Leonardo A. Martucci

Identity and Anonymity in Ad Hoc Networks

Karlstad University Studies
2009:25

Leonardo A. Martucci. *Identity and Anonymity in Ad Hoc Networks*

Dissertation

Karlstad University Studies 2009:25
ISBN 978-91-7063-248-8
ISSN 1403-8099

Distribution:
Karlstad University
Division for Faculty of Economy, Communication and IT
Department of Department of Computer Science
SE–651 88 KARLSTAD
SWEDEN
+46 54 700 1000

www.kau.se

Printed at: Universitetstryckeriet, Karlstad 2009

À meu avô e minha avó
Moacyr e Dirce Martucci
pelo amor ao seu filho
Moacyr

À memória de meu avô e avó
Masashi e Takako Sakatsume
pelo amor à sua filha
Olga

"No, that is not quite it. Let's try again from the beginning, Adso. But I assure you, I am attempting to explain to you something about which I myself am not sure I possess the truth. . . ."

Brother William of Baskerville
— The Name of the Rose (1980)
Umberto Eco

Identity and Anonymity in Ad Hoc Networks

LEONARDO AUGUSTO MARTUCCI
Department of Computer Science, Karlstad University, Sweden

Abstract

In ad hoc networks every device is responsible for its own basic computer services, including packet routing, data forwarding, security, and privacy. Most of the protocols used in wired networks are not suitable for ad hoc networks, since they were designed for static environments with defined borders and highly specialized devices, such as routers, authentication servers, and firewalls.

This dissertation concentrates on the achievement of privacy-friendly identifiers and anonymous communication in ad hoc networks. In particular, the objective is to offer means for better anonymous communication in such networks. Two research questions were formulated to address the objective:

I. *How to design proper and trusted privacy-friendly digital identifiers to be used in ad hoc network environments?*

II. *How to provide anonymous communication in ad hoc networks and what is the performance cost in relation to the obtained degree of anonymity?*

To address the first research question we studied and classified the security and privacy threats, enhancements, and requirements in ad hoc networks and analyzed the need for privacy and identification. The analysis led us to the relationship between security, identification, and anonymous communication that we refer to as the "identity-anonymity paradox". We further identified the requirements for privacy-friendly identifiers and proposed the self-certified Sybil-free pseudonyms to address such requirements.

The second research question was addressed with the design and implementation of the Chameleon protocol, an anonymous communication mechanism for ad hoc networks. The performance of Chameleon was evaluated using a network simulator. The results were used to find out the trade-off between anonymity and performance in terms of the expected end-to-end delay.

The solutions proposed in this dissertation are important steps towards the achievement of better anonymous communications in ad hoc networks and complement other mechanisms required to prevent leaks of personal data.

Keywords: privacy, identity, anonymity, pseudonymity, Sybil attack, security, ad hoc, and computer networks.

Acknowledgments

You are now holding a copy of my doctoral dissertation. It is the result of my research studies, but I certainly wouldn't have been able to complete it without the collaboration and support of many others. I would not consider this work complete without proper acknowledging they who somehow contributed to it.

First, I have Prof. Simone Fischer-Hübner to thank for giving me the opportunity to pursue my doctoral studies under her supervision, for introducing me to the research field of computational privacy, and for providing advice and directions. I was privileged for having her as supervisor and also as a friend.

I would like to thank the European Network of Excellence FIDIS (Future of Identity in the Information Society), PRIME (Privacy and Identity Management in Europe), Newcom and Newcom++ (Network of Excellence in Wireless Communications) for the financial support.

I am thankful for have been given the chance of working and collaborating with so many wonderful people with whom I share not only the authorship of publications, but also problems, solutions, ideas and so many hours of interesting and fruitful discussions. They all have my deepest respect.

All my colleagues at the Department of Computer Science deserve credit for providing such a nice and friendly work environment. I also want to thank all present and former members of the Privacy and Security Group. It was an honor to be part of such fine group. I owe a special thanks to Stefan Lindskog and Christer Andersson for volunteering to review the dissertation, and also to my secondary advisor Thijs J. Holleboom. Prof. Erland Jonsson and my friends from Chalmers University of Technology deserve my most sincere gratitude. I am also thankful to the opponent of my licentiate thesis Mats Näslund.

I am lucky and grateful for having made such good friends during my stay in Karlstad. Torbjörn Andersson, Ximena Dahlborn, Johan Eklund, Peter Dely, Ulf Larson, Stefan Lindskog, Christian Becker, Ge Zhang and Stefan Berthold definitely deserve a special mention. I also would like to thank my dear friends Marcel Castro and Roberta Agostini. I am especially fortunate to have Christer Andersson as an incredible friend, with whom I have shared not only the office and publications, but also very good times. Further, I am much obliged to my dearest Albin and Linda Zuccato for all the joy and fun we had together.

My brother Daniel and my everlasting friends Paulo de Andréa, Bidu, Bruno Galiotto, Luciano Maciel and Fernando Szterling deserve a big thank you for knowing me for so long and, strangely enough, still seem to enjoy having me around. And the most that I can do for you is to be your friend.

All my gratitude goes to my parents, d.Olguinha and s.Moacyr, for their unconditional support and love. For everything, I have only you to thank.

Karlstad, May 2009 Leonardo A. Martucci

List of Publications

Parts of this dissertation include material that have appeared in peer-reviewed papers, technical reports, book chapters and edited books. These publications, and other publications that I, Leonardo A. Martucci, authored, co-authored, or edited are listed next:

I. Ge Zhang, Simone Fischer-Hübner, Leonardo A. Martucci, and Sven Ehlert. Revealing the Calling History on SIP VoIP Systems by Timing Attacks. In *Proceedings of the 4th International Conference on Availability, Reliability and Security (ARES 2009)*, IEEE Computer Society, pages 135–142. Fukuoka, Japan, 16–19 Mar, 2009.

II. Christer Andersson, Markulf Kohlweiss, Leonardo A. Martucci, and Andriy Panchenko. A Self-Certified and Sybil-Free Framework for Secure Digital Identity Domain Buildup. In Jose A. Onieva, Damien Sauveron, Serge Chaumette, Dieter Gollmann, and Konstantinos Markantonakis, editors *Proceedings of the 2nd IFIP WG 11.2 International Workshop (WISTP 2008) Information Security Theory and Practices: Smart Devices, Convergence and Next Generation Networks*, Springer LNCS 5019, pages 64–77, Seville, Spain, 13–16 May, 2008.

III. Leonardo A. Martucci, Markulf Kohlweiss, Christer Andersson, and Andriy Panchenko. Self-Certified Sybil-Free Pseudonyms. In *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec'08)*, ACM Press, pages 154–159. Alexandria, VA, USA, Mar 31– Apr 2, 2008.

IV. Simone Fischer-Hübner, Dogan Kesdogan, and Leonardo A. Martucci. Privacy and Privacy-Enhancing Technologies. In Steven M. Furnell, Sokratis Katsikas, Javier Lopez, and Ahmed Patel, editors, *Securing Information and Communication Systems: Principles, Technologies, and Applications*, chapter 11, pages 213–242. Artech House Publishers, Boston, MA, USA, Apr 2008.

V. Christer Andersson, Leonardo A. Martucci, and Simone Fischer-Hübner. Privacy and Anonymity in Mobile Ad Hoc Networks. In Yang Zhang, Jun Zheng, and Miao Ma, editors, *Handbook of Research on Wireless Security*, volume 2, chapter 27, pages 431–448. Information Science Reference, USA, Mar 2008.

This book chapter also appears in:

- Christer Andersson, Leonardo A. Martucci, and Simone Fischer-Hübner. Privacy and Anonymity in Mobile Ad Hoc Networks. In David Taniar, editor, *Mobile Computing: Concepts, Methodologies, Tools,*

and Applications, volume 5, chapter 7.9. Information Science Reference, USA, Nov 2008.

VI. Simone Fischer-Hübner, Penny Duquenoy, Albin Zuccato, and Leonardo A. Martucci, editors, *Proceedings of the IFIP WG9.2, 9.6/11.7, 11.6 3rd International Summer School - The Future of Identity in the Information Society*, 461 pages. Springer, USA, Jun 2008.

VII. Leonardo A. Martucci, Albin Zuccato, and Simone Fischer-Hübner. Identity Deployment and Management in Wireless Mesh Networks. In Simone Fischer-Hübner, Penny Duquenoy, Albin Zuccato, and Leonardo A. Martucci, editors, *Proceedings of the IFIP WG9.2, 9.6/11.7, 11.6 3rd International Summer School - The Future of Identity in the Information Society*, pages 223–233, Springer. Karlstad, Sweden, 6–10 Aug, 2007.

VIII. Stefan Lindskog, Hans Hedbom, Leonardo A. Martucci, and Simone Fischer-Hübner. Experiences from Educating Practitioners in Vulnerability Analysis. In Lynn Futcher and Ronald Dodge, editors, *Proceedings of the 5th World Conference on Information Security Education (WISE'5)*, Springer, pages 73–80. West Point, NY, USA. 19–21 Jun, 2007.

IX. Leonardo A. Martucci, Christer Andersson, and Simone Fischer-Hübner. Chameleon and the Identity-Anonymity Paradox: Anonymity in Mobile Ad Hoc Networks. In *Proceedings of the 1st International Workshop on Security (IWSEC 2006) - Short Papers*, pages 123–134. Kyoto, Japan, 23–24 Oct, 2006.

Part of this paper summarizes results reported in:

- Leonardo A. Martucci. Identification in Mobile Ad Hoc Networks. In Denis Royer, editor, *FIDIS Section 6.1.1 from Deliverable 11.1: Collection of Topics and Clusters of Mobility and Identity - Towards a Taxonomy of Mobility and Identity*, 9 Jun, 2006.
- Leonardo A. Martucci, Christer Andersson, and Simone Fischer-Hübner. Towards Anonymity in Mobile Ad Hoc Networks: The Chameleon Protocol and its Anonymity Analysis. In *Karlstad University Studies 2006:35*, Karlstad University, Sweden, Aug 2006.

X. Leonardo A. Martucci, Christer Andersson, Wim Schreurs, and Simone Fischer-Hübner. Trusted Server Model for Privacy-Enhanced Location Based Services. In Viiveke Fåk, editor, *Proceedings of the 11th Nordic Workshop on Secure IT-systems (NordSec 2006)*, pages 13–25. Linköping, Sweden, 19–20 Oct, 2006.

XI. Leonardo A. Martucci, Hans Hedbom, Stefan Lindskog, and Simone Fischer-Hübner. Educating System Testers in Vulnerability Analysis: Laboratory Development and Deployment. In Cynthia Irvine, Matthew Rose and Naomi Falby, editors, *Practical and Experimental Approaches to Information Security Education, Proceedings of the 7th Workshop on Education in Computer Security (WECS7)*, pages 51–65. Monterey, CA, USA, 4–6 Jan, 2006.

