study the reception of reports at a vehicle u_1 positioned in the middle of the scenario on the 90 km/h lane. We assume that an event (e.g., "ice on the road") is generated by honest vehicles between coordinates 1300 m and 1400 m (the icy section). The attackers report the opposite event ("no ice on the road"). As u_1 moves towards the icy section, it receives reports from vehicles that pass inside this section. Only the last report from each vehicle is considered; this allows u_1 to update its decision as vehicles enter the icy section and change their reports.

The parameters for this scenario are listed in Table 5.1. We observe that each vehicle receives on average (over 100 runs) 17 reports sent by distinct vehicles from inside the icy section. The received reports are assigned corresponding trust levels in MATLAB. In Fig. 5.2(e), the percentage of false reports received in each timestep is drawn from a Binomial distribution with a probability 0.5 (i.e., the mean percentage of false reports is 50); this figure shows the stability of the decision logics when the percentage of false reports varies. In Fig. 5.2(f), the total percentage of false reports is also 50, but all false reports are received at the beginning of the simulation time to simulate the speed of convergence of the decision logics.

By examining Figs. 5.2(e) and 5.2(f), we can see that the speed of convergence of all four decision logics depends on the number of received reports and hence the scenario parameters (event generation time, vehicle density, etc.). We leave further investigation of these parameters to future work due to the lack of space. Nevertheless, we can observe the general trend of convergence to a stable output value before u_1 reaches the icy section and despite variations in the percentage of false reports (of course, as long as the total percentage of false reports is constant). The output values roughly correspond to the results of Fig. 5.2(a), as expected.

5.6.5 Discussion

Based on the above results, we can see that there is no clear winner among the decision logics that fits best all scenarios. But we can elaborate several guidelines for the evaluation of data-centric trust:

- If the uncertainty in the network is low, BI is the most resilient to false reports. To avoid the case of few highly trustworthy false reports (Fig. 5.2(b)), the decision of BI should be positioned with respect to another logic, such as DST or WV, and the most conservative value (i.e., the one that yields the lowest probability of attack success) should be taken.

- The availability of prior knowledge can further improve the resilience of BI.

- If the uncertainty in the network is high, DST performs consistently better than other methods (MTR does not always yield better results).

5.7 Summary

In this chapter, we developed the notion of data trust. We also addressed ephemeral networks that are very demanding in terms of processing speed. We instantiated our general framework by applying it to VANET that are both highly data-centric and ephemeral. We evaluate data reports with corresponding trust levels using several decision logics, namely weighted voting, Bayesian inference, and Dempster-Shafer Theory. Simulation results show that Bayesian inference and Dempster-Shafer Theory are the most promising approaches to evidence evaluation, each one performing best in specific scenarios. More specifically, Bayesian inference performs best when prior knowledge about events is available whereas Dempster-Shafer Theory handles properly high uncertainty about events. In addition, the local processing approach based on either one of the above techniques converges to a stable correct value, which satisfies the stringent requirements of a life-critical vehicular network.

Chapter 6

Solving the Trust-Privacy Tradeoff in an Ephemeral Environment

6.1 Introduction

Security and privacy are often said to be at odds. Much of this tension boils down to the cost of establishing the trustworthiness of entities (e.g., network nodes), which requires them to disclose their private information. Typical solutions to this problem propose trading privacy for trust [86], i.e., gradually revealing an entity's private information to gain a sufficient level of trust. But the majority of these works address e-commerce environments where, rightfully, the emphasis is on establishing the trustworthiness of individual entities. Yet, with the emergence of collective intelligence where only the opinion of a group matters, the data is becoming more important than its sources. This naturally calls for establishing the trustworthiness of the data itself.

In addition to special instances of online environments (e.g., Wikipedia), certain types of wireless networks are data-centric by nature. For example, users of sensor networks require the sensed data to be correct while completely abstracting the sensing platform, i.e., the individual sensor nodes. Another example is *ephemeral* networks, such as VANETs (Vehicular Ad hoc NETworks) [44] and DTNs (Delay-Tolerant Networks) [38], where encounters among nodes are often short-lived. Building entity-centric trust in such cases would be an overkill, if not impossible altogether. These scenarios have resulted in the definition of *data-centric trust* [84].

In a nutshell, data-centric trust is built by collecting all the evidence corroborating a piece of information. Intuitively, the more entities that corroborate the information, the higher the resulting trust value is. This has a valuable side effect if the required trust level is a threshold that is independent of the number of participants: The individual contributions of entities decrease as the number of entities increases and hence the amount of privacy that needs to be traded for trust also decreases. In this chapter, we investigate this collective feature of data-centric trust and show how much privacy is saved with respect to entity-centric trust.[1]

Designing a data-centric trust establishment mechanism can pose several challenges when entities are privacy-preserving and adversaries are present. In fact, a privacy-preserving entity is *rational* by definition, from the privacy point of view. This is because it optimizes its own utility (actually, it minimizes its privacy loss) and hence prefers contributing nothing to the

[1] In this chapter, entity-centric trust refers to trust values derived from the information about a single entity.

system while getting all the benefits, thus creating the free rider problem [39]. The presence of adversaries only aggravates this problem as more investments are required in order to counter attacks. To address these issues, we use game theory to model the strategies of rational entities that try to establish data-centric trust in the presence of rational adversaries. Based on the initial analysis, we prove that using incentives can enable trust establishment and reduce the amount of disclosed privacy.

By addressing the trust-privacy tradeoff in a rational environment, we believe that this chapter constitutes one of the first steps towards studying privacy systems with "human" properties (e.g., rationality), a necessary research area as computers increasingly reflect the preferences of their owners. And privacy is one of the first things that users will try to rationally protect.[2] Thus, our work is the extension to privacy of the works bridging cryptography and game theory [56], and notably rational multiparty computation [46].

To make our approach more concrete, we will make use of the running example of ephemeral networks in this thesis, namely VANETs. The information broadcast by vehicles contains, in addition to reports of traffic jams and accidents, attributes of the senders, such as their credentials and location information. This repetitive disclosure of information can pose privacy threats to the drivers and hence make the system less appealing to potential customers. An additional problem is the validity of event reports in the presence of malicious adversaries or malfunctioning devices in the system [44]. Data-centric trust resolves the latter problem by correlating reports from several senders and relying, for validating the reports, mostly on the attributes relevant to the event rather than the reporters. For example, the closer to the event vehicles are, the more trustworthy their reports are. The reduction in privacy loss stems from the fact that, as long as enough trust in the event is established, reporters do not need to reveal all their attributes.

6.2 Related Work

Seigneur and Jensen [86] propose an approach to achieve the tradeoff between trust and privacy in online transactions where entities use pseudonyms. To preserve privacy, entities use different pseudonyms for different transactions, thus preventing the linkability of these transactions; a reputation level is attributed to each of these pseudonyms. But to increase the level of trust in an entity, the latter has to link several pseudonyms, thus combining the corresponding reputation levels. The number of pseudonyms to link depends on the required trust level.

Lilien and Bhargava [63] discuss the conflict between privacy preservation and trust establishment in online interactions. They assume that users have a set of private attributes that they want to conceal and a set of corresponding credentials that are helpful in establishing trust in these users. The tradeoff problem is formulated as the choice of the minimum number of credentials to be revealed for satisfying trust requirements, such that the users' privacy loss is minimized.

Bhaduri et al. [14] address the privacy issues in P2P data mining. The privacy concern arises when user information must be shared for computing basic operations, such as sum and average, for mining distributed data. They propose secure multi-party computation techniques to protect each peer's local information. Yet, because the collusion of some ma-

[2] Obviously, people are not completely rational, but a properly configured software agent will be able to better enforce the rational preferences of its operator.

licious users could threaten the privacy of other users, they propose a simple game-theoretic model for punishing the misbehaving parties. One of the assumptions in the paper is that the attackers will reveal themselves when they collude. Moreover, the proposed punishing algorithm punishes not only the attackers but also all the other parties.

Providing data aggregation while preserving data privacy in wireless sensor networks is another related problem. To reduce the network overhead of frequent reporting to the sink, sensor nodes send their data to intermediate nodes to be aggregated, thus revealing these data to the aggregators. As end-to-end (node-to-sink) encryption of the data makes the data aggregation expensive, He et al. [47] propose a privacy-preserving data aggregation mechanism, called PDA, to solve the problem. They use techniques such as data slicing, where encrypted data slices are forwarded to random nodes in the network for aggregation; only the sink is able to collect enough encrypted slices to decrypt the data.

Online reputation systems, where users report the quality of products, pose problems essentially of data-centric trust. Jurca and Faltings [55] analyze incentive mechanisms for honest reporting in these systems and propose payment schemes that rely on the correlation among the reports of different reporters. Acquisti et al. address the use of incentives for providing and using anonymity services [9].

Another parallel that can be drawn is between our game-theoretic model and privacy-preserving auctions, first proposed by Naor et al. [74]. In the latter, participants try to win the auction while keeping their valuations secret. Recent work by Miltersen et al. [69] proposes a hybrid utility model that includes monetary and privacy components. In our model, players have to build trust, by revealing their private information, in the presence of adversaries.