XII. Christer Andersson, Leonardo A. Martucci, and Simone Fischer-Hübner. Requirements for Privacy-Enhancements in Mobile Ad Hoc Networks. In Armin B. Cremers, Rainer Manthey, Peter Martini, and Volker Steinhage, editors, *3rd German Workshop on Ad Hoc Networks (WMAN 2005), Proceedings of INFORMATIK 2005 - Informatik LIVE! Band 2*, pages 344–348. Lecture Notes in Informatics (LNI), Volume P-68, Gesellschaft für Informatik (GI), Bonn, Germany, 19–22 Sep, 2005.

This paper extends results reported in:

- Leonardo A. Martucci. Comparison of Anonymous Communication Mechanisms for Ad Hoc Networks. In Günter Müller and Sven Wohlgemuth, editors, *FIDIS Section 5.3 from Deliverable 3.3: Study on Mobile Identity Management*, 9 May, 2005.

XIII. Leonardo A. Martucci, Tereza C. Carvalho, and Wilson V. Ruggiero. A Lightweight Distributed Group Authentication Mechanism. In Steven M. Furnell and Paul S. Dowland, editors, *Proceedings of the 4th International Network Conference (INC 2004)*, pages 393–400. Plymouth, Devon, United Kingdom, 6–9 Jul, 2004.

This paper extends results reported in:

- Leonardo A. Martucci, Tereza C. Carvalho, and Wilson V. Ruggiero. Domínios Virtuais para Redes Móveis Ad Hoc. In *Proceedings of the 21st Brazilian Symposium on Computer Networks (SBRC 2003)*, pages 599–614. Natal, RN, Brazil, 19–23 May, 2003.

XIV. Leonardo A. Martucci, Christiane M. Schweitzer, Yeda R. Venturini, Tereza C. Carvalho, and Wilson V. Ruggiero. A Trust-Based Security Architecture for Small and Medium-Sized Mobile Ad Hoc Networks. In Ian F. Akyildiz, Erdal Cayirci, Eylem Ekici, and Giacomo Morabito, editors, *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2004)*, pages 278–290. Bodrum, Turkey, 27–30 Jun, 2004.

XV. Yeda R. Venturini, Christiane M. Schweitzer, Leonardo A. Martucci, Fernando F. Redigolo, Armin W. Mittelsdorf, Wilson V. Ruggiero, and Tereza

C. Carvalho. Security Model for Ad Hoc Networks. In *Proceedings of the 2002 International Conference on Wireless Networks (ICWN 2002)*, pages 185–191. Las Vegas, NV, USA, 24–27 Jun, 2002.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "No, that is not quite it. Let's try again from the beginning, Adso. But I assure you, I am attempting to explain to you something about which I myself am not sure I possess the truth. . . ."
>
> *Brother William of Baskerville*
> *— The Name of the Rose (1980), Umberto Eco*

Ubiquitous computing consists of computational environments providing information instantaneously through invisible interfaces[1], allowing unlimited spreading and sharing of information and offering an invaluable support for many aspects of the society and its institutions. This futuristic scenario is foreseen to be materialized with the advent of seamless communication networks combined with pervasive computing and natural human-computer interfaces, ultimately leading to an omnipresent distributed computing environment. These environments represent a paradigm shift from the current networking and computer models. However, the eventual realization of such environments is dependent on the development of new solutions and protocols.

The research presented here is focused on a single, but fundamental, core technology needed to enable ubiquitous computing: ad hoc networking. Ad hoc networks consist of computers, often mobile, that establish on demand network connections through their wireless interfaces, enabling instantaneous networking independently of the presence or aid of any central devices. Hence, ad hoc networks are decentralized computer networks. The upcoming of such networks requires critical changes in the current network infrastructure model, in which networks have defined borders and where basic network services, such

[1]The term invisible interface was coined at the Computer Science Laboratory at XEROX PARC. In this context, invisibility means that the technology, i.e., the user interface, should be only used as an enabler to the accomplishment of the task, and never as the tasks' centerpiece [Weiser, 1994].

as addressing, packet routing, data forwarding, security, and privacy are provided by dedicated devices.

In ad hoc networks every device is responsible for its own basic network services. Thus, most of the protocols employed in wired networks are not suitable for ad hoc networks since such protocols were designed for network environments with defined borders and highly specialized devices, such as routers, servers that provide network addresses, firewalls, and network intrusion detection systems. Moreover, such an absence of infrastructure potentially augments the risk of losing control over personal information since data is routed and forwarded through many unknown devices and users can easily be monitored. Hence, information regarding a user's communicating partners and even the contents of transmitted messages can be obtained by devices forwarding packets on the behalf of a user, if proper security measures are not implemented. Furthermore, data collection is especially not transparent in ubiquitous environments since invisible interfaces can greatly reduce the users awareness regarding when and what personal data is being collected by the ubiquitous environment. The scope of this dissertation involves two of those basic services for ad hoc networks: security and privacy.

The security mechanisms included in the wireless technology standards depend on the constant presence of centralized services deployed in the wired network and, thus, are not suitable for ad hoc networking, and consequently for ubiquitous computing. Moreover, the security mechanisms provided by central devices, usually located in the wired network, are restricted to the lower layers in the TCP/IP stack, such as the data link layer in IEEE 802.11 wireless networks. For instance, IEEE 802.11 security with support to RSNA[2] (Robust Secure Network Association) requires either the use of pre-shared keys or a RADIUS (Remote Authentication Dial-In User Service) server for authentication based on the port-based network access control standard IEEE 802.1X.

A similar underlying rationale is valid for privacy. Privacy can be obtained in computational environments with privacy enhancing technologies, such as anonymous communication mechanisms. The existing anonymous communication mechanisms available for wired networks are not suitable or directly applicable for ad hoc networks, since they usually rely either on a constant presence of centralized services, constant network traffic flow, or both, which implies traffic contention during periods when the amount of traffic is higher than the expected amount of traffic, or the usage of dummy, traffic, i.e., artificially created traffic, when this amount is below the expected. Relying on the assumption of the constant presence of a centralized service does not fulfill the requirements for an ad hoc network. Moreover, keeping a constant traffic flow in the network may compromise the overall network performance or shorten the device lifetime due to excessive transmissions of dummy traffic, consider-

[2]RSNA defines a number of security features in IEEE 802.11 networks.

ing that the majority of the wireless devices have a limited amount of battery power.

The implementation of security mechanisms can often result in a negative impact in the privacy properties, especially when authentication services require unique user identification when such information is not required to provide the requested service. Security and privacy are, however, two complementary features that if properly designed can be implemented beside one another in ad hoc networks. As discussed in this dissertation, the relationship between security and privacy is the need of trusted identifiers, i.e., identifiers that are unique in a sense that a user can have at most one digital identifier. Trusted identifiers are required to verify if users are actually who they claim to be. Trusted identifiers are also needed by privacy enhancing technologies, such as anonymous communication mechanisms, since users are anonymous within an anonymity set. The elements of such a set are all users that can be associated with a given action, such as sending a message. Every element of the anonymity set has thus to be associated to one, and only one, user. Otherwise, the degree of anonymity offered by privacy enhancing technologies can be seriously degraded by an attacker holding multiple identifiers. We have named the need of trusted identifiers to achieve anonymity as the identity-anonymity paradox.

The remainder of this chapter is organized in seven sections. Section 1.1 introduces the objectives and the scope of the dissertation. The terminology and background are presented in Section 1.2, and Section 1.3 outlines the research questions that are investigated in the dissertation. Section 1.4 describes the research methodology used to achieve the contributions of this dissertation, which are summarized in Section 1.5. Section 1.6 outlines the structure and briefly describes the contents of the following chapters. Finally, Section 1.7 clarifies the work done in collaboration.

1.1 Objective and Scope

The objective of this dissertation is to offer means for better anonymous communication in ad hoc network environments. The objective is divided into three goals. The first goal is to establish the connection between the need of trusted identifiers and the provisioning of anonymity. The second goal is to design and evaluate privacy-friendly identifiers that are suitable for ad hoc network environments. The third goal is to design and evaluate an anonymous communication mechanism for such environments.

The scope of the research presented includes identification, privacy-friendly identifiers, and anonymous communication mechanisms in ad hoc network environments.

1.2 Terminology and Background

The key terms and concepts used in this dissertation are defined in this section. The section is divided in three parts. The first part is related to the definition of privacy. The terms associated to identities and identifiers are outlined in the second part. Finally, the third part is related to the definition of computer and network security.

1.2.1 Privacy

The concept of privacy is not universal and easily defined, since its understanding is a cultural construct, and, hence, subjective. In the end of the 19^{th} century, two American lawyers defined privacy as the "right to be let alone" [Warren and Brandeis, 1890]. In 1967, Alan Westin, from Columbia University, wrote the most accepted definition of privacy: "the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others" [Westin, 1967]. The Universal Declaration of Human Rights states in its Article 12 that "no one shall be subjected to arbitrary interference with his privacy" [United Nations, 1949]. Nevertheless, the understanding of privacy changes significantly between different societies [Lunheim and Sindre, 1993]. Although it seems impossible to provide a precise and universal understanding of privacy, it is feasible to identify the three underlying aspects that construct the concept of privacy independently of the cultural background. These aspects of privacy are [Fischer-Hübner, 2001]: informational privacy, territorial (or spatial) privacy and privacy of the person.

Informational privacy is related to the person's right to determine when, how and to what extent information about him or her is communicated to others [Westin, 1967]. Territorial privacy refers to the ability of controlling the information that enters and leaves the personal sphere, i.e., the close physical area surrounding an individual. Finally, privacy of the person describes the people's right to be protected against physical undue interference. In this dissertation, we restrict the work to informational privacy aspects, and thus whenever the term privacy is used, it actually means informational privacy.

The principle of necessity of data collection and processing, which is one of the most essential privacy requirements, determines that the collection and processing of personal data should only be allowed if it is necessary for the tasks falling within the responsibility of the data processing agency. Hence, personal information should not be collected or used for identification purposes when not absolutely necessary. The best strategy to enforce such a requirement is the avoidance or minimization of personal data [Fischer-Hübner, 2001]. Therefore, privacy is best protected with anonymity.

Anonymity means that a subject is not identifiable within a set of subjects, i.e., the anonymity set. The anonymity set include all subjects that can be

connected to a given action. Anonymity can also be defined in terms of unlinkability. Unlinkability means that two or more items of interest, such as senders, recipients, and messages, cannot be related from the perspective of an attacker. Hence, sender anonymity means that it is not possible to associate a given message to a given sender. Recipient anonymity, on the other hand, means that it is not possible to associate a given message to a given recipient. Relationship anonymity means that sender and recipient are unlinkable [Pfitzmann and Hansen, 2008].

1.2.2 Identities and Identifiers

The definition of an identity is certainly a philosophical problem that will not be much discussed here. Nevertheless, such a discussion is important for understanding how identities are perceived and comprehended, and the remainder of this paragraph is reserved for some remarks on the philosophical aspects regarding identities. The definition of identities is philosophically related to the definition of the self and the concept of sameness. The self can be understood following different lines of thought, such as the Cartesian or the Aristotelian soul-body relations [Shields, 2008]. A physical entity can be mapped according to different conceptions of the self. Followers of the distinction of the self proposed by the French philosopher Ricœur [Ricœur, 1992] understand that digital identifiers are assigned to *idem* identities, but not to *ipse* identities[3]. On the other hand, following the philosophy of Cartesian dualism [Robinson, 2008], identifiers are assigned to bodies, which are physical entities, and not to souls or minds.