Last but not least, there is a close similarity between our problem and rational multiparty computation (MPC) in cryptography [46]. After all, data-centric trust is a function that needs to be computed by several entities. To realize the similarities and differences between the two problems, we first need to present our model in more detail and hence devote Section 6.6.2 to drawing a comparison.

6.3 System and Threat Model

We assume a wireless network where communication among entities (i.e., network nodes) is locally broadcast (e.g., within a range of 300 m in VANETs) and each entity is able to receive an ongoing broadcast before sending a new message, i.e., communication is *sequential*. We also assume that there are deadlines on making decisions and model this by making communications right before the deadline simultaneous and not sequential, implying that collisions may happen and some messages may not be received by the deadline. Entities are computationally powerful and are capable of using public-key cryptography. Hence, we assume that messages are digitally signed and their senders can be anonymously authenticated, e.g., using anonymous public keys with valid credentials issued by a Certificate Authority (CA). All entities are distinguishable, i.e., they have different identifiers. If an entity has several legitimate identities, such as pseudonyms, we consider them as different entities. This means that the CA is responsible for preventing the abuse of multiple identities. One way to achieve this is to issue identities with non-overlapping validity periods and make the obtention of new identities costly.[3] Credentials have different trust values, i.e., some entities are more

[3]We assume that entities have synchronized clocks, e.g., by means of the GPS units onboard vehicles.

trustworthy than others. In addition to credentials, entities communicate other attributes, such as their location. We assume that entities cannot lie about these attributes without being detected (e.g., due to a position verification system [62]). We also assume that entities are privacy-preserving (i.e., they are rational from the privacy point of view) and hence minimize their privacy losses (i.e., they optimize their own utilities).

The goal of the system is to disseminate truthful information. Entities are divided into two groups: benign and adversarial. Adversaries have the same properties as benign nodes, but disseminate false information. This can be due to an intended attack or merely a fault in an entity's information acquisition or generation systems. Adversaries can also collude by disseminating the same false information to increase its trustworthiness. Entities reveal their attributes (e.g., credentials, precision of location, etc.) to increase the trust level of the information they disseminate. But by revealing these attributes, entities also reduce their privacy.

In our model, we do not address the leakage of private information by mechanisms other than the trust establishment mechanism. For example, individual entities may reveal their network-layer identifiers, such as IP addresses, when sending information. Although this constitutes a privacy loss, it is orthogonal to our model. In fact, we model only the loss of information that entities cannot conceal (they need to provide evidence), whereas network-layer identifiers can be anonymized.

6.4 Trading Privacy for Trust

One of the difficult questions about trading privacy for trust is how to convert one into the other. Hence we need quantitative metrics to represent both in "convertible" units. A straightforward way to do this is to represent both trust and privacy as functions of common parameters, notably because both depend on the type and amount of information revealed during communication. In this section, we develop mathematical expressions for trust and privacy, explain how data-centric trust reduces privacy loss and we quantify the corresponding privacy improvement with respect to entity-centric trust.

6.4.1 Trust

To express trust, we further develop the formalism initially proposed in [84]. We assume that an information verifier V (in the VANET example, V is a vehicle) needs to make a decision about a reported event. Assuming that K entities provide to V reports e_k, where $k \in \{1, \ldots, K\}$, the entity-centric trust t_k in each report is:

$$t_k = F(e_k) \tag{6.1}$$

and the data-centric trust d in this event is given by a weighted voting rule[4]:

$$d = \sum_{k=1}^{K} F(e_k) \tag{6.2}$$

[4]Other methods for evaluating data-centric trust are also possible, but the voting mechanism has a good performance and a simple expression suitable for mathematical analysis.

If there are several possible conflicting decisions about the event, the reports supporting each decision are combined, according to the above equation, and the resulting trust levels are compared. The decision with the highest trust level is acted upon.

The trust computation function is given by:

$$F(e_k) = G(s_k, f(\tau_k, \lambda), \mu_k) \tag{6.3}$$

where $G : [0,1]^3 \rightarrow [0,1]$ is a function, u_k is the identity of a reporting entity, s_k is its security status (e.g., a reputation value or a bit to indicate whether the entity has been revoked), τ_k is the type of the entity (some entities, such as police cars, are more trustworthy than others), λ is the type of the event (e.g., accident, traffic jam, etc.), $f(\tau_k, \lambda)$ represents the fact that certain entities are better capable of reporting certain types of events (e.g., police cars are better informed of road accidents than other vehicles), and μ_k is a trust metric that depends on dynamic parameters, such as entity location with respect to the event (assuming that the closer the reporter is, the more trustworthy its reports are). We further develop F as a function of the above parameters:

$$F(e_k) = s_k(\omega_1 f(\tau_k, \lambda) + \omega_2 \mu_k) \tag{6.4}$$

where ω_1 and ω_2 are weights such that $\omega_1 + \omega_2 = 1$.

The reason for making s_k a multiplicative factor is that when $s_k = 0$ (i.e., the entity is revoked), the entity is not trustworthy and hence we need $F(e_k) = 0$. The parameters $f(\tau_k, \lambda)$ and μ_k do not have this property and hence should not be used as multiplicative factors. In our analysis, we take their weighted sum.

To obtain a simple yet illustrative closed-form solution, we consider a scenario where, $\forall k \in \{1, \ldots, K\}, s_k = 1$ (i.e., none of the entities are revoked), $\omega_1 = 0$ and $\omega_2 = 1$. This represents the scenarios where proximity to the event is the only source of trust. It is important to note that $0 \leq \mu_k \leq 1$, where $\mu_k = 0$ means that the reporting entity is able to observe the event (without any further precision of its location) and $\mu_k = 1$ means that the reporting entity is generating the event. Of course, entities can choose to reveal their proximity to the event only partially, by reporting smaller values of μ_k. Based on these assumptions, we obtain the following simplified expressions where $\overline{\mu}$ is the average precision over all μ_k:

$$t_k = \mu_k \tag{6.5}$$

$$d = \sum_{k=1}^{K} \mu_k = K\overline{\mu} \tag{6.6}$$

6.4.2 Privacy

Although there has been extensive research on privacy metrics, in this work we will use one of the simplest, for the sake of clarity. More precisely, we will measure the privacy of a given entity u_k by the size of its *anonymity set* $\mathcal{A}_k$, the set of entities where the entity in question is not identifiable [82]. For a given μ_k, the size of the anonymity set can be expressed by:

$$|\mathcal{A}_k| = K(1 - \mu_k)^2 \tag{6.7}$$

where K is the total number of reporting entities and hence the maximum size of the anonymity set. Assuming that the reporting entities are uniformly distributed around the

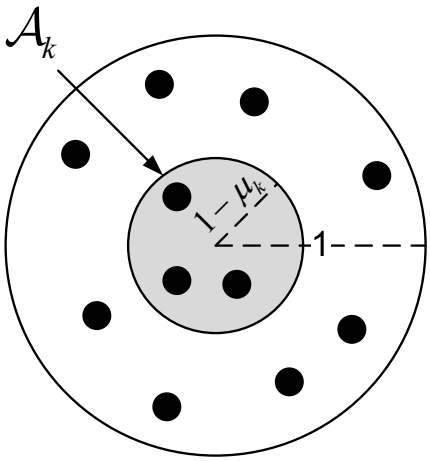


Figure 6.1: Privacy reduction by revealing the precision of proximity to the event (the center of the disk). The information verifier can be any entity receiving information from the reporting entities.

event, the whole anonymity set can be modeled by a disk centered at the event and with K entities inside. Now assuming that an entity reports a precision of μ_k, it limits the anonymity set to a smaller disk with a radius $1 - \mu_k$. The number of entities inside the smaller disk is a fraction of K equal to the ratio of the respective areas of the two disks. Figure 6.1 illustrates the above reasoning.

6.4.3 Tradeoff

Figure 6.2 illustrates how privacy is traded for trust. Each entity u_k possesses attributes s_k, τ_k, μ_k as defined in Section 6.4.1. To support a piece of information, this entity reveals one or more of its attributes. The more attributes are revealed to the information verifier V, the more trustworthy the information is, but also the more privacy the entity loses. In the case of entity-centric trust, a single entity has to reveal its attributes to supply a sufficient level of trust in the information it is providing [86]. In the case of data-centric trust, several entities can reveal a smaller subset of their attributes such that the aggregate of these attributes provides a sufficient level of trust.