Digital identifiers are connected only to tangible forms and are not more than tags that aim to differentiate one object from another. A digital identifier is digital information that is linked to something in the physical world, such as a person, a group of persons, or a physical device, and it is necessarily stored in one or more physical devices. Thus, an identifier may identify the device that stores it or the user, or group of users that controls the device. A digital identifier exist in both abstract terms, as information, and concrete terms, as the raw data that is stored on a magnetic disk or a solid state memory, for instance. In any case, an identifier is an artifact used to identify an identity. Hence, we restrict the understanding of identities to a unique piece of information, i.e., an identifier, associated with one or more physical entities.

Proper and trusted digital identifiers are recognized and acknowledged by other devices that are part of the computer network. In privacy-friendly applications, identifiers should not be linkable to their owners, unless the owner of such an identifier performs a non-authorized action, for instance. Pseudonyms

[3]According to the philosopher Ricœur, the self's identity is divided into the *idem* and the *ipse* identities. The *idem* identity provides the self with its spatiotemporal sameness, while the *ipse*-identity gives the ability to initiate something new and imputable to the self [Dauenhauer, 2008].

are identifiers that can fulfill such requirement. A pseudonym is an identifier other than one of the real names of the pseudonym holder. There are five different classes of pseudonyms [Pfitzmann and Hansen, 2008] with respect to the degree of linkability and the purpose of use. These classes are briefly described below:

- *person pseudonyms* are just substitutes for the pseudonym holder's name and a representation for the holder's civil identity;
- *role pseudonyms* are identifiers that are used for a specific application or situation;
- *relationship pseudonyms* are identifiers that are used exclusively for the communication with a given communicating partner;
- *role-relationship pseudonyms* are a combination of role and relationship pseudonyms, i.e., a pseudonym exist for each role and for each communicating partner, and;
- *transaction pseudonyms* or one-time-use pseudonyms are identifiers that are used only one time, i.e., for each transaction performed, a new pseudonym is used.

The privacy-friendly identifiers proposed in this dissertation are pseudonyms. Such pseudonyms can be used in different applications and for different tasks. Therefore, the pseudonym class will depend on the purpose of the application. For instance, the proposed identifiers can be used as transaction pseudonyms in electronic voting applications or as role pseudonyms in applications that implement reputation schemes.

1.2.3 Computer and Network Security

The provisioning of network security is fundamental to keep the network infrastructure and the applications running on top of such an infrastructure continuously available to provide network services. Security services are of utmost importance in the provisioning of such network applications and they must be embed not only in the application layer, but also in the lower layers, such as the network layer. The ultimate role of network security is to provide a safe and reliable network environment.

Network security can be defined as the achievement of the five security services, i.e., authentication, confidentiality, integrity, non-repudiation, and access control, specified in the Telecommunications Standardization Sector of the International Telecommunications Union (ITU-T) Recommendation X.800 [ITUT X.800], along with the provisioning of availability [Bishop, 2004]. A standard definition of security services is found in the Internet Engineering Task Force

(IETF) RFC 4949 [Shirey, 2007], which describes a security service as: "a processing or communication service that is provided by a system to give a specific kind of protection to system resources".

1.3 Research Questions

The two research questions addressed in this dissertation involve the achievement of anonymous communications in decentralized computer network environments. The research questions are:

I. *How to design proper and trusted privacy-friendly digital identifiers to be used in ad hoc network environments?*

Addressing such a question required first to acknowledge the need of proper and trusted identifiers in ad hoc networks. Such an acknowledgment is drawn from what we refer to as the "identity-anonymity paradox", which establishes the relationship between security, identification, and anonymous communications. This paradox is further discussed in Chapter 3. This analysis is followed by the definition of the requirements for privacy-friendly identifiers, which are pointed out in Chapter 4. Self-certified Sybil-free pseudonyms are identifiers that address the aforementioned requirements. The design of such pseudonyms is presented and analyzed in Chapter5.

II. *How to provide anonymous communication in ad hoc networks and what is the performance cost in relation to the obtained degree of anonymity?*

The requirements for anonymous communication mechanisms in ad hoc networks were first investigated and listed. Such requirements are outlined in Chapter 4. It is followed by the design of an overlay mechanism that is situated in between the application and the transport layer. The aim of the overlay is to provide an anonymous communication mechanism for ad hoc network environments. The mechanism is designed to address this research question and is called Chameleon. Chameleon is further described in Chapter 6. The theoretical anonymity analysis of Chameleon is presented in Chapter 7, whereas its performance analysis is presented in Chapter 8. The anonymity and performance analysis are used to derive the cumulative distribution function for the expected end-to-end delay according to the projected resistance against malicious users.

1.4 Research Method

The scientific research method used during the research that led to this dissertation had the recurrent steps: literature study, problem statement, hypothesis

formulation, testing and evaluation, and conclusions. Such a research method is classified as deductive research, since hypotheses (or theories, according to the deductive research terminology) were proposed and afterwards tested in order to verify the validity of their claims [Chalmers, 1999]. Hypotheses testing and evaluation was either done with analytical methods or by simulation.

There are essentially three techniques for performance evaluation: analytical modelling, simulation, and measurement. The life-cycle stage of a system mostly determines the evaluation technique to be selected. Measurements are only possible if something similar to the proposed system already exists. That was not the case regarding the contributions of this dissertation. Furthermore, measurements may not give accurate results because of the environmental parameters and the time of measurement, which may be unique to the experiment. In general, new concepts are evaluated using analytical modelling or simulation [Jain, 1991]. Analytical modelling usually provides the best insight into effects of various parameters and their interactions. The drawback of analytical modelling is that it requires many simplifications and assumptions. Simulations allow searching the space of parameter values, which is hardly possible with measurements for instance. In addition, simulations can incorporate more details than analytical modelling, but usually take a long time to develop.

The literature study showed a lack of low-latency anonymous communication mechanisms in ad hoc networks that could be deployed in between the application and the transport layer, i.e., that were not connected to the ad hoc routing protocol. This absence of solutions led to the definition of the problem statement and of a set of requirements for anonymous communication mechanisms in ad hoc networks. The hypothesis was formulated with the design of an anonymous communication mechanism, the Chameleon protocol. The network performance of the Chameleon protocol was evaluated using analytical modelling and simulation. Both techniques were used together to verify and validate the results of each one, following one of the three rules of validation defined in [Jain, 1991]:

- do not trust the results of a simulation model until they have been validated by analytical modelling or measurements;
- do not trust the results of an analytical model until they have been validated by a simulation model or measurements;
- do not trust the results of measurements until they have been validated by analytical modelling or simulation.

The first research question presented in Section 1.3 was formulated after the identification of a conflict between the need and the absence of trusted identifiers in ad hoc network environments during the initial literature study. A hypothesis was formulated and called the identity-anonymity paradox, which was

evaluated with theoretical analysis. A set of requirements for privacy-friendly identifiers were defined, and they were followed by another step of literature study that resulted in the proposal of the self-certified Sybil-free pseudonyms. The privacy-related properties of such pseudonyms were evaluated with analytical methods. Analytical methods are appropriate for evaluating the security and privacy properties of such pseudonyms, since such properties cannot be evaluated using simulations, and no prototype of such pseudonyms was implemented, and, therefore, measurements were not possible to be collected.

Analytical methods were also used to evaluate the anonymity properties of the Chameleon protocol. Live measurements were not considered in the evaluation of the anonymity properties of the Chameleon protocol since no prototype was implemented.

1.5 Contributions

There are three major contributions of the dissertation: the identity anonymity paradox, the self-certified Sybil-free identifiers, and the Chameleon protocol. In this section we list the contributions and compare them with the related work.

The identification of the identity-anonymity paradox is the first contribution of the dissertation. It analyzes the intrinsic relationship between the need for trusted identifiers and the provisioning of anonymity in ad hoc networks, which is a fundamental starting point for the deployment of anonymous communication mechanisms in such networks. The complete absence of trusted identifiers was, so far, used as a mechanism for implementing anonymity in wireless networks. Logical and hardware addresses, i.e., IP and MAC addresses, are pseudonyms, and changing such information periodically may indeed enhance privacy in the data link and network layers [Gruteser and Grunwald, 2003], if such a change can obfuscate other sources of identification in the data link layer [Franklin et al., 2006]. However, such an approach does not provide some key privacy properties such as unlinkability between senders and receivers and sender anonymity towards the recipient. Moreover, Sybil attacks [Douceur, 2002] can easily be launched in ad hoc networks whose participants have no trusted identifiers. The identity-anonymity paradox explains that trusted identifiers are a fundamental building block to construct anonymity sets and to implement anonymity services. Such a conclusion is of uttermost importance in the design of anonymity services for ad hoc networks.

The self-certified Sybil-free pseudonyms are the second contribution of the dissertation. They are privacy-friendly identifiers that are locally produced from an initial trusted identifier obtained from a trusted third party during the system bootstrap phase. The pseudonym generation is executed without the presence or assistance from the trusted third party and cannot be linked back to their holders. Such pseudonyms have two intrinsic properties: unlink-

ability among the pseudonyms generated from a given initial trusted identifier, and detection of Sybil identifiers, which allows devices that are part of the anonymity set to detect a Sybil attack. The base for the instantiation of the self-certified Sybil-free pseudonyms are the periodic *n*-times spendable e-tokens [Camenisch et al., 2006], which were adapted in several ways to meet the aforementioned properties[4].

There are other proposals used for the construction of privacy-friendly identifiers. The combination of a group signature scheme, a centralized group key distribution scheme, and a distributed key-agreement scheme into a secure secret handshake can provide unlinkability, anonymous authentication, and detection of Sybil identifiers [Tsudik and Xu, 2006]. However, such an approach requires the continuous presence of a group controller, i.e., a trusted third party, for admitting new users into the group and for rekeying every time a new device joins the group. The self-certified Sybil-free pseudonyms do not have such requirements. Anonymous communication in ad hoc networks can also be implemented using a pseudonym-based encryption scheme based on pairings and constructed on top of an identity-based encryption scheme and short signatures from the Weil pairing [Huang, 2007]. Such scheme can generate pseudonyms without the presence of a trusted third party. However, the main disadvantage of such a scheme is that it is vulnerable to Sybil attacks. X.509 attribute certificates can be made privacy-friendly by assigning a pseudonym in the *holder* field instead of binding it directly to an identity certificate [Benjumea et al., 2007]. The drawback of such an approach is that it does not provide unlinkability between multiple shows of a same attribute certificate.