Hence, data-centric trust allows each entity to preserve more private information than entity-centric trust, but without decreasing the level of established aggregate trust. To quantify the resulting reduction in privacy loss, we define the *Privacy Improvement* (PI), which is the amount of privacy saved due to data-centric trust. Let $\mathcal{A}_d$ and $\mathcal{A}_k$ be the anonymity sets corresponding to data-centric trust and entity-centric trust (for an entity u_k), respectively. Let θ be the threshold trust level that the information verifier requires to accept a piece of information. Based on (6.5), (6.6), and (6.7), entities need to reveal the following minimal precisions of their locations:

$$t_k = \mu_k = \theta \Rightarrow |\mathcal{A}_k| = K(1-\theta)^2 \tag{6.8}$$

$$d = K\overline{\mu} = \theta \Rightarrow \overline{\mu} = \frac{\theta}{K} \Rightarrow |\mathcal{A}_d| = K\left(1 - \frac{\theta}{K}\right)^2 \tag{6.9}$$

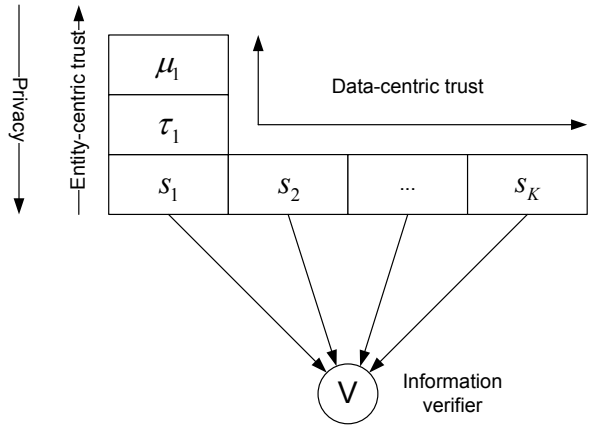


Figure 6.2: The entity-centric trust increases along one dimension only, whereas data-centric trust increases along two dimensions. Entity privacy decreases only along one dimension.

Based on (6.7), (6.8), and (6.9), the resulting privacy improvement PI is:

$$PI = |\mathcal{A}_d| - |\mathcal{A}_k| = \theta(K-1)\left[2 - \theta\left(1 + \frac{1}{K}\right)\right] \tag{6.10}$$

Figure 6.3 shows how PI varies with K and θ. For a given θ, the larger K is (i.e., the more entities contribute to trust establishment), the larger PI is. Alternately, the individual contributions of entities decrease as more entities contribute.

6.5 Trust-Privacy Games

By definition, privacy-preserving entities try to minimize the loss of their private information, which implies that they behave *rationally* (i.e., they try to optimize a given utility function) from the privacy point of view. And given that private information is traded for trust, rational entities make data-centric trust establishment difficult because it has to compromise conflicting requirements: Privacy-preserving entities have to contribute a sufficient level of trust without unnecessarily revealing too much private information. This optimal level is obviously the threshold trust level in the absence of an adversary, but when an adversary tries to surpass the benign entities, the latter have to enter a competition with the adversary while trying to minimize the attributes they reveal. In addition, as different combinations of entities (from the total set of K) can reach the threshold trust level, with each entity contributing an adjustable amount of its private attributes, it becomes paramount to find and implement a mechanism that makes the individual contributions both sufficient and fair. It is even questionable whether the entities would contribute at all to trust establishment.

The above questions naturally call for the use of game theory to solve two related games: the *Attacker-Defender Game* $\mathbf{G_{AD}}$ and the *Trust Contribution Game* $\mathbf{G_{TC}}$. $\mathbf{G_{AD}}$ captures the competition between attackers and defenders to support their respective versions of the truth. $\mathbf{G_{TC}}$ models the details of $\mathbf{G_{AD}}$ by defining the individual amounts of privacy to be contributed by benign entities to collectively win $\mathbf{G_{AD}}$. Put differently, $\mathbf{G_{AD}}$ is on the macroscopic level where the attacker and the defender represent the sets of adversarial and

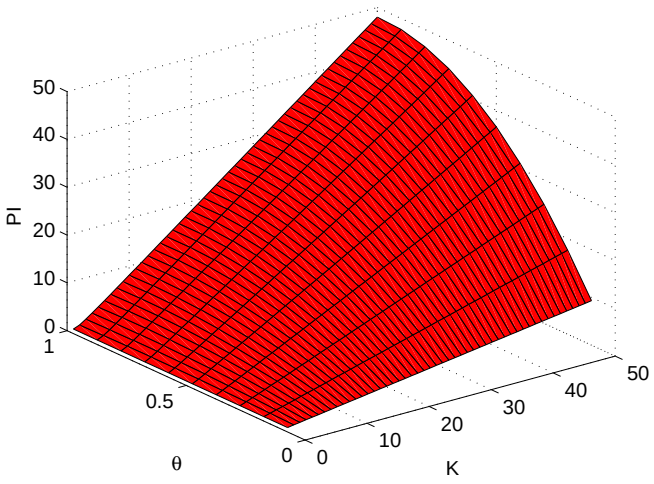


Figure 6.3: Privacy Improvement is the increase in the anonymity set size when using data-centric trust instead of entity-centric trust.

benign entities, respectively. $\mathbf{G_{TC}}$ analyzes at the microscopic level how benign entities behave individually to collect the defender's trust level in $\mathbf{G_{AD}}$. Figure 6.4 illustrates this duality. In the following, we refer to players in $\mathbf{G_{AD}}$ as *macroplayers* and to players in $\mathbf{G_{TC}}$ as *microplayers*.

6.5.1 Game-Theoretic Model

To model the trust-privacy games, we need to make some assumptions about the way the macroplayers interact (Section 6.3 describes the model for microplayers). First, we assume that the information verifier needs to make a decision by a given deadline (e.g., in VANETs, vehicles need to learn about an accident before entering the corresponding danger zone). Second, we assume that macroplayers can observe, before acting, the actions of preceding macroplayers in all but the last stage of the game (cf. Section 6.3). In this last stage, just before the deadline, both macroplayers try to act in order to win the game and are thus unaware of the actions of the other macroplayer. As the action of only one macroplayer will be retained in the last stage, we need to assign the probabilities of winning to each macroplayer. The resulting game is called *dynamic Bayesian game* where "dynamic" means that the game is sequential and "Bayesian" refers to the probabilistic nature of the game. Dynamic games are represented by a tree where the players occupy the nodes of the tree and their actions are the branches descending from the respective nodes. Last but not least, we assume that macroplayers have enough privacy to trade for trust. This assumption is reasonable in an environment where new entities can appear, thus increasing the available privacy resources. The information verifier has two options of action when receiving information with an insufficient trust level. The first option is to take the information with the highest, though insufficient, trust level. The second option, if time permits, is to broadcast a request for new evidence, thus restarting the trust games. The solution concept for dynamic Bayesian games is called *Perfect Bayesian Equilibrium* (PBE) and can be computed by finding the *best response* of

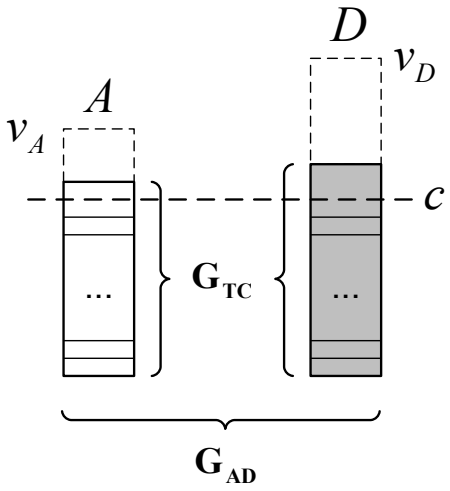


Figure 6.4: The duality between the trust-privacy games. $\mathbf{G_{AD}}$ is between the two groups A and D, whereas $\mathbf{G_{TC}}$ determines how microplayers in each group contribute to $\mathbf{G_{AD}}$. The winner of $\mathbf{G_{AD}}$ is indicated by the shaded rectangle (note that D reaches a higher level of trust, by revealing more private information, than A). The dotted rectangles represent the gains v_A and v_D of the macroplayers in the case of winning the game. c is the minimum amount of privacy required to reach the threshold trust level θ.

each player, i.e., the set of actions that maximize the player's utility given the actions of the other players. For further conceptual details, we refer the interested reader to [39].

6.5.2 Attacker-Defender Game

In $\mathbf{G_{AD}}$, both the attacker and the defender try to prove to the information verifier their respective versions of the truth about an event. It is worth reiterating here that the two macroplayers (attacker and defender) actually represent the groups of adversarial and benign nodes, respectively, on the macroscopic level. The winner is the macroplayer that succeeds in providing the higher trust level before the game deadline. As the game is dynamic, each macroplayer has the possibility to play more than once, each time surpassing the other macroplayer's previous action. As the attributes needed to increase the trust level also diminish a macroplayer's privacy, each macroplayer should try to reveal as few attributes as possible in surpassing the other macroplayer.

Figure 6.5 illustrates $\mathbf{G_{AD}}$. The two macroplayers are A (attacker) and D (defender) with two possible actions: S (send attributes to the information verifier V) and W (wait until the next stage). When sending, each macroplayer increases the level of trust in its information but the opponent can surpass it in the next stage, thus requiring the first macroplayer to disclose even more attributes in the subsequent stage.[5] By waiting, a macroplayer has a probability (p_A for A and p_D for D) of winning the game without the escalation of attribute investment. The winner has to provide a trust level at least equal to a defined threshold, θ as defined in Section 6.4.3. Let c be the privacy loss required to reach θ. Hence, each macroplayer is

[5] Although more stages can be included in the model, we consider only 3 stages to keep the analysis tractable. We leave the extension to a general number of stages n to future work.