The third contribution of the dissertation is the design and evaluation of a low-latency overlay anonymous communication protocol for ad hoc networks. The proposed anonymous communication protocol, which is called Chameleon, operates in between the application layer and the transport layer. It is evaluated using theoretical analysis and simulation tools. Moreover, we also evaluate the anonymity and performance trade-off of such a protocol and identified the cumulative distribution function of the expected end-to-end delay and the amount of packet losses in relation to the degree of anonymity in an ad hoc network scenario. Other anonymous communication protocols have been proposed for ad hoc networks, mostly anonymous routing protocols, which operate at the network layer. A major disadvantage of embedding a privacy enhancing technology directly in the routing mechanism at the network layer is that it turns the solution incompatible with the standard ad hoc routing protocols, which may result in a reduced anonymity set. There is no such a drawback in solutions operating above the network layer, such as the Chameleon protocol. The Mix Route Algorithm (MRA) is an overlay anonymous communication mechanism that adapts the Chaumian mix concept to mobile ad hoc networks [Jiang

[4]The comprehensive list of adaptations is presented in Section 5.2.

et al., 2004]. MRA thus batches and reorders data traffic, and uses bandwidth-consuming dummy traffic between mixes, which can result in a high-latency, depending on the amount of traffic in the network. To the best of our knowledge, Chameleon was the first low-latency overlay anonymous communication protocol for ad hoc networks.

Other contributions of this dissertation are the summary of sources of device identification presented in Chapter 2 and the definition of the requirements for anonymous communication mechanisms and privacy-friendly identifiers in Chapter 4.

1.6 Structure

This dissertation is organized in nine chapters and one appendix. All chapters, excluding the last one, have a summary section that outlines the chapter's contents and briefly introduces the following chapter. Below, we provide a summary of the contents of the remainder chapters and present the connections between them.

- Chapter 2: Security and Privacy in Ad Hoc Networks.

  This chapter presents security and privacy threats and approaches that enhance security and privacy in ad hoc networks. It begins with an introduction to ad hoc networks and the security and privacy threats that can be identified in these networks. It also discusses some approaches to enhance security and privacy in ad hoc networks, and presents a taxonomy of security models for such networks. Furthermore, anonymous communication mechanisms in the context of ad hoc networks are presented and classified according to their functionality regarding their placement in the TCP/IP stack.

- Chapter 3: The Identity Anonymity Paradox.

  This chapter presents the problem of identification and authentication in ad hoc networks and its consequences to security and privacy. It revisits the definition of ad hoc networks and discusses the provisioning of addressing information in such networks. Moreover, it shows the connection between the absence of device identifiers in ad hoc networks and the Sybil attacks [Douceur, 2002]. In addition, this chapter discusses the relationship between the absence of identifiers and the provisioning of anonymity properties and presents the current countermeasures against Sybil attacks in ad hoc networks. Furthermore, it introduces the identity-anonymity paradox by presenting the relationship between security, the absence of identifiers and the provisioning of anonymous communications in ad hoc networks and its consequences.

- Chapter 4: Security and Privacy Requirements for Ad Hoc Networks.

  This chapter reviews the requirements for security and privacy in ad hoc networks. The objective of this chapter is to outline the security and privacy requirements that are used in the remainder of this dissertation. These requirements are used in the design of privacy-friendly identifiers for ad hoc networks in Chapter 5. Moreover, they are also used for defining the trade-offs between the offered degree of anonymity and the network performance parameters, such as end-to-end delay, for anonymous communication mechanisms that are suitable for ad hoc networks, such as Chameleon, which is presented in Chapter 6.

- Chapter 5: Self-certified Sybil-free Identifiers.

  This chapter presents a framework for the provisioning of identifiers that are bound to a group and are Sybil-free and self-certified, i.e., they are issued by the device that holds it and locally signed and supports the detection of a device that issues more than one identifier in a given group. The framework provides unlinkability between different identifiers issued to different groups by the same device. The objective of this chapter is to present the framework and the self-certified Sybil-free identifiers. Such identifiers may be used in the construction of the anonymity sets used in the Chameleon protocol.

- Chapter 6: The Chameleon Protocol.

  This chapter presents Chameleon, an overlay anonymous communication mechanism designed according to the requirements for anonymous communication mechanisms presented in Chapter 4. Chameleon is tailored for ad hoc environments and provides sender anonymity against recipients and relationship anonymity against local observers. In addition, Chameleon provides conditional anonymity against malicious Chameleon users, as well as protection against single attackers trying to compromise large portions of a network by assuming multiple identities. This chapter also presents the attacker model used in our evaluation of the Chameleon protocol.

- Chapter 7: Anonymity Analysis of the Chameleon Protocol.

  This chapter presents the anonymity analysis of the Chameleon protocol against the attacker model defined in the Chapter 6. The attacker model considered consists of the following five types of attackers: local observers, malicious insiders, malicious outsiders, destination devices, and malicious devices hosting a directory service. The following aspects of anonymity are evaluated: sender anonymity, receiver anonymity and relationship anonymity. Furthermore, the chapter outlines the metric used to measure anonymity.

- Chapter 8: Anonymity and Performance Trade-offs.

  This chapter evaluates the network performance of the Chameleon protocol and identifies the trade-off between anonymity and performance. A network simulator was used to simulate an ad hoc network. Simulation results include the amount of packets lost, and the extra transmission delay introduced by the Chameleon protocol running in an ad hoc environment and to isolate such delays from other transmission delays caused by the ad hoc routing protocol. The performance impact is simulated according to multiple values attributed to the probability of forwarding, which determines the degree of anonymity protection.

- Chapter 9: Final Remarks.

  This final chapter of this dissertation summarizes its contributions, discusses future directions and the presents the concluding remarks.

This dissertation also includes an appendix that presents the cryptographic foundations required to build the unlinkable and unique pseudonyms presented in Chapter 5.

1.7 Work and Collaboration

This section clarifies the work done in collaboration. The objective of this section is to point out that this dissertation is indeed a product of my research achievements but credit must definitely be given to those who contributed to the contents and results presented here.

The work presented in Section 4.2.1 was published in conjunction with Christer Andersson and Simone Fischer-Hübner in [Andersson et al., 2005b]. I contributed with the idea, and sketched the first version of the requirements for anonymous communications mechanisms and also did the initial evaluation of anonymous peer-to-peer communications mechanisms in the context of ad hoc networks. This initial version was later extended by Christer Andersson and me.

The work presented in Chapter 5 was done and published in conjunction with Markulf Kohlweiss, Christer Andersson, and Andriy Panchenko. Some parts of this chapter were published in [Martucci et al., 2008a] and in [Andersson et al., 2008a]. I was initially responsible for defining the problem and sketching a possible solution using anonymous credentials. The research team was created after my presentation in an interdisciplinary workshop where I presented the problem and indicated the direction of how this could be solved. The solution was later discussed and refined by the aforementioned research team. Markulf Kohlweiss contributed mostly with the cryptography protocols

which are presented in Section 5.2 and the underlying cryptography foundations, presented in the Appendix A.

Parts of the work presented in Chapters 6 and 7 were published in conjunction with Christer Andersson and Simone Fischer-Hübner. I was responsible for defining the problem and describing the initial sketch of the Chameleon protocol included in Chapter 6. Chameleon was later refined in collaboration with Christer Andersson and Simone Fischer-Hübner. Christer Andersson was mainly responsible for the anonymity analysis, to which Simone Fischer-Hübner and I also contributed.

In Chapter 8, I was responsible for defining the goals of the simulation, for outlining the simulation scenario and the network topology, for selecting the simulation platform, for implementing the Chameleon protocol into the simulator, for running the simulations, and for acquiring the simulation results. I also analyzed such results and defined how they should be presented. I was assisted by my co-supervisor Thijs Holleboom with the definition of the theoretical cumulative distribution function.

1.8 Summary

This chapter introduced this dissertation and its research topics. Moreover, it outlined the objectives and contributions of the dissertation and presented the motivation, the scope, and the research methods used. The structure of the dissertation and a summary of its chapters were also given.

In the next chapter security and privacy threats and approaches that enhance security and privacy in ad hoc networks will be presented.

Chapter 2

Security and Privacy in Ad Hoc Networks

"Then why do you want to know?"
"Because learning does not consist only of knowing what we must or we can do, but also of knowing what we could do and perhaps should not do."

Adso of Melk and Brother William of Baskerville
— The Name of the Rose (1980), Umberto Eco

This chapter presents security and privacy threats and approaches that enhance security and privacy in ad hoc networks. The chapter is divided into four sections. The first section presents an introduction to ad hoc networks. The second section introduces security and privacy threats that can be identified in these networks. The third section discusses some approaches to enhance security and privacy in ad hoc networks, including a taxonomy of security models for ad hoc networks. Finally, anonymous communication mechanisms in the context of ad hoc networks are presented and classified according to their functionality regarding their placement in the TCP/IP stack in the last section.

2.1 Introduction to Ad Hoc Networks

The prospect of having access to information anywhere at anytime pushes the popularity of wireless technologies. The dissemination of wireless data networks has been increasing since the first release of the IEEE 802.11 standard in 1999 [IEEE 802.11]. Figures regarding the wireless expansion are barely needed since the increase in the last decade of the amount of wireless hot spots available in public areas, such as airports, high-speed trains and hotels, is

easily noticeable. Wireless access points have become so popular that domestic wireless local area networks are a commonplace. In parallel, wireless personal network technologies based on IEEE 802.15 standards [IEEE 802.15.3; IEEE 802.15.1; IEEE 802.15.4], such as Bluetooth [Bluetooth], are widespread in high-end and even low-end mobile devices. Furthermore, the IEEE 802.16 [IEEE 802.16] standardized the technology for the last mile broadband wireless access. The scope of those standards encompasses personal, local and metropolitan area networks and provides a full range of wireless solutions for both enterprises and end-users. The growth and importance of the wireless market is undeniable with the upcoming new services specifically designed for wireless networks. Hence, the future expectations regarding the wireless market are certainly positive.

The aforementioned wireless standards were originally designed to operate in single-hop scenarios and in controlled environments, i.e., a wireless network with an access point or two or more wireless nodes that can communicate directly. The standards cover physical and data link aspects of wireless technologies. However, there is a large set of application scenarios that are not covered in single-hop wireless networks, such as sensor networks and vehicular networks. Those scenarios require multi-hop wireless networks with eventually highly dynamic topologies, on which wireless nodes may vanish and reappear in a different geographical locations. In addition, the absence of a fixed and online infrastructure might be part of some scenarios, such as military applications. Furthermore, the resources of a wireless node, e.g., battery and processing power, might be scarce. These wireless, multi-hop, and autonomous networks are set to perform a specific task and may disappear after the completion of this task. Thus, these networks were named *ad hoc*, which can be translated as *for a particular end or purpose* from Latin.

The RFC 2501 [Corson and Macker, 1999] lists some characteristics of ad hoc networks, such as dynamic topologies, bandwidth and energy (regarding battery power) constrains, variable capacity links, and limited physical security. In addition, this RFC also refers to operational modes for ad hoc networks: they may operate in isolation, or may have gateways to and interface with a fixed network.

Ad hoc networks have specific demands that are not fulfilled by protocols designed for networks that can rely on the continuous availability of a fixed and established online network infrastructure, such as the Internet. The services available in ad hoc network are instead defined by the devices that are part of it, e.g., a printing service may be available in an ad hoc network only if a device with printing capabilities is part of such a network and offers such a service. In addition, each device is also responsible for its own basic network services, such as packet routing, data forwarding, network addressing, security, and privacy. Furthermore, establishing an ad hoc network demands cooperation from the network participants.