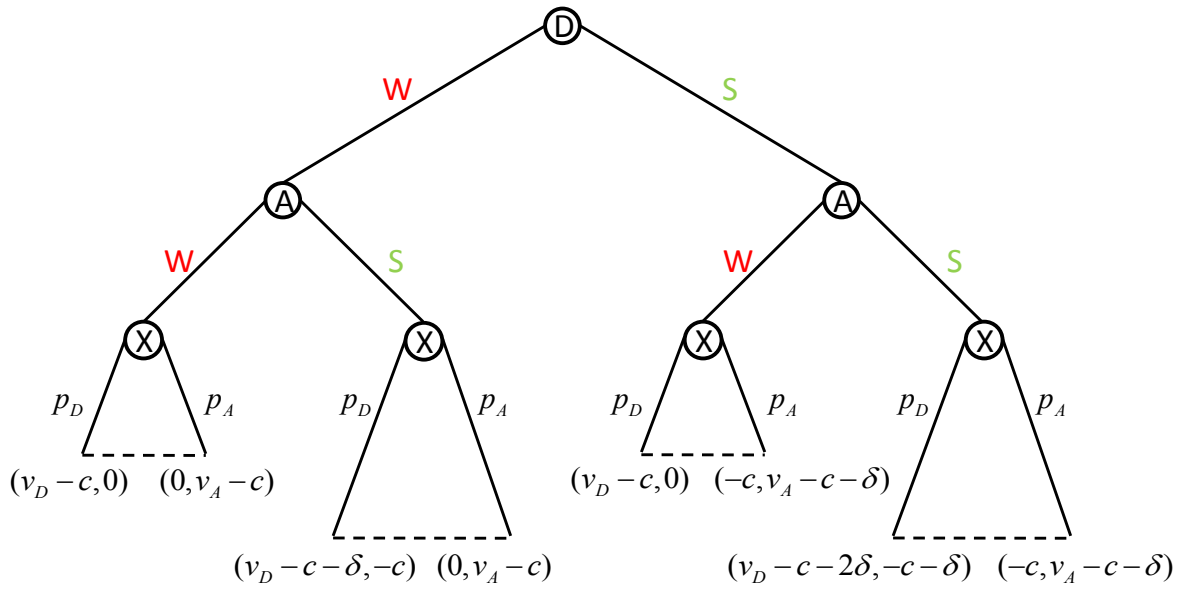


Figure 6.5: The Attacker-Defender Game $\mathbf{G_{AD}}$. The X in the last node indicates that both macroplayers play in the last stage. The tuples at the leaves of the tree represent the payoffs of the macroplayers; the negative values are the costs (invested privacy) and the positive values are the gains when winning the game. The probabilities of winning are such that $p_D + p_A = 1$. For example, the rightmost leaf of the tree represents the case where both macroplayers play S and A wins. A realizes a gain of v_A at the cost of surpassing D's previous action (D revealed the minimum amount of privacy c) by the increment δ.

required to invest at least an amount c of privacy to win the game. Let δ be the minimum increment required to surpass the previous action. Last but not least, let v_A and v_D be the gains of macroplayers A and D, respectively, when winning the game. v_A represents how much the attacker benefits from a successful attack, whereas v_D represents the cost that the defender avoids by preventing the attack.

In practice, p_A and p_D depend on the individual entities that compose A and D. For example, in a shared communication medium (such as an IEEE 802.11 wireless network), if all entities have the same access probability, p_A and p_D are the fractions of entities in A and D, respectively, out of the total number of entities. It should be noted here that even if a macroplayer knows that its probability of winning is small (e.g., because it is outnumbered), it can still participate as it loses nothing in the last stage (only the winner of this stage gets to transmit its private information).

Equilibrium

By solving $\mathbf{G_{AD}}$, we can find its equilibrium, i.e., the set of player strategies from which none of the macroplayers can unilaterally deviate while realizing a better payoff. We assume that the defender plays first, as it has to establish trust in the information it provides and may not be aware of the attacker (this is its typical behavior when no attacker is present). In the following solution of $\mathbf{G_{AD}}$, the tuple (σ_D,σ_A) contains the strategies of D and A, respectively.

Theorem 6.5.1. *The strategy* (W, WW) *is a PBE of* $\mathbf{G_{AD}}$.

This means that D's best strategy is to play always W and A's best-response strategy is to play W when D plays either W or S. All proofs of theorems are provided in the Appendix to this chapter. In practice, both macroplayers wait until the last stage where they rely on their respective probabilities p_D and $p_A = 1 - p_D$ to win; both macroplayers actually play S in the last stage.

Incentives

The previous result for $\mathbf{G_{AD}}$ is not desirable because the information verifier can decide on the information only at the deadline. In addition, the defender can win only probabilistically. Hence, it is beneficial to design a version of the game where macroplayers send their attributes earlier (i.e., play S from the beginning). One way to achieve this is to give both macroplayers incentives to play earlier. In this section, we analyze the game with incentives $\mathbf{G^I_{AD}}$. Figure 6.6 illustrates this game. The only difference with respect to $\mathbf{G_{AD}}$ is that if a macroplayer plays S before the last stage, it receives a reward r (this can be a *bonus* trust level, as defined in Section 6.6.1) from the information verifier regardless of whether it wins or loses the game. The resulting equilibrium is not constrained to waiting as the theorem below shows.

Theorem 6.5.2. *The PBE of* $\mathbf{G^I_{AD}}$ *is achieved by the following strategies:*

$$
\begin{array}{lll}
(W, W) & \textit{if} & r \leq \min\{p_D c, p_A c\} \\
(S, S) & \textit{if} & r > \max\{p_D(c+\delta), p_A c + p_D \delta\} \\
(W, S) & \textit{if} & (p_D c < r \leq \min\{p_D(c+\delta), p_A c - p_D \delta\}) \\
 & & \vee (p_D(c+\delta) < r \leq p_A c + p_D \delta) \\
(S, W) & \textit{otherwise} &
\end{array}
$$

To enforce the strategy (S, S), we need $r > \max\{p_D(c+\delta), p_A c + p_D \delta\} \quad \forall p_D \leq 1$. This implies $r > c + \delta$. If $r = c + \delta$ and $p_D = 1$, the best-response strategies are (S, W), but as there is only player D (because $p_A = 0$), (S, W) is equivalent to (S, S) in this case. This justifies not requiring the strict inequality $r > c + \delta$ and leads to the following corollary:

Corollary 6.5.1. *The strategy* (S, S) *can be enforced by choosing* $r \geq c + \delta$.

As both the defender and the attacker can receive the reward r, the amount of the reward should be minimal and still satisfy the condition for enforcing (S, S).

The definition of the last stage in the attacker-defender games merits some clarifications. One of the main questions is "When does the last stage start?" As the probability of successfully sending evidence is 1 until the last stage, we assume that this stage begins when the above probability becomes lower than 1. Determining this exact moment is admittedly difficult, hence there is a possibility that, after one macroplayer plays, the other macroplayer could succeed in transmitting more evidence and surpass the first macroplayer. Although in this case the model of $\mathbf{G_{AD}}$ does not apply anymore, that of $\mathbf{G^I_{AD}}$ does because macroplayers actually send evidence before the last stage. Thus, $\mathbf{G^I_{AD}}$ implicitly models the case when macroplayers incorrectly estimate the beginning of the last stage.

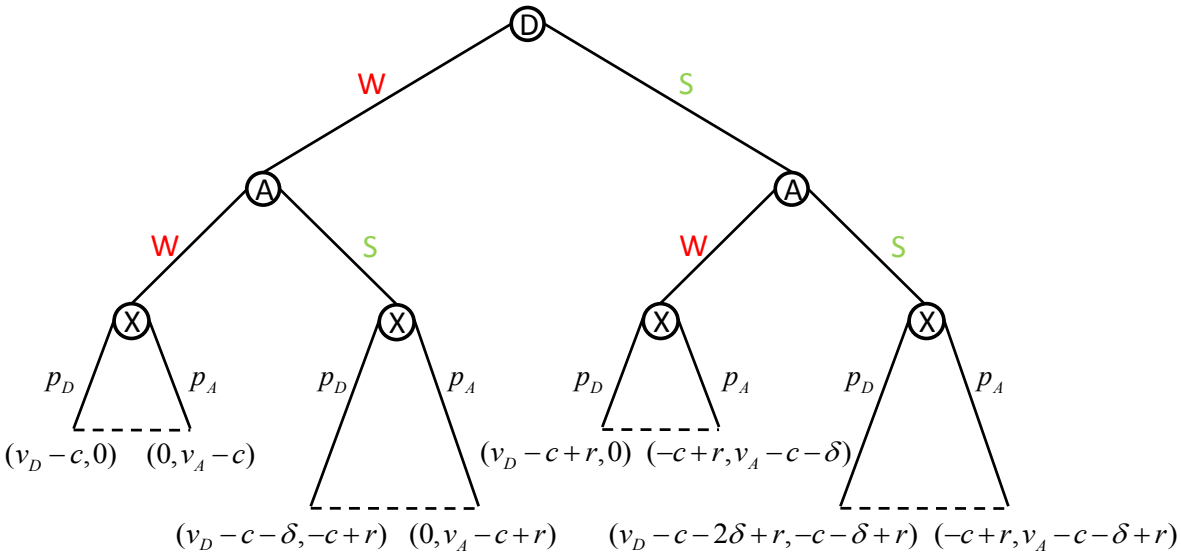


Figure 6.6: The Attacker-Defender Game with Incentives $\mathbf{G^I_{AD}}$. Macroplayers that play S before the last stage of the game receive a reward r.

6.5.3 Trust Contribution Game

The trust contribution game $\mathbf{G_{TC}}$ captures how individual benign rational entities contribute to the defender trust level in $\mathbf{G_{AD}}$. As before, let $t_k \leq 1$ be the entity-centric trust levels contributed by $K \geq 2$ entities. Each entity has to set the value of t_k by providing some private information. Before going into the details of the analysis, we need to define exactly how private information (i.e., attributes) is converted into trust. Let ϕ_k be the *trust-privacy conversion factor*: 1 unit of private information $= \phi_k$ units of trust. Based on (6.5), $\phi_k = \phi = 1$, although generally it is different for each entity according to (6.4). Based on this, the threshold trust and privacy levels are linked as follows: $\theta = \phi c = c$. We assume that entities play sequentially, i.e., they observe the trust levels contributed by previous microplayers. We also assume that entities know the target collective trust level based on the analysis of $\mathbf{G_{AD}}$ above.

Equilibrium

In $\mathbf{G_{TC}}$, the payoff of entity u_k, in private information units, is:

$$\forall k \in \{1,\ldots,K\}, \quad \pi_k(t_1,\ldots,t_K) = \frac{v_D}{K} - \frac{t_k}{\phi} \tag{6.11}$$

This means that winning $\mathbf{G_{AD}}$ benefits all contributing entities equally (e.g., by avoiding the cost induced by a false information attack), but each one contributes a different level of privacy. To solve $\mathbf{G_{TC}}$, we compute its *Subgame-Perfect Equilibrium* (SPE).[6] The resulting equilibrium is, unsurprisingly:

[6]The SPE is stronger than the typical Nash equilibrium because there is only one SPE equilibrium in a game, whereas there can be several Nash equilibria. See Appendix 6.C for more details.

Theorem 6.5.3. *The SPE of* $\mathbf{G_{TC}}$ *is defined by:*

$$t_k^* = 0$$

In practice, this result means that no entity will contribute in $\mathbf{G_{TC}}$, thus making it impossible to collect the required trust levels in $\mathbf{G_{AD}}$. We solve this problem in the next section.

Incentives

Adding incentives to $\mathbf{G_{AD}}$ results in macroplayers sending their attributes in earlier stages of the game (Section 6.5.2). In this section, we investigate the effect of incentives on $\mathbf{G_{TC}}$. Let $\mathbf{G^I_{TC}}$ be the version of $\mathbf{G_{TC}}$ with incentives; $\mathbf{G^I_{TC}}$ corresponds to $\mathbf{G^I_{AD}}$. The payoff of entity u_k, in private information units, is:

$$\forall k \in \{1, \ldots, K\}, \quad \pi_k(t_1, \ldots, t_K) = \frac{v_D}{K} + r\frac{t_k}{\sum_{i=1}^{K} t_i} - \frac{t_k}{\phi} \tag{6.12}$$

This payoff function takes into account the reward r attributed to the early movers in $\mathbf{G^I_{AD}}$ (i.e., the attacker or the defender). Among the rewarded entities of one group (e.g., the entities constituting the defender), the reward should be distributed proportionally to the individual contributions of these entities to encourage high contributions. The resulting equilibrium solves the problem in Theorem 6.5.3.

Theorem 6.5.4. *The SPE of* $\mathbf{G^I_{TC}}$ *is defined by:*

$$\forall k \in \{1, \ldots, K\}, \quad t_k^* = \frac{\phi r(K-1)}{K^2}$$

We still need to compute r while considering Corollary 6.5.1. Back to $\mathbf{G^I_{AD}}$, if macroplayer D (the defender) wants to win the game, it should contribute at least $c + 2\delta$ of private information. This results in the following values for r and t_k^*:

$$\sum_{k=1}^{K} t_k^* \geq \phi(c + 2\delta) \tag{6.13}$$

Substituting t_k^* from Theorem 6.5.4 and taking into account the requirement that r should be minimal to prevent generously rewarding the adversary, we obtain:

$$r = \frac{(c + 2\delta)K}{K - 1} \tag{6.14}$$

$$t_k^* = \frac{\phi(c + 2\delta)}{K} \tag{6.15}$$

Equation (6.14) satisfies the lower bound condition on r. The upper bound on δ can be computed, based on (6.15), as follows:

$$t_k^* \leq 1 \Rightarrow \delta \leq \frac{K - \phi c}{2\phi} \tag{6.16}$$

6.6 Discussion

In this section, we address several important issues not covered earlier in the chapter. First, we elaborate on the reward mechanism. Then, we compare our framework with MPC.

6.6.1 Reward Mechanism

A straightforward question concerning the individual rewards in $\mathbf{G^I_{TC}}$ (Section 6.5.3) is about how to actually distribute them to the contributors. As entities have to trade privacy for trust, a straightforward way to reward them is to increase their trust levels (by adding a bonus trust value) based on the amount of their contributions in $\mathbf{G^I_{TC}}$. These bonus trust levels can be used in the next $\mathbf{G^I_{TC}}$. The information verifier V distributes the rewards at the end of each game $\mathbf{G^I_{TC}}$. Let $b_{k,V}$ indicate the bonus trust value awarded by V to entity u_k. Based on (6.12), (6.14), and Theorem 6.5.4:

$$b_{k,V} = \phi r \frac{t_k^*}{\sum_{i=1}^{K} t_i^*} = \phi \frac{r}{K} = \frac{\phi(c+2\delta)}{K-1} \tag{6.17}$$

This means that the bonus trust level is slightly larger than the trust contribution based on (6.15), which actually justifies the choice of entities to contribute early.

The reward mechanism should satisfy several requirements. First, as there may be no reputation system, centralized or distributed, the value of an entity's reward should be self-contained. Alternately, the information verifiers in subsequent games should be able to learn the bonus trust level from the reward itself and not from a different mechanism (e.g., a reputation system). Second, the use of the reward should be limited; this can be done by attributing an expiration time or a use counter to the reward. In addition, if an entity signs, using the same key, several rewards for another entity, only one of these is considered valid.[7] Last but not least, an entity that receives rewards from different sources should be able to aggregate these rewards.

To address the above issues, we propose using *general aggregate signatures* [19] where the signing keys have short lifetimes. General aggregate signatures allow any entity to aggregate the signatures of other entities in any order, and short lifetimes limit potential abuse and encourage entities to participate repeatedly in the $\mathbf{G^I_{TC}}$ games. The lifetime $T_{k,V}$ can be set to the upper bound on the time lapse between two consecutive encounters for a typical entity. Let $\mathsf{GenSign}$ designate the algorithm for generating individual signatures. Although any general aggregate signature scheme can be used, we suggest using the BGLS signature [19] due to its small size. Thus, the format of an individual reward will be:

$$u_k \| b_{k,V} \| T_{k,V}, \mathsf{GenSign}_V(u_k \| b_{k,V} \| T_{k,V}), Cert_V \tag{6.18}$$

where $Cert_V$ is the certificate of the information verifier's signing key.

6.6.2 Comparison with MPC

In cryptography, multiparty computation (MPC) aims at computing a function based on secret inputs held by different parties. Each party wants to learn the output of the function,

[7] It should be noted that an entity cannot issue several distinct rewards to another entity (e.g., the case of collusion) because the issuing entity has only one valid signing key at any instant (cf. Section 6.3).

but without revealing its secret input. An example of MPC is Shamir's secret sharing scheme [88]. In traditional MPC, the parties are divided into "good" and "bad" where the latter try to disrupt the computation or learn the secrets of the other parties. Recent work by Halpern and Teague [46] considers a rational version of MPC where all parties are primarily interested in learning the output of the function and secondarily in keeping this output secret from as many other participants as possible. They show that, under these assumptions, Shamir's scheme will not work as none of the parties will share their secret. This result is actually close in spirit to Theorem 6.5.3. It should be clear that the objectives of players in rational MPC are different from those in trust-privacy games. In addition, most rational MPC protocols assume a simultaneous channel where all parties play at once. Although non-simultaneous channels were recently analyzed, they still require a predefined broadcast order [59], which is not the case in our scheme.

Another distinction between rational MPC and trust-privacy games is the notion of resilience to coalitions. Whereas *k-resilient equilibria* tolerate deviations by coalitions smaller than k [8], the winner in an attacker-defender game does not necessarily depend on the size of its respective coalition. For example, a single entity with enough attributes can surpass a large coalition of untrustworthy entities. Hence, we consider our model to be more flexible than the typical approach that considers only the coalition size.

MPC meets game theory in another context. Some game-theoretic solutions require a trusted *mediator* and MPC is an appropriate tool for simulating this mediator with a distributed cryptographic protocol [33]. This is typically achieved with *cheap talk*, costless communication among the parties prior to playing the game. In the trust-privacy games, we do not need a trusted mediator. Instead, all the information is locally combined by the information verifier.

To summarize, rational MPC can potentially solve the trust-privacy games, but it incurs, due to its generality, several modeling and efficiency constraints (general MPC protocols typically require interactive computations, such as the distribution and combination of secret shares). By creating a customized model, we avoid these complications and provide an efficient solution. It is worth noting that the application of rational MPC to trust inference in distributed systems is one of the proposed future directions for research in the MPC community [56].