Protocols designed for infrastructure dependent networks that rely on the continuous availability of online services, i.e., the Internet, are in general not suitable for ad hoc networks in general because of the dynamic, decentralized and sometimes unpredictable nature of ad hoc networks. Therefore, ad hoc networks need suitable protocols and solutions to be developed. Suitable protocols and solutions are clearly dependent on the application scenario because different application scenarios have different requirements and assumptions. Furthermore, there are numerous applications scenarios for ad hoc networks.

This section continues with an introduction to the most common ad hoc applications, their assumptions and their requirements, and is followed by a classification of ad hoc networks regarding assumptions on the availability of external services and the conditions of such an availability. This section ends with an introduction to routing in ad hoc networks followed by a summary of the security and privacy issues that exist in ad hoc networks in Section 2.2.

2.1.1 Applications, Assumptions and Requirements

Nearly all applications for ad hoc networks can be applied in both civilian and military environments. The most common application scenario for ad hoc networks is an ephemeron network that is set to address the current communication needs and it is established spontaneously by personal computers and mobile high-end devices [Feeney et al., 2001], e.g., mobile phone, laptops, in closed or open environments, such as meeting rooms, airport lounges or even battlefields.

Another scenario is an ad hoc network formed by many small, unassisted and inexpensive devices equipped with one or more sensors that are used to monitor environmental parameters of a given geographical location and send the gathered data to a collecting (sink) node, i.e., a so-called sensor network [Kahn et al., 2000; Perrig et al., 2004].

Ad hoc networks can be used by telecommunication providers to extend the radio range of wireless access points that are directly connected to their network infrastructure by using an ad hoc network of wireless relays (either mobile or stationary) that provide connectivity to users located far away from the first wireless access point, i.e., a wireless mesh network[1] [Glass et al., 2008; Martucci et al., 2008b].

Ad hoc network devices can also be installed in automobiles to set a vehicular ad hoc networks (vanet) that can be used either for communication between an automobile and a roadside access point or for exchanging data among cruising automobiles [Lin et al., 2008].

[1]The term wireless mesh network was coined in a 1995 survey on defence technologies published in The Economist magazine [Morton, 1995], but their view of a wireless mesh network fits better to what we call nowadays a sensor network.

Since the requirements and assumptions for the aforementioned application scenarios, i.e., sensor networks, wireless mesh networks, and vehicular ad hoc networks, differ a lot, each of these applications of ad hoc networks have their own set of proposed solutions and protocols especially when regarding security and privacy issues. Thus, sensor networks, wireless mesh networks and vehicular networks correspond each to a specific research area with its own special needs and requirements. The scope of this dissertation is ad hoc networks formed by personal computers and mobile high-end devices in open or closed environments.

2.1.2 Classification of Ad Hoc Networks

This section presents a classification of ad hoc networks regarding assumptions on the availability of external services and the conditions of such an availability. The classification presented in this section is used later as a support for a taxonomy of security models in ad hoc networks in Section 2.3 and also as a base for the discussion regarding identities in ad hoc networks in Chapter 3. The classification is a variant of a security taxonomy presented in [Martucci, 2006], and extends the classifications presented in [Merwe et al., 2007; Čapkun et al., 2006][2].

Regarding assumptions on the availability of external services and the conditions of such an availability, we classify ad hoc networks in the following three groups:

- *intermittently connected to an established infrastructure* — ad hoc networks that connect periodically (or occasionally) to an established infrastructure, such as the Internet. Therefore, it is possible to rely on some deployed services and infrastructure. Examples of such a scenario can be found in [Kargl et al., 2006; Montenegro and Castelluccia, 2002];

- *one or more privileged devices in the ad hoc network* — the assumption is that one or more devices have a special role in the network. There are two basic approaches to set one or more privileged devices in ad hoc networks:

  - one or more devices have special roles in the network, and can perform privileged or exclusive tasks on behalf of or as a service for other non-privileged devices, such as issuing identities to other network devices, aggregating data in a sensor network or distributing cryptographic keys. Examples of such a scenario can be found in [Balfanz et al., 2002; Martucci et al., 2004b; Stajano, 2001; Stajano and Anderson, 1999];

[2] Čapkun et al. [2006] and Merwe et al. [2007] divide ad hoc networks in two types: *fully self-organized* and the *authority-based*. The former type can be directly mapped to the third group of the classification presented in this section and the latter corresponds to the first two groups.

  - a set of privileged devices that have special roles in the network but with an additional constraint: a single privileged device cannot perform any privileged task by itself. Thus, it needs to cooperate with other similar devices to perform a privileged task, such as generating and distributing cryptographic keys. Examples of such a scenario can be found in [Luo et al., 2002; Zhou and Haas, 1999];

- *fully independent and self-organized ad hoc networks* — ad hoc networks that can operate in complete isolation from any online or offline infrastructure or central server. In this case, the ad hoc network is created solely by end-users [Merwe et al., 2007; Čapkun et al., 2006] in a flat hierarchical topology. Such networks are dependent upon the cooperation and trusting nature of the devices that form the network [Buttyán and Hubaux, 2003]. Examples of such a scenario can be found in [Buttyán and Hubaux, 2003; Hubaux et al., 2001; Čapkun et al., 2003a,b].

According to RFC 2501 [Corson and Macker, 1999], ad hoc networks *may* operate in isolation, or *may* have gateways to and interface with a fixed network (a stub ad hoc network). There is a gray zone in this definition regarding devices that may have occasional connectivity to an infrastructure. This can be argued as being the most common case for wireless devices.

Despite the definition of ad hoc networks in the RFC 2501, some researches still consider that ad hoc networks must be able to be formed and operate in complete isolation [Merwe et al., 2007]. However, in this dissertation we argue that fully independent and self-organized ad hoc networks have intrinsic problems regarding the provisioning of identification, security and privacy. This point is further discussed in Chapter 3.

2.1.3 Introduction to Routing in Ad Hoc Networks

One of the first problems studied in the context of ad hoc networks was the problem of routing in multi-hop wireless networks. Routing protocols designed for infrastructured networks are not suitable for ad hoc networks because of the dynamic network topology and the unreliability of the wireless links. Pioneering research efforts on multi-hop packet radio networks were led by U.S. governmental and military agencies, such as the Defence Advanced Research Projects Agency (DARPA) Global Mobile, the U.S. Army's Task Force XXI Advanced Warfighting Experiment and the U.S. Navy and Marines' Extending the Littoral Battlespace [Freebersyser and Leiner, 2000].

In the late 1990's the IETF Mobile Ad Hoc Network (manet) Working Group (WG) was created to develop and standardize IP routing protocol functionality suitable for wireless routing applications within both static and dynamic topologies [Corson and Macker, 1999]. In the last decade, the manet WG has

published a number of RFC memoranda regarding experimental routing protocols for ad hoc networks. Although RFC do not specify de jure Internet standards, they are usually taken as de facto standards in the industry.

Routing protocols for ad hoc networks can be classified, according to the manet WG, in either reactive manet protocols (RMP) or proactive manet protocols (PMP). RMP are also known as on-demand routing protocols because route discoveries are only triggered when there is a need for communication. PMP are also known as table-driven protocols because routing tables are periodically distributed and maintained among ad hoc network nodes. At the time of writing, manet WG has published four RFC regarding manet routing protocols, two RMP and two PMP: Ad Hoc On-Demand Distance Vector (AODV) Routing [Perkins et al., 2003], the Dynamic Source Routing Protocol (DSR) [Johnson et al., 2007], Optimized Link State Routing Protocol (OLSR) [Clausen and Jacquet, 2003] and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [Ogier et al., 2004]. AODV and DSR are RMP while OLSR and TBRPF are PMP. There are also many proposed, but not standardized, routing protocols that are not exclusively an RMP or a PMP, but instead combine elements of both, and therefore are known as hybrid routing protocols.

However, ad hoc networking demands more than just appropriate routing protocols. As already mentioned, other network services and mechanisms for network addressing, security and privacy needed to be addressed as well. Sections 2.2 and 2.3 introduces the state-of-the-art security and privacy research in ad hoc networks.

2.2 Security and Privacy Threats

This section presents an introduction to security and privacy threats in ad hoc networks. Basically, the concerns about security in ad hoc networks have three sources:

- the shared physical medium used in wireless communications;
- the lack and independence of an online infrastructure, and;
- the limited physical security of mobile devices.

The nature of wireless communications voids the possibility of introducing geographical boundaries in wireless networks. As no network borders exist to be defended, security provisioning cannot be deployed in the same manner as in a wired network, i.e., by setting a network perimeter and protecting this perimeter with traditional network security enablers, such as border firewalls and network-based intrusion detection systems (IDS). Therefore, the concept of

insider and outsider attackers is fuzzy and it usually depends on the application scenario, e.g., an outsider attacker in a sensor network does not control any sensor devices, while an insider attacker has tampered one or more devices.

Yet another consequence of a shared wireless physical medium is that eavesdropping is much simpler in such wireless environments compared to wired networks. Although data confidentiality can be secured using end-to-end cryptography, the informational privacy aspects are not easily protected. For instance, message senders, messages and message recipients can be linked by an eavesdropper that analyzes *source* and *destination* fields of captured IP packets (as long as the IP header is not encrypted).

In addition, the nature of wireless communications often, if not usually, infers mobility as well. The majority of mobile devices, e.g., mobile phones, laptops, are also personal devices, and are therefore bearers of an intrinsic information related to mobility: the user geographical location information. The protection of such location data is usually referred to as location privacy. Location privacy is further discussed in Section 2.2.3.

The lack and independence of an online infrastructure prevents participants of an ad hoc network from having unrestricted and continuous access to centralized key security services, such as authentication, authorization and access control.

Limited physical security means that devices are more easily lost or stolen than their wired counterparts. Thus, physical access to a device is more difficult to prevent in ad hoc networks on which nodes are relatively small and mobile, e.g., portable computers and mobile phones, and are usually not stored in a locked server room. Therefore, it is often assumed that ad hoc network devices are easier to compromise than wired devices.

Security threats in ad hoc networks are an extension of the threats found on wired networks. The threats are basically a combination of known threats for wireless and wired local area networks, threats that are derived from the general characteristics and constrains of ad hoc networks, and threats that are dependent on each specific application scenario, e.g., security and privacy threats in military ad hoc network scenarios differ significantly from threats in civilian ad hoc networks.

The acceptable trade-off between the level of security and privacy achieved and the amount of network performance loss is important in ad hoc networks to establish the cost of security and privacy in terms of network performance parameters. Network performance can be measured in terms of quality of service parameters, such as data throughput, end-to-end delay, dropped packet rate, transmission errors, and jitter. Such an evaluation is fundamental for devices with limited capabilities since it is possible to anticipate the expected lifetime of a device that is battery-driven, for instance. In Chapter 8 we analyze the performance of the overlay anonymous communication mechanism called Chameleon [Martucci et al., 2006a], which is presented in Chapter 6.