6.7 Summary

In this chapter, we bring three novel contributions to the research on privacy in wireless networks: (i) We develop an analytical model of the trust-privacy tradeoff, (ii) show that data-centric trust can reduce, with respect to entity-centric trust, the amount of private information traded for trust and (iii) optimize the trust-privacy tradeoff under the assumption of privacy-preserving entities that rationally minimize their privacy loss. Using game-theoretic models, we show that individual players do not contribute to trust establishment, unless they receive appropriate incentives. We believe that explicitly modeling the notion of privacy-oriented rationality will shed additional light on the appropriate mechanisms, such as incentives, for building privacy-preserving systems. We also hope that this work is only the beginning of an effort to bridge privacy and game theory, two fields that share their main focus: realizing the preferences of the human in the loop.

Appendix

6.A Proof of Theorem 6.5.1

To compute the equilibrium of $\mathbf{G_{AD}}$, we will find the best-response strategies of each macroplayer, i.e., the strategies maximizing their payoffs.

Using the payoffs and probabilities in Fig. 6.5, if D plays W, the payoffs of A are $p_A(v_A-c)$ if it plays W and $p_A(v_A-c)+p_D(-c)=p_Av_A-c$ if it plays S. As $p_A(v_A-c)>p_Av_A-c$, A plays W. If D plays S, the payoffs of A are $p_A(v_A-c-\delta)$ if it plays W and $p_Av_A-c-\delta$ if it plays S. Hence A plays W. Given that A will always play W, the payoffs of D are $p_D(v_D-c)$ if it plays W and p_Dv_D-c if it plays S. Hence D plays W. The PBE is thus (W, WW). □

6.B Proof of Theorem 6.5.2

To compute the equilibrium of $\mathbf{G^I_{AD}}$, we proceed similarly to the proof above. If D plays W, the payoffs of A are $p_A(v_A-c)$ if it plays W and p_Av_A-c+r if it plays S. Hence, if $p_A(v_A-c)\geq p_Av_A-c+r \Leftrightarrow r\leq p_Dc$, A plays W and otherwise it plays S. Consequently, the payoff of D is $p_D(v_D-c)$ if $r\leq p_Dc$ and $p_D(v_D-c-\delta)$ otherwise. Similarly, if D plays S, the payoffs of A are $p_A(v_A-c-\delta)$ if it plays W and $p_Av_A-c-\delta+r$ if it plays S. Hence, if $p_A(v_A-c-\delta)\geq p_Av_A-c-\delta+r \Leftrightarrow r\leq p_D(c+\delta)$, A plays W and otherwise it plays S. The payoff of D is thus p_Dv_D-c+r if $r\leq p_D(c+\delta)$ and $p_D(v_D-2\delta)-c+r$ otherwise.

Based on the above, we will consider all possible conditions on r with respect to the other parameters. If $r\leq p_Dc \Rightarrow r<p_D(c+\delta)$, D has a choice between two payoffs: $p_D(v_D-c)$ if it plays W and p_Dv_D-c+r if it plays S. If $p_D(v_D-c)\geq p_Dv_D-c+r \Rightarrow r\leq(1-p_D)c=p_Ac$, D plays W (because the corresponding payoff is higher) and otherwise it plays S. A plays W in both cases as $r\leq p_Dc$.

By applying similar reasoning, if $r>p_D(c+\delta)$, the possible payoffs for D are $p_D(v_D-c-\delta)$ if it plays W and $p_D(v_D-2\delta)-c+r$ if it plays S. Hence, if $p_D(v_D-c-\delta)\geq p_D(v_D-2\delta)-c+r \Rightarrow r\leq p_Ac+p_D\delta$, D plays W and otherwise S. A plays S in both cases.

Finally, if $p_Dc<r\leq p_D(c+\delta)$, D can choose between two payoffs: $p_D(v_D-c-\delta)$ if it plays W and p_Dv_D-c+r if it plays S. If $p_D(v_D-c-\delta)\geq p_Dv_D-c+r \Rightarrow r\leq p_Ac-p_D\delta$, D plays W and A plays S, otherwise D plays S and A plays W.

By regrouping the conditions in the last three paragraphs, we can derive Theorem 6.5.2. □

6.C Proof of Theorem 6.5.3

To find the SPE of $\mathbf{G_{TC}}$, we can apply a technique called *backward induction* that goes backwards starting from plausible outcomes at the leaves of the game tree. This allows eliminating Nash equilibria that are unreachable in practice (called incredible threats). In $\mathbf{G_{TC}}$, the best response of any entity is the one that maximizes its payoff given the contributions of the other entities. It is obvious that the value of t_k that maximizes (6.11) is 0. □

6.D Proof of Theorem 6.5.4

Similarly to the above proof, we use backward induction to solve Theorem 6.5.4. $BR_k(t_{j\neq k})$, the best response of entity u_k to $t_{j\neq k}$, solves the following problem based on (6.12):

$$\pi_k(t_1,\ldots,BR_k(t_{j\neq k}),\ldots,t_K) = \max_{t_k}\left(\frac{v_D}{K} + r\frac{t_k}{\sum_{i=1}^{K} t_i} - \frac{t_k}{\phi}\right) \tag{6.19}$$

As π_k is a strictly concave function with respect to t_k, it has a maximum that we can find by equating its partial derivative, with respect to t_k, to 0:

$$BR_k(t_{j\neq k}) = \sqrt{\phi r \sum_{j\neq k} t_j} - \sum_{j\neq k} t_j \tag{6.20}$$

Let t_k^* denote the best response of u_k at equilibrium (i.e., all entities play their best-response strategies):

$$t_k^* = \sqrt{\phi r \sum_{j\neq k} t_j^*} - \sum_{j\neq k} t_j^* \tag{6.21}$$

$$\Rightarrow \forall i,k \in \{1,\ldots,K\}, \quad \sum_{j\neq k} t_j^* = \sum_{j\neq i} t_j^* \tag{6.22}$$

$$\Rightarrow \forall i,k \in \{1,\ldots,K\}, \quad t_k^* = t_i^* \tag{6.23}$$

$$\Rightarrow K t_k^* = \sqrt{\phi r (K-1) t_k^*} \tag{6.24}$$

This results in the value of t_k^* in Theorem 6.5.4. □

Chapter 7

Conclusion

Communication technologies constantly evolve, entailing necessary adaptations in their components. Being a fundamental enabler of communications (of all kinds), trust should be one of the first components to adapt. In this work, we identify a new generation of communication networks where devices interact shortly and call these networks ephemeral. We also argue that in the near future, mobile devices will reflect to a large extent the personalities of their respective owners, thus fundamentally changing the way these devices interact. One of the major shifts will be a strong focus on self-optimization, or rationality in human terms, thus rendering many of the existing communication protocols inefficient.

This work constitutes one of the first steps towards building security mechanisms for the ephemeral and rational environments described above. Mark Twain said: "What a good thing Adam had. When he said a good thing he knew nobody had said it before." The following are the novel contributions of this thesis.

In Chapters 2 and 3, we describe a security and privacy architecture for VANETs, our running example of ephemeral networks. We cover topics including a list of new attacks, a privacy-preserving pseudonym-changing mechanism, and a set of revocation protocols.

In Chapter 4, we set up an analytical model for comparing revocation mechanisms in ad hoc networks, thus far an untouched area in the network security community. The keystone of our model is the assumption of costly revocation, which becomes particularly relevant if the entities participating in a revocation protocol are rational. Our game-theoretic model actually shows that none of the existing solutions consistently outperforms the others. Using this insight, we develop RevoGame, a hybrid protocol that performs better than any of its ancestors. This result encourages further application of economic tools to security problems, a trend that is currently gaining momentum among security researchers.

In Chapter 5, we formally define the notion of data-centric trust. Although the literature contains works describing heuristic plausibility checks for data verification, our framework is a systematic approach to evaluating the trustworthiness of data. Existing heuristic mechanisms can easily fit in this framework. In addition, we compare several methods for evaluating data-centric trust and spot the potentially negative effects of high uncertainty in data. We also provide a solution to this problem by applying the Dempster-Shafer Theory to the cases of high uncertainty. This actually highlights the need for establishing more bridges between the data fusion and networking communities.

In Chapter 6, we show how data-centric trust can reduce the unavoidable privacy loss resulting from trust establishment. We develop a simple analytical model for converting private

information to trust and quantify, although in an example scenario, the privacy improvement brought by making trust establishment data-centric rather than entity-centric. We also analyze with game theory how and when privacy-preserving entities trade their privacy for trust. Our analysis leads to the conclusion that proper incentives are needed to make the tradeoff possible. This sets the ground for designing privacy protocols suitable for the "humanized" wireless devices that will dominate the future.

Directions for Future Work

Dear reader, as you have probably concluded from the above summary, we have only touched the surface of a burgeoning research area. We hope that we have also laid down important blocks for designing security and privacy solutions for the next generation of wireless networks. There are many research directions that can be followed beginning from this work. Hereafter are some pointers:

- The models we developed in this thesis can be extended in several ways. For example, we only mention the possibility of using reputation information in our mechanisms. This possibility can be materialized by introducing repeated encounters among entities and consequently designing repeated games. Moreover, the game-theoretic models in Chapters 4 and 6 can be applied to other problems in security and privacy.