In conclusion, the aforementioned characteristics of ad hoc networks imply that every device has to be able to provide its own security and privacy services without being dependent on unrestricted access to central servers [Buttyán and Hubaux, 2003; Feeney et al., 2001; Čapkun et al., 2006]. Even though these threats are described in several published works, such as [Feeney et al., 2001; Hubaux et al., 2001; Stajano and Anderson, 1999], we intend to provide a brief description of security threats and their relation with ad hoc network characteristics, in order to deliver enough background for the good understanding of the rest of this dissertation.

The remainder of this section is divided in three parts. In the first and second parts we present passive and active security threats in ad hoc networks, in Sections 2.2.1 and 2.2.2, respectively. Section 2.2.3 discusses privacy threats in ad hoc networks.

2.2.1 Security Threats: Passive Attacks

Ad hoc networks are susceptible to eavesdropping due to the nature of the wireless communication medium. Interception of radio carriers and of the data contained in them is usually considered unavoidable if an attacker is eavesdropping the radio spectrum of a victim user. Therefore, attacker models for ad hoc networks always assume that an attacker can perform a passive attack.

Eavesdropping communication channels and subsequent storage of the captured data always precede traffic analysis. Traffic analysis is a powerful method that can be exploited in the context of both security and privacy (see more in Section 2.2.3). It can be used to discover cryptographic keys used and bypass authentication in legacy security modes of IEEE 802.11 networks [Arbaugh et al., 2002; Borisov et al., 2001].

2.2.2 Security Threats: Active Attacks

Active attacks in ad hoc networks include the same set of attacks against security services on wired networks plus a set of attacks that are specific to ad hoc networks. In this section we list active attacks in ad hoc network environments. These attacks are classified in two groups. The first group lists attacks that are common to both wired networks and ad hoc networks, with focus on vulnerabilities found in wireless networks. The second group lists attacks that are specific to ad hoc networks or are otherwise very uncommon in wired networks.

The following active attacks are common to both wired and ad hoc networks [Bishop, 2004; Menezes et al., 1996]:

- *replay attacks* — replay attacks involve capturing, storing and retransmission of a message or a sequence of messages. Replay attacks often

prelude other security attacks. Wireless networks are highly susceptible to replay attacks, as messages are transmitted "over-the-air" and are, thus, vulnerable to be intercepted and replayed.

- *masquerade or impersonation attacks* — masquerade or impersonation attacks occur when one entity pretends to be another entity. Unprotected or weak authentication mechanisms usually lead to this security threat, as message sequences can be replayed and data link addresses can easily be spoofed in wireless networks [Barbeau et al., 2006]. Man-in-the-middle (MitM) attacks often prelude impersonation attacks. An impersonation attack in a wireless local area network is often the result of a MitM attack caused by flaws in tunnelled authentication mechanisms [Asokan et al., 2002]. Impersonation attacks can also lead to privacy threats, as discussed in Section 2.2.3.

- *message modification attacks* — message modification attacks happen when a message or a sequence of messages are captured or intercepted, altered and retransmitted. Intentional delaying and message reordering are also considered to be modification attacks. In order to prevent this kind of attack, data integrity must be guaranteed. Protection against modification attacks is essentially based on the same suite of protocols in wireless as in conventional wired networks. However, ad hoc networks are more susceptible to message modification, as data is relayed by every node, trusted or not, along the routing path that connects a sender and the recipient devices in the wireless network.

- *denial of service (DoS) attacks* — DoS attacks prevent or inhibit the service provisioning by a device. DoS attacks can be launched in different layers of the TCP/IP stack. Wireless networks are particularly vulnerable to physical layer DoS attacks. The disruption of wireless networks involves jamming the frequency range used in the wireless communication. Since the spectrum range, encoding schemes and frequency patterns are standardized in every civilian communication system, such as in IEEE 802.11 [IEEE 802.11], IEEE 802.15 [IEEE 802.15.3; IEEE 802.15.1; IEEE 802.15.4] and IEEE 802.16 [IEEE 802.16] technologies, DoS attacks in the physical layer are feasible to any resourceful attacker.

All the aforementioned attacks exploit the unbounded and shared nature of the wireless communication. Replay, masquerade and message modification attacks are preceded by eavesdropping data traffic in the wireless network. The following list present active attacks that are specific to ad hoc networks and their variants or are potentially more harmful and more difficult to protect against in ad hoc networks environments than in wired networks.

- *attacks on routing mechanisms* — routing protocols designed for ad hoc networks are usually vulnerable to a set of attacks aiming to influence or interfere on data communication flows in an ad hoc network. Attacks over routing protocols poison routing tables with erroneous or incorrect information. The goal of such attacks is to cause communication disruption, to logically isolate one or more devices from the rest of the network, DoS, or for gathering data for future traffic analysis. Attacks against ad hoc routing protocols often try to build wormholes or set sinkholes in the ad hoc network:

  - *wormholes* consist of bidirectional tunnels in an ad hoc network that are used to forward packets, including routing control messages, from one geographical location of an ad hoc network to another distant location. Setting a wormhole needs two or more colluding nodes in the ad hoc network. Wormholes prevents the logical topology of such a network from reflecting the actual physical topology, with undesired effects on routing protocols [Naït-Abdesselam et al., 2008].
  - *sinkholes*, also called *blackholes*, are malicious devices that lure other nodes to forward traffic through it, usually sending false routing control messages and thus manipulating the ad hoc routing table of other nodes in the proximity [Karlof and Wagner, 2003]. A device acting as a sinkhole can either capture and store the forwarded traffic for future traffic analysis, selectively drop packets, e.g., forward only control packets but no data packets [Hu et al., 2003], or simply block all network traffic.

  Attacks against ad hoc routing protocols usually depend on the operation details or operation mode, i.e., PMP or RMP, of each particular routing mechanism. Examples of such attacks are the Byzantine attack, the rushing attack, and flooding of specific routing control messages.

  - *the Byzantine attack* is a family of attacks that involves any authorized device or set of authorized devices in an ad hoc network to cause routing service disruption or degradation [Awerbuch et al., 2002; Lamport et al., 1982]. A Byzantine attack is deployed by devices that present a Byzantine behavior. An example of a Byzantine attack directed to service degradation in ad hoc networks is the jellyfish attack. A jellyfish attack consist of one or more nodes in an ad hoc network that maliciously batch and delay packets in an ad hoc network [Aad et al., 2004].
  - *the rushing attack* is an attack designed for RMP and consists of sending multiple route requests and more quickly than the other devices in the ad hoc network in an attempt to force other devices

to include a hop through the attacker [Hu et al., 2003]. Byzantine attacks and rushing attacks can be used to build wormholes and set sinkholes in an ad hoc network.

- *the flooding of HELLO messages attack* target some ad hoc routing protocols that use HELLO messages, which are routing control messages to discover neighbor devices. A flooding of HELLO messages occurs when an attacker that has a more powerful radio transmitter device than the other network nodes broadcasts routing control messages (or replay messages from other devices) to nodes that are located geographically far away in the ad hoc network, i.e., two or more hops away, from the attacker [Karlof and Wagner, 2003]. Thus, under a flooding of HELLO messages attack, the logical network topology is inconsistent with the physical network topology.

- *Sybil attacks* — a Sybil attack [Douceur, 2002] is an identification attack that occurs when a malicious user influences the network by controlling multiple logical identifiers from a single physical device. In a Sybil attack, malicious users assume multiple identifiers, preventing the usage of security mechanisms based on filters, reputation or trust assumptions. This attack was first identified in peer-to-peer networks, but can also be used to disrupt ad hoc networks, including ad hoc routing protocols. Sybil attacks have deep implications in the general security expectations of an ad hoc network as discussed in Chapter 3. This attack is also strongly correlated to privacy issues in ad hoc networks.

- *battery exhaustion attacks* — battery exhaustion attacks are a variant of DoS attacks. They are sometimes referred to as sleep-deprivation attacks [Stajano, 2001]. Wireless network devices are usually mobile devices that are battery-driven, i.e., they depend on battery to remain active. Thus, mobile devices are susceptible to battery exhaustion attacks. In this attack, the attacker aims to exhaust the battery power of a target device and render it useless by forcing it to receive, transmit or process data that this device should not need to in a normal situation.

2.2.3 Threats to Privacy

The threats to informational privacy in ad hoc networks are the same that exist in other computer systems. However, the aforementioned characteristics of ad hoc networks, i.e., the shared physical medium used in wireless communications, the lack and independence of an online infrastructure, and the limited physical security of mobile devices, contribute to make these networks more vulnerable to privacy infringements than their wired counterparts.

Applications can leak vast amounts of possibly sensitive data if the application data is being transmitted among the participating mobile devices. In

addition, traffic information generated inside such networks can potentially reveal sensitive data about users and their communicating partners. Moreover, ad hoc network devices can be geographically pinpointed by non-authorized parties. Malicious users may even track other users by following beacon signals emitted by their mobile devices, such as neighborhood discovery messages present in some ad hoc routing protocols. Finally, the personal data collected in an ad hoc network can be used to build user profiles that include the history of communicating partners and current and past geographical locations.

It is fundamental for an attacker whose objective is to profile users in an ad hoc network to uniquely identify devices and also to recognize distinct occurrences of the same device in different contexts, i.e., to gather historical data regarding the devices that are being monitored and connect them to unique identifiers.

Unique identifiers can be obtained from different sources in an ad hoc network device, ranging from the physical layer to the application layer. Hence, to identify potential threats to privacy it is necessary to list possible sources of identifiable data that can be used by an attacker. The TCP/IP stack organization is used as a support for listing sources of identification in an ad hoc network, and for explaining the existing threats to these identifiers. A bottom-up approach is used in the rest of this section, i.e., from the physical layer to the application layer.

Location privacy is another aspect of informational privacy that is related to the geographical information associated with a user. The nature of wireless communications allows user mobility and seamless connectivity. However, mobile devices can be fingerprinted and users' geographical location information and mobility patterns can be profiled. Location data are personal data that can be related to an identified or identifiable individual, and could therefore be misused for criminal purposes, unsolicited profiling, or for revealing information about the users' social contacts. Even when consent has been given by a user to an application and the location data are processed accordingly, users practically lose control over what happens with their location data [Martucci et al., 2006b].

The Pfitzmann and Hansen terminology [Pfitzmann and Hansen, 2008] version 0.31 is followed for the definition of privacy related terms such as anonymity, unlinkability, and pseudonymity in the rest of this dissertation.

Physical Layer

Physical layer attacks against privacy aim either to discover the geographical location of a device in a wireless network or to identify patterns in the emitted radio frequency (RF) signals that can be uniquely associated with a given device. RF triangulation and fingerprinting are two techniques that can be used to uniquely identify a device in a wireless network.

RF triangulation is a technique used to pinpoint the geographical location of a given device. This technique requires the deployment of passive devices in the wireless network that are able to collect signal strength information of RF transmissions emitted by a target device. The location of the sensors is assumed to be known. By combining the data collected by the sensors it is possible to determine the geographical location of the target device. Access points in IEEE 802.11 networks can be used as sensor nodes for RF triangulation, as presented in [Bahl and Padmanabhan, 2000; Ladd et al., 2002], and produce results with errors in the order of meters. RF triangulation can effectively locate the position of a transmitting device, but lacks the ability to link historical information to identify multiple appearances of a specific device [Brik et al., 2008]. RF triangulation is thus mainly a threat to location privacy, since it allows an attacker to locate the geographical position of a given device.