- We chose VANETs to be our ephemeral environment of choice. But the concepts we developed have a broader scope of application. For example, online environments have become ephemeral as the number of users interacting with each other (e.g., through online forums, blogs, and social networks) has exploded in the last few years. A particularly interesting feature of online environments is that the rational agents in this case are actually human, which implies a more complicated agent model than that of a computer.

- If data-centric trust is applied to online interactions, this will provide a testbed for validating the ideas we elaborated in this work. As VANETs only exist in standardization committees and simulations, we could not perform a real, experimental validation of our mechanisms. Feedback from real tests will definitely have an impact on our theoretical and simulation results.

- Among the things we show in this thesis is that using tools from other disciplines (e.g., economics) can positively affect the design of security and privacy protocols. One such source of inspiration could be nature with its myriad of defense mechanisms that have taken their time to evolve. This is also an area relatively unexplored thus far by the security community. For example, the kind of decision problems we address in this thesis are common for bees and ants [6]. Hence, we believe that security mechanisms can benefit from the blueprints drawn by nature.

Bibliography

[1] IBM 4764 PCI-X Cryptographic Coprocessor. http://www-03.ibm.com/security/cryptocards/pcixcc/overhardware.shtml.

[2] Trusted Platform Module (TPM). https://www.trustedcomputinggroup.org/groups/tpm/.

[3] http://trans.epfl.ch.

[4] Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems – 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ASTM E2213-03*, 2003.

[5] IEEE P1609.2 Version 1 - Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. *In development*, 2006.

[6] Decisions, decisions. *The Economist*, Feb. 2009.

[7] M. Abdel-Aty, A. Pande, C. Lee, V. Gayah, and C. Dos Santos. Crash risk assessment using intelligent transportation systems data and real-time intervention strategies to improve safety on freeways. *Journal of Intelligent Transportation Systems*, 11(3):107–120, Jul. 2007.

[8] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of PODC'06*.

[9] A. Acquisti, R. Dingledine, and P. Syverson. On the economics of anonymity. In *Proceedings of FC'03*, volume 2742 of *LNCS*, pages 439–443.

[10] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov. Cryptographic processors–A survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

[11] R. Anderson and T. Moore. The economics of information security. *Science*, 314(5799):610–613, Oct. 2006.

[12] G. Arboit, C. Crepeau, C.R. Davis, and M. Maheswaran. A localized certificate revocation scheme for mobile ad hoc networks. *Ad Hoc Networks*, 6(1):17–31, January 2008.

[13] D. Ariely. *Predictably Irrational*. HarperCollins, 2008.

[14] K. Bhaduri, K. Das, and H. Kargupta. Peer-to-peer data mining, privacy issues, and games. In *Proceedings of AIS-ADM'07*, volume 4476 of *LNCS*, pages 1–10.

[15] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[16] J. Blum and A. Eskandarian. The threat of intelligent collisions. *IT Professional*, 6(1):24–29, Jan.-Feb. 2004.

[17] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing (extended abstract). In *Proceedings of CCS'07*.

[18] D. Boneh. A brief look at pairings based cryptography. In *Proceedings of FOCS'07*.

[19] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proceedings of EUROCRYPT'03*, volume 2656 of *LNCS*, pages 416–432.

[20] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[21] S. Buchegger and J.-Y. Le Boudec. A robust reputation system for P2P and mobile ad-hoc networks. In *Proceedings of P2PEcon'04*.

[22] S. Buchegger and J.-Y. Le Boudec. Self-policing mobile ad hoc networks by reputation systems. *Communications Magazine, IEEE*, 43(7):101–107, 2005.

[23] J. Burgess, G. D. Bissias, M. Corner, and B. N. Levine. Surviving attacks on disruption-tolerant networks without authentication. In *Proceedings of MobiHoc'07*.

[24] L. Buttyan and J.-P. Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge University Press, 2007. http://secowinet.epfl.ch.

[25] X. Lin P.-H. Ho C. Zhang, R. Lu and X. Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *Proceedings of InfoCom'08*.

[26] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P. Hubaux. Efficient and robust pseudonymous authentication in VANETs. In *Proceedings of VANET'07*.

[27] S. Capkun, L. Buttyan, and J.P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.

[28] H. Chan, V. D. Gligor, A. Perrig, and G. Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 2(3):233–247, 2005.

[29] T. M. Chen and V. Venkataramanan. Dempster-Shafer Theory for intrusion detection in ad hoc networks. *IEEE Internet Computing*, 9(6):35–41, Nov.–Dec. 2005.

[30] B. Cobb and P. Shenoy. On the plausibility transformation method for translating belief function models to probability models. *International Journal of Approximate Reasoning*, 41(3):314–330, Apr. 2006.

[31] C. Crépeau and C. Davis. A certificate revocation scheme for wireless ad hoc networks. In *Proceedings of SASN'03*.

[32] R. Dingledine and P. Syverson. Reliable mix cascade networks through reputation. In *Proceedings of FC'02*, volume 2357 of *LNCS*, pages 253–268.

[33] Y. Dodis and T. Rabin. *Algorithmic Game Theory*, chapter Cryptography and Game Theory, pages 181–206. Cambridge University Press, 2007.

[34] S. Duri, M., X. Liu, P. Moskowitz, R. Perez, M. Singh, and J.-M. Tang. Framework for security and privacy in automotive telematics. In *Proceedings of the International Workshop on Mobile Commerce'02*.

[35] W. Enkelmann. FleetNet - applications for inter-vehicle communication. In *Proceedings of the IEEE Intelligent Vehicles Symposium'03*.

[36] L. Eschenauer, V. D. Gligor, and J. Baras. On trust establishment in mobile ad hoc networks. In *Proceedings of the International Security Protocols Workshop'02*.

[37] L. Buttyan M. Muter E. Schoch B. Wiedersheim T.-V. Thong G. Calandriello A. Held A. Kung J.-P. Hubaux F. Kargl, P. Papadimitratos. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):110–118, Nov. 2008.

[38] K. Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of SIGCOMM'03*.

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

[40] S. Ganeriwal and M. Srivastava. Reputation-based framework for high integrity sensor networks. In *Proceedings of SASN'04*.

[41] M. Gerlach, A. Festag, T. Leinmüller, G. Goldacker, and C. Harsch. Security architecture for vehicular communication. In *Proceedings of WIT'07*.

[42] V. Gligor, S. Luan, and J. Pato. On inter-realm authentication in large distributed systems. In *Proceedings of the IEEE Symposium on Security and Privacy'92*.

[43] L. Gollan and C. Meinel. Digital signatures for automobiles. In *Proceedings of Systemics, Cybernetics and Informatics (SCI)'02*.

[44] P. Golle, D. Greene, and J. Staddon. Detecting and correcting malicious data in VANETs. In *Proceedings of VANET'04*.

[45] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, Jan. 1997.

[46] J. Halpern and V. Teague. Rational secret sharing and multiparty computation (extended abstract). In *Proceedings of STOC'04*.

[47] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T.T. Abdelzaher. PDA: Privacy-preserving data aggregation in wireless sensor networks. In *Proceedings of InfoCom'07*.

[48] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, 2002.

[49] Y.-C. Hu, A. Perrig, and D. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. In *Proceedings of MobiCom'02*.

[50] Y.-C. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of InfoCom'03*.

[51] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security and Privacy Magazine*, 2(3):49–55, May-June 2004.

[52] M. Jakobsson and S. Wetzel. Efficient attribute authentication with applications to ad hoc networks. In *Proceedings of VANET'04*.

[53] T. Jiang and J.S. Baras. Trust evaluation in anarchy: A case study on autonomous networks. In *Proceedings of InfoCom'06*.

[54] A. Jøsang. An algebra for assessing trust in certification chains. In *Proceedings of NDSS'99*.

[55] R. Jurca and B. Faltings. Collusion-resistant, incentive-compatible feedback payments. In *Proceedings of EC'07*.

[56] J. Katz. Bridging game theory and cryptography: Recent results and future directions. In *Proceedings of TCC'08*, volume 4948 of *LNCS*, pages 251–272.

[57] L.A. Klein. *Sensor Technologies and Data Requirements for ITS Applications*. Artech House Publishers, 2001.

[58] R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *Proceedings of PKC'00*, volume 1751 of *Lecture Notes in Computer Science*, pages 93–112. Springer-Verlag.

[59] G. Kol and M. Naor. Games for exchanging information (extended abstract). In *Proceedings of STOC'08*.

[60] J. Kong, H. Luo, K. Xu, D. Gu, M. Gerla, and S. Lu. Adaptive security for multilevel ad hoc networks. *Wireless Communications and Mobile Computing*, 2(5):533–547, 2002.

[61] B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: theory and practice. *SIGOPS Oper. Syst. Rev.*, 25(5):165–182, 1991.

[62] T. Leinmüller, E. Schoch, and F. Kargl. Position verification approaches for vehicular ad hoc networks. *IEEE Wireless Communications*, 13(5):16–21, October 2006.

[63] L. Lilien and B. Bhargava. *Trading Privacy for Trust in Online Interactions*. Idea Group, 2008.

[64] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen. Security in vehicular ad hoc networks. *IEEE Communications Magazine*, 46(4):88–95, Apr. 2008.