RF fingerprinting is a general umbrella term for different techniques involving the analysis and identification of unique characteristics in the RF emission by a transmitting node. The perceived signal-to-noise (S/N) ratio can be used as unique temporal characteristic to identify a device [Gruteser and Grunwald, 2005]. Signal processing and profiling is an RF fingerprinting technique that can be used to discriminate RF emitted from different wireless Ethernet cards based on signal fragments [Gerdes et al., 2006].

Transient signal detection and analysis is concerned with the characteristics of transmitted RF signal during the transient period, i.e., the start-up period prior to the actual transmission. The radio transmission during the transient period has consistent features, such as amplitude and phase components that are not easily forged yet not necessarily unique [Barbeau et al., 2006]. Modulation domain techniques compare received signals to the expected ideal in the modulation domain and are used to identify specific transmitters. Modulation domain techniques require previous knowledge of the modulation scheme being employed [Brik et al., 2008]. This requirement is hardly a hindrance, since modulation schemes are standardized and public.

RF fingerprinting is particularly useful to detect devices that deliberately change their hardware address information in attempt to prevent tracking. Transient signal analysis, signal processing, transient detection, and modulation techniques rely on the fact that transceivers are not exactly equal. Hardware imperfections in the transceivers create unique radio characteristics that enable devices to be uniquely identified. Eliminating such imperfections during manufacture is possible, but is not economically viable [Brik et al., 2008].

Changing upper layer identifiers, i.e., MAC and IP addresses, cannot prevent the possible fingerprinting of a radio transmitter of a given device. RF fingerprinting can be exploited to infringe location privacy rights, since the attacker acquires information about the approximate location of a given device, i.e., the targeted device is on the attacker's radio range. Furthermore, an attacker can identify the presence of a target device in different periods of time

and different locations using RF fingerprinting even if its hardware and logical have changed. Thus, this is a threat to unlinkability since multiple appearances of a device in different instants of time or locations can be linked.

Data Link Layer

Data link layer attacks against privacy involve identifying and tracking unique characteristics that exist at this layer. The standard identifier in this layer is the hardware (MAC) address. Hardware addresses are assigned by the manufacturer and their intent is to uniquely identify a network interface card in a local area network. Due to this fact, hardware addresses are the easiest and most feasible way to track a wireless device. Still, these addresses can also easily be changed by software.

In addition, there are further techniques to identify devices using data link information. The sequence number information that exists in the IEEE 802.11 header can be used to detect MAC address spoofing by identifying gaps in the sequence number of the frames transmitted by a device [Guo and Chiueh, 2005]. This feature could also potentially be used to detect MAC address changes or a device using multiple MAC addresses. This technique of following sequence numbers is also known as the “who am I?” attack [Danezis, 2004].

Another technique is to identify differences in the implementations of the active scanning algorithm in IEEE 802.11 wireless network card drivers. This technique is however limited since it cannot make any distinction between two devices running the same driver. Other limitations that can thwart the device driver fingerprinting also include driver code modification and noise generation [Franklin et al., 2006].

Similarly to physical layer threats to privacy, data link threats can be exploited to infringe location privacy rights, since hardware addresses rarely change. Furthermore, even if a hardware identifier is understood as a pseudonym and it is changed, the previous and the current MAC addresses can possibly be linked using the techniques described in this section, i.e., a threat to unlinkability between two pseudonyms of a device.

Network Layer

The standard identifier in the network layer is the IP address. IP addresses are logical addresses that can be either static or dynamic. Static means that the IP address is configured locally at the computer, while a dynamic means that it is assigned to a device by a central server, usually using the Dynamic Host Configuration Protocol (DHCP) [Droms, 1997]. IP addresses can easily be modified by software. IP is the standard routed protocol of the TCP/IP suite and it is used by routing protocols to select to which node a packet should be forwarded.

At the time of writing, there are no specific standards for IP address assignment in ad hoc networks. The Ad Hoc Network Autoconfiguration (autoconf) IETF WG [IETF autoconf] task is to develop a network addressing model that allow ad hoc network devices to configure their network addresses in a transparent way, i.e., without interfering with other parts of the system.

Privacy threats in the network layer include the tracking of devices using the IP address as a unique identifier and ascertaining about the linkability between two communicating devices, i.e., a violation of relationship anonymity by analyzing the network data traffic and dissecting the *source* and *destination* fields of an IP packet. The standard ad hoc routing protocols AODV [Perkins et al., 2003] and DSR [Johnson et al., 2007] leak the IP addresses of the sender and the destination during their path discovery phase, for instance.

The Internet Control Message Protocol (ICMP) [Postel, 1981] can be used for active fingerprinting based on the clock skew of the target device (clock skews are further explained in the next section) [Kohno et al., 2005]. There are two requirements for the success of ICMP fingerprinting: the implementation of the TCP/IP stack of the target device must answer ICMP Timestamp Request messages[3], and target device should maintain its system time using the Network Time Protocol (NTP) [Mills, 1992].

In comparison to physical and data link privacy threats, threats in the network layer have a significant difference regarding the attack range, i.e., the geographical area affected in an ad hoc network. The attacker in the latter case needs only to be part of the path linking the source to the destination, and not necessarily in the radio range of the target device.

Transport Layer

Transport layer information can be used to fingerprint network devices by analyzing the clock skew information [Kohno et al., 2005]. The underlying assumption of this attack against privacy is that different devices have different clock skews, and a given device has a constant clock skew in general. Thus, it is possible for an attacker to retrieve and collect a target's perceived time information from the 32-bit timestamp field present in the TCP header. The TCP timestamp option was introduced in the RFC 1323 [Jacobson et al., 1992]. Results reported in [Kohno et al., 2005] using passive and semi-passive attacks in different scenarios (including a non ad hoc wireless scenario) suggest that clock skew estimation is in general independent of topology and distance between targets and attacker devices. Attackers do not necessarily have to be in the radio range of the target device when deploying a transport layer fingerprinting attack. Instead, it is enough to be part of the path connecting the sender to the recipient.

[3]The MAC OS X 10.3 Panther does not reply to ICMP Timestamp Request.

Transport layer threats to privacy can be exploited to link different pseudonyms of a device. Thus, if an attacker is able to establish or eavesdrop a TCP connection of a target device that exchange TCP timestamps, the attacker may be able to find out if the target device has previously appeared in the network under a different identifier, i.e., another combination of hardware and logical addresses.

Application Layer

Information encapsulated in the application layer or other personal data contained in the message payload may identify the sender or the recipient of a message, or both sender and recipient and thus expose their communication relationship. The information collected in the application layer is highly dependant of the application itself, e.g., sender and recipient fields of Simple Mail Transfer Protocol (SMTP) message envelope and the input data generated by the user. In the same way as network and transport layer fingerprinting, attackers do not necessarily have to be in the radio range of the target device to analyze application layer information. It is enough for an attacker to be part of the path connecting the sender to the recipient. Application layer data is, furthermore, end-to-end information, i.e., the recipient of the message is guaranteed to be the final recipient, and not just an intermediary (proxy) device. Leaks of personal information in the application layer were analyzed in [Aura et al., 2008], in application layer protocols such as the Domain Name System (DNS), the NetBIOS over TCP (NBT), and the Dynamic Host Configuration Protocol (DHCP).

2.3 Enhancing Security in Ad Hoc Networks

The literature on security enhancements for ad hoc network is vast and includes adaptations and adjustments of existing security mechanisms designed for wired networks, such as secure routing protocols, key management schemes, host-based intrusion detection mechanisms, trust management, and neighborhood and service discovery. Security enhancements for ad hoc networks differ significantly according to the authors' assumptions regarding the characteristics of an ad hoc network, especially regarding the existence and availability of external services and trusted third parties. Thus, security enhancements for ad hoc networks can be organized according to the classification of ad hoc networks presented in Section 2.1.2.

The objective of this section is to provide an overview of security enhancements for ad hoc networks regarding their requirements and assumptions. This section includes the introduction and presentation of a comprehensive list of security enhancements for ad hoc networks that represent each of the three

groups regarding the classification introduced in Section 2.1.2. The security enhancements are then sorted regarding the way that identifiers are generated, obtained and, eventually, transferred [Martucci, 2006].

The generation and distribution of digital identifiers is an operation that is usually linked to the initialization of the ad hoc network. Cryptographic key generation and distribution are also linked to the initialization of such a network and are usually related or connected to digital identifiers, since cryptographic keys are often used to perform authentication. Thus, the categorization presented in this section can also be related to classifications of key distribution schemes in ad hoc networks [Merwe et al., 2007; Wu et al., 2008]. For this reason, this section includes very few security mechanisms that do not deal with these fundamental issues in the initialization of ad hoc networks, i.e., mechanisms that assume a set of pre-distributed cryptographic keys, such as the majority of secure ad hoc routing mechanisms. A short selection of secure routing mechanisms is, however, presented in this section.

The section follows with a presentation of the security mechanisms designed for ad hoc network devices that have intermittently connectivity to an infrastructure. Enhancements that are based on one or more privileged devices in the ad hoc network are introduced in Section 2.3.2. Fully independent and self-organized security mechanisms are the topics of Section 2.3.3. Secure routing protocols are briefly presented in Section 2.3.5. Finally, an introduction to the limitations of physical and data link security is presented in Section 2.3.6.

2.3.1 Intermittently Connected to an Infrastructure

Security models belonging to this group assume that the ad hoc network devices have occasional or periodic connectivity to an established infrastructure. Security solutions that belong to this category often also assume the existence of an online trusted third party and eventually other services that are already present in the Internet for the distribution of cryptographic keys or digital certificates in an ad hoc network, such as a Public Key Infrastructure (PKI).

The bootstrapping stage of the self-certified Sybil-free framework presented in Chapter 5 and in [Martucci et al., 2008a] also relies on a trusted third party. The distribution, renewal and revocation of identifiers in [Kargl et al., 2006] is based on intermittent connection with a certificate authority. Likewise, the bootstrapping phase in [Sanchez and Baldus, 2005] also depends on the presence of a PKI for the distribution of digital certificates in the ad hoc network.

Cryptographically generated addresses (cga) and statistically unique and cryptographically verifiable (SUCV) identifiers are IPv6 addresses that are bound to a public key [Aura, 2005; Montenegro and Castelluccia, 2002]. These cryptographic based techniques could be used for secure identification in ad hoc networks if such keys were issued by a trusted third party.

2.3.2 One or More Privileged Devices

Security solutions that are part of this group assume that one or more devices have a special role in the network, such as personal certificates authorities and local certificate repositories. These certificate authorities issue identifiers, such as credentials or X.509 compliant certificates [ITUT X.509] to other devices in the ad hoc networks usually through a secure channel. Following the classification of Section 2.1.2, this group can be divided into two subgroups: one or more trusted devices in the ad hoc network or having trust divided among several devices.

One or More Trusted Devices

This group has the straightforward solution for having one or more devices that have a special role in the network, i.e., having devices with extra privileges and rights collocated in the ad hoc network. Such privileged devices have the same behavioral properties as other nodes in the ad hoc network in the sense that they may be mobile, battery-driven and may occasionally be present in the ad hoc network. These devices have exclusive and reserved roles in the ad hoc network, such as issuing certificates and publishing certificate revocation lists.