[65] M. Lott, R. Halfmann, E. Schultz, and M. Radimirsch. Medium access and radio resource management for ad hoc networks based on UTRA TDD. In *Proceedings of MobiHoc'01*.

[66] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Transactions on Networking*, 12(6):1049–1063, December 2004.

[67] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Experiences applying game theory to system design. In *Proceedings of PINS'04*, 2004.

[68] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MobiCom'00*.

[69] P. B. Miltersen, J.B. Nielsen, and N. Triandopoulos. Privacy-enhancing first-price auctions using rational cryptography. Cryptology ePrint Archive, Report 2008/418, 2008. http://eprint.iacr.org/.

[70] A. Mishra, K. Nadkarni, and A. Patcha. Intrusion detection in wireless ad hoc networks. *IEEE Wireless Communications*, 11(1):48–60, Feb. 2004.

[71] T. Moore, J. Clulow, S. Nagaraja, and R. Anderson. New strategies for revocation in ad-hoc networks. In *Proceedings of ESAS'07*.

[72] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, and J.-P. Hubaux. Fast exclusion of errant devices from vehicular networks. In *Proceedings of SECON'08*.

[73] J. Mundinger and J.-Y. Le Boudec. Reputation in self-organized communication systems and beyond. In *Proceedings of Inter-Perf'06 (Invited Paper)*.

[74] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proceedings of EC'99*.

[75] B. Ostermaier, F. Dotzer, and M. Strassberger. Enhancing the security of local danger warnings in VANETs - a simulative analysis of voting schemes. In *Proceedings of ARES'07*.

[76] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, Nov. 2008.

[77] P. Papadimitratos, L. Buttyan, J-P. Hubaux, F. Kargl, A. Kung, and M. Raya. Architecture for secure and private vehicular communications. In *Proceedings of ITST'07*.

[78] P. Papadimitratos, V. Gligor, and J.-P. Hubaux. Securing vehicular communications - assumptions, requirements, and principles. In *Proceedings of escar'06*.

[79] B. Parno and A. Perrig. Challenges in securing vehicular networks. In *Proceedings of HotNets'05*.

[80] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[81] D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231–257, Jun. 2002.

[82] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity – a proposal for terminology. In *Proceedings of PET'00*, volume 2009 of *LNCS*, pages 1–9.

[83] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of SASN'05*.

[84] M. Raya, P. Papadimitratos, V. Gligor, and J.-P. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *Proceedings of InfoCom'08*.

[85] M. Scott, N. Costigan, and W. Abdulwahab. Implementing cryptographic pairings on smartcards. In *Proceedings of CHES'06*, volume 4249 of *LNCS*, pages 134–147.

[86] J.-M. Seigneur and C.D. Jensen. Trading privacy for trust. In *Proceedings of iTrust'04*, volume 2995 of *LNCS*, pages 93–107.

[87] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[88] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[89] D. Shaw and W. Kinsner. Multifractal modelling of radio transmitter transients for classification. In *Proceedings of WESCANEX'97: Communications, Power and Computing*.

[90] C. Siaterlis and B. Maglaris. Towards multisensor data fusion for DoS detection. In *Proceedings of SAC'04*.

[91] Y. Sun, W. Yu, Z. Han, and K.J. Ray Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305–317, 2006.

[92] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, Feb. 2006.

[93] M. Wolf A. Weimerskirch and T. Wollinger. State of the art: Embedding security in vehicles. *EURASIP Journal on Embedded Systems*, 2007.

[94] P. Wohlmacher. Digital certificates: a survey of revocation methods. In *Proceedings of Multimedia'00*.

[95] R. Wright, P. Lincoln, and J. Millen. Efficient fault-tolerant certificate revocation. In *Proceedings of CCS'00*.

[96] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-vehicle safety messaging in DSRC. In *Proceedings of VANET'04*.

[97] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang. Security in mobile ad hoc networks: challenges and solutions. *Wireless Communications, IEEE*, 11(1):38–47, 2004.

[98] X. Yang, J. Liu, F. Zhao, and N. Vaidya. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Proceedings of MobiQuitous'04*.

[99] S. Yi and R. Kravets. MOCA: Mobile certificate authority for wireless ad hoc networks. In *Proceedings of PKI'03*.

[100] M. El Zarki, S. Mehrotra, G. Tsudik, and N. Venkatasubramanian. Security issues in a future vehicular network. In *Proceedings of European Wireless'02*.

[101] P. Zheng. Tradeoffs in certificate revocation schemes. *SIGCOMM Computing Communication Review*, 33(2):103–112, 2003.

[102] L. Zhou and Z. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[103] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas. Robust cooperative trust establishment for MANETs. In *Proceedings of SASN'06*.

Index

Maxim Raya

EPFL-IC-ISC-LCA
Station 14
CH-1015 Lausanne, Switzerland

Office phone: +41 21 693 26 48
E-mail: maxim.raya@epfl.ch
Web: http://people.epfl.ch/maxim.raya

Personal

Born in Kiev, Ukraine on May 12, 1979. Citizen of Lebanon

Languages: English (fluent), French (fluent), Russian (native), Arabic (native)

Education

June 2004 - June 2009, **PhD in Communication Systems**, EPFL
Thesis title: *Data-Centric Trust in Ephemeral Networks*
Thesis advisor: *Prof. Jean-Pierre Hubaux*

October 2002 - October 2003, **Doctoral School in Communication Systems**, EPFL

July 1999 - June 2002, **BEng in Computer and Communications Engineering**, AUB (American University of Beirut), Lebanon

Professional Experience

June 2004 - present, **Research and teaching assistant**, EPFL

Nov 2003 - May 2004, **Research engineer**, CSEM (Swiss Center for Electronics and Microtechnology)

Publications

- M. Raya, M.H. Manshaei, M. Félegyházi and J.-P. Hubaux, **Revocation Games in Ephemeral Networks**, *In Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2008
- M. Raya, P. Papadimitratos, V. D. Gligor, J.-P. Hubaux, **On Data-Centric Trust Establishment in Ephemeral Ad Hoc Networks**, *In Proceedings of the IEEE Conference on Computer Communications (InfoCom)*, 2008
- P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung and J.-P. Hubaux, **Secure vehicular communication systems: design and architecture**, *IEEE Communications Magazine*, Vol. 46, Nr. 11, pp. 100-109, 2008.

- T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, J-P. Hubaux, **Fast exclusion of errant devices from vehicular networks**, *In Proceedings of the IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008

- M. Piorkowski, M. Raya, A. Lugo, P. Papadimitratos, M. Grossglauser, J.-P. Hubaux, **TraNS: Realistic Joint Traffic and Network Simulator for VANETs**, *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 12, Nr. 1, pp. 31-33, 2008

- A. Wegener, M. Piorkowski, M. Raya, H. Hellbrück, S. Fischer, J.-P. Hubaux, TraCI**: An Interface for Coupling Road Traffic and Network Simulators**, *In Proceedings of the Communications and Networking Simulation Symposium (CNS)*, 2008

- M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, **Eviction of Misbehaving and Faulty Nodes in Vehicular Networks**, *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, Vol. 25, Nr. 8, pp. 1557-1568, 2007

- M. Raya and J.-P. Hubaux, **Securing Vehicular Ad Hoc Networks**, *Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks*, Vol. 15, Nr. 1, pp. 39-68, Oct. 2007

- J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, J.-P. Hubaux, **Mix-Zones for Location Privacy in Vehicular Networks**, *In Proceedings of the IEEE International Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007

- P. Papadimitratos, L. Buttyan, J-P. Hubaux, F. Kargl, A. Kung, M. Raya, **Architecture for Secure and Private Vehicular Communications**, *In Proceedings of the International Conference on ITS Telecommunications (ITST)*, 2007

- M. Raya, I. Aad, J.-P. Hubaux, A. El Fawal, **DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots**, *IEEE Transactions on Mobile Computing*, Vol. 5, Nr. 12, Dec. 2006

- M. Raya, P. Papadimitratos, J.-P. Hubaux, **Securing Vehicular Communications**, *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, Vol. 13, Nr. 5, pp. 8-15, Oct. 2006

- M. Raya, A. Aziz, J.-P. Hubaux, **Efficient secure aggregation in VANETs**, *In Proceedings of the ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, 2006

- M. Raya and J.-P. Hubaux, **The Security of Vehicular Ad Hoc Networks**, *In Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2005

- M. Raya and J.-P. Hubaux, **Security Aspects of Inter-Vehicle Communications**, *In Proceedings of the Swiss Transport Research Conference (STRC)*, 2005

- M. Raya, J.-P. Hubaux, I. Aad, **DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots**, *In Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004

Projects

- Co-design of the open-source framework TraNS (`http://trans.epfl.ch`) for realistic simulations of vehicular networks
- Design and development of the DOMINO system (`http://domino.epfl.ch`) that shows and fixes a vulnerability in the IEEE 802.11 (Wi-Fi) standard

Professional Activities

Technical Program Committee member: ACM VANET 2007, 2008, 2009

Reviewer: IEEE Transactions on Mobile Computing, IEEE Transactions on Vehicular Technology, IEEE JSAC, IEEE Communications Magazine, ACM MobiCom, ACM MobiHoc, ACM WiSec, ACM SenSys, ACM VANET, IEEE InfoCom, and other events