The basic rationale behind the solutions included in this category is that personal devices can be used to set up personal ad hoc networks using devices that belong or are known to the user. One or more known, i.e., trusted, devices then issue certificates to other devices. Security models included in this group of solutions are usually designed for ad hoc networks that are limited in size since such a model include only devices that are known to the user. On the other hand, they provide reasonable and practical solutions for generation and distribution of digital identifiers.

The resurrecting duckling model [Stajano, 2001; Stajano and Anderson, 1999], for instance, is based on a unique trusted device that can bind or remove other devices to its own ad hoc network using a secure side channel for bootstrapping. A secure authentication protocol that is derived from the aforementioned model is described in [Balfanz et al., 2002].

A secure model for ad hoc network and its implementation, a trust-based security architecture that rely on one or more trusted devices are presented in [Venturini et al., 2002] and [Martucci et al., 2004b], respectively. The model is built on top of a service-oriented network, i.e., an abstraction layer on which devices are seen as either service providers or service consumers and can be located according to provided services[4].

[4]Service-oriented networks include platforms sponsored by the industry, such as Jini [Jini] and UPnP (Universal Plug and Play) [UPnP], but also some proposals like Konark, which is a service discovery protocol designed for ad hoc networks [Helal et al., 2003].

The privileged device in this secure model for ad hoc networks runs a registration service that distributes certificates, and assigns access control rights and privileges to other network devices. Privileges could include the issuing of certificates to other devices, in a sense that a hierarchical authority structure could be built on an ad hoc network. Likewise, flat authority structures could be set by linking two different registration services. Certificates are distributed through a secure channel using shared keys that are entered as pin numbers on both the registration service and the requesting device. Users that are not registered may benefit from some services that are public. Public key authentication is used and preceded by a pre-authentication mechanism based on shared keys whose purpose was to provide a limited form of authentication to low-end devices [Martucci et al., 2004a].

Another fundamental and exclusive task assigned to registration services is to follow the behavior of ad hoc network devices by evaluating security reports generated by other devices in the ad hoc network. Behavior plays a key role in this security model. It is used to compute trust values that are assigned to network devices. Trust is an inherent part of this proposal that adds flexibility to the security architecture [Martucci et al., 2004b].

The dynamic nature of trust is used as a tool to enhance access control by establishing minimum trust requirements to grant access to network services. The trust requirements are defined when adding new services to the network. Every device joining the network is assigned with an initial trust value by the registration service during a bootstrapping phase. Trust values are assigned to both users and devices. Hence, a network service evaluates the trust values assigned to both user and the device against its trust requirements. For assessing trust, this security model used the algebra for assessing trust in certification chains presented in [Jøsang, 1999].

Trust information may change throughout the network lifetime. Thus, trust values are recomputed by the registration service into new trust information tables, which are distributed among the network devices. Registration services can eventually revoke access rights temporarily or definitely. In the absence of the registration service to update trust information tables, local tables, which are locally and frequently updated, are used to offer protection against misbehaving users [Martucci et al., 2004b].

Prototypes were implemented in Java for the mechanisms presented in [Balfanz et al., 2002], [Martucci et al., 2004a] and [Martucci et al., 2004b]. These implementations explicit the application scenarios for ad hoc networks that were being aimed for the security models included in this group, i.e., user-oriented, targeting devices that belong or are known to the user that is setting the ad hoc network. Such solutions are, however, not designed for some application scenarios that involve many devices without any previous knowledge or trust relationship.

Distributed Trust Among Devices

Security models included in this group assume that trust is distributed among devices in the ad hoc network. The solutions belonging to this model use a $(n, t+1)$ threshold signature scheme to form a distributed certificate authority [Merwe et al., 2007]. The private key of the distributed certificate authority is shared among a set of network devices. As long as there are $t+1$ devices in the radio range, i.e., single-hop distance, that have a share of the private key, digital certificates can be issued. The usage of threshold cryptography was first proposed in the context of ad hoc networks in [Zhou and Haas, 1999] and later extended in [Kong et al., 2001; Luo et al., 2002], such that n can be all devices in the ad hoc network. The amount $t+1$ of devices needed to issue a certificate depends on the implementation.

Intrinsic problems of such implementations are how to set the private key and how to distribute the key shares [Merwe et al., 2007]. In [Luo et al., 2002] the generation of the private key and distribution of its shares are given to trusted third party that is needed during network bootstrap.

2.3.3 Fully Independent and Self-Organized

Security solutions that are part of this group assume that devices generate their own public and private key pairs (one or more), issue their own digital certificates and distribute them to other devices in the ad hoc network. Such approaches are similar to Pretty-Good-Privacy (PGP) in the aspects of cryptographic public and private key pair generation and they are based on the concept of Web of Trust [Zimmermann, 1995]. Methods included in this group do not require any central repository for storing digital certificates, but require users to establish security associations consciously [Čapkun et al., 2006]. Thus, solutions included in this category are fully independent of any pre-deployed infrastructure, have no central point of trust and are designed to operate in complete isolation from any deployed infrastructure. The first solution proposed to secure ad hoc networks that is based on the concept of Web of Trust was presented in [Čapkun et al., 2003a]. In this proposal, users authenticate other users and issue certificates that are used as recommendation to other users. These recommendations are used to build certificate graphs that are exchanged between neighbor devices. The proposal assumes that certificate chains are possible to be constructed under a small world scenario assumption [Čapkun et al., 2002, 2003a; Watts, 1999].

An earlier solutions not based on Web of Trust is presented in [Feeney et al., 2001]. The authors proposed the distribution of session keys among the ad hoc devices using a secure side channel, such as a low-power infrared communication interface. Another proposal for fully independent and self-organized networks that also uses a secure side channel for the distribution of cryptographic

keys and identifiers is presented in [Čapkun et al., 2003a].

A major criticism over such self-organized solutions relying on a certificate chains, i.e., Web of Trust, is that such approaches provide weak authentication [Merwe et al., 2007]. Weak authentication is resulting from the assumption that trust is transitive, i.e., that the following assumption is true [Abdul-Rahman and Hailes, 1997]:

$$(Alice.trusts.Bob).\&.(Bob.trusts.Cathy) \Rightarrow (Alice.trusts.Cathy) \quad (2.1)$$

However, this type of unconditional transitivity of trust is not generally true as concluded in [Abdul-Rahman and Hailes, 1997; Christianson and Harbison, 1996]. In addition, transitivity creates a recursive trust characteristic [Jøsang, 1996] that is not modelled in none of the aforementioned solutions. Such considerations regarding trust transitivity might also impact other models that are not fully independent and self-organized, depending on the selected trust model. Furthermore, the chain is as strong as its weakest link. Then, any compromised device along the chain can result in the compromising of the whole chain [Merwe et al., 2007].

An important drawback in such fully independent and self-organized solutions is that there are basically no guarantees regarding protection against Sybil attacks [Douceur, 2002] since there are no limits for the number of identities (authentic or not) that a user may issue. Therefore, impersonation attacks can easily be deployed and certificate chains poisoned with Sybil identifiers.

2.3.4 Hybrid Models

The three aforementioned security models are not necessarily disjointed. There are some solutions that are hybrid, i.e., they combine characteristics of the different models. For instance, a distributed certificate authority using threshold cryptography is combined with a certificate chain structure in [Yi and Kravets, 2004].

2.3.5 Security Enhancements in the Network Layer

Security enhancements in the network layer of ad hoc networks basically consist of secure routing proposals. The literature regarding secure routing in ad hoc networks is vast. In this section, we briefly introduce five selected secure routing mechanisms from the perspective of assumptions and requirements needed in each mechanism. These five protocols are representative since they include security extensions for standardized ad hoc routing protocols and also standalone routing protocols. They also include both proactive and reactive routing protocols. The five protocols presented in this section are: SRP [Papadimitratos and Haas, 2002], SEAD [Hu et al., 2002], SAODV [Zapata and

Asokan, 2002], ARAN [Sanzgiri et al., 2002], and Ariadne [Hu et al., 2005]. They are presented in chronological order of publication.

Detailed descriptions regarding the functionality of these protocols are not provided in this dissertation. For such information refer to the papers and articles referred to in this section. Moreover, surveys on secure routing protocols can be found in [Hu and Perrig, 2004] and in [Argyroudis and O'Mahony, 2005].

Secure Routing Protocol (SRP)

The Secure Routing Protocol (SRP) [Papadimitratos and Haas, 2002] is an RMP protocol that can be used as a security extension to other routing protocols that have their route discovery mechanism based on broadcast of query packet, such as the Dynamic Source Routing (DSR) protocol [Johnson et al., 2007].

SRP assumes pre-existent secure association between the source and the destination devices. This pre-existing relationship between source and destination is needed for the establishment of a shared key between them. The details on how such association, i.e., trust relationship, is built is not explicit in the paper. Public keys are, however, suggested as a possible solution to this problem.

Secure Efficient Ad Hoc Distance Vector (SEAD)

The Secure Efficient Ad Hoc Distance Vector (SEAD) protocol [Hu et al., 2002] is based on the design of the Destination-Sequenced Distance-Vector (DSDV) routing protocol [Perkins and Bhagwat, 1994], which is a PMP, i.e., a proactive (table-driven) protocol. SEAD uses one-way hash chains for authentication and assumes the existence of a mechanism to distribute such hash chains.

The authors of SEAD suggest, among other options, the use of public keys to sign elements of a hash chain as a possible solution to securely distribute an authenticated hash chain. This option implies the usage digital certificates issued by a trusted third party. The other options include the usage of: Web of Trust based solutions [Hubaux et al., 2001], pre-distributed symmetric keys, or the usage of a secure side channel to distribute the hash chains. The last two approaches implicitly mean the presence of a trusted third party.

Secure Ad Hoc On-Demand Distance Vector (SAODV)

The Secure Ad Hoc On-Demand Distance Vector (SAODV) protocol [Zapata and Asokan, 2002] is an RMP that consists of security extensions to the Ad Hoc On-Demand Distance Vector (AODV) protocol [Perkins et al., 2003]. SAODV uses two mechanisms to secure AODV control messages: digital signatures for authentication of the static fields, i.e., non-mutable, fields of a message and hash chains to secure the hop count information.

The public key pairs used to generate and verify the digital signatures are obtained from a pre-existing key management system. Furthermore, public key pairs are needed to be bound to the device identity and every other device should be able to verify the correctness of such binding. The implementation of such a key management system is left open in the SAODV proposal. The authors suggest that digital certificates issued by a certificate authority (CA) can be used as a key management system in SAODV.

Authenticated Routing for Ad Hoc Networks (ARAN)

The Authenticated Routing for Ad Hoc Networks (ARAN) [Sanzgiri et al., 2002] is an RMP, i.e., on-demand routing protocol, designed to provide end-to-end authentication, message integrity and non-repudiation in ad hoc networks.

ARAN assumes a preliminary certification process and requires every device to have a certificate. Such certificates are issued by a trusted certificate server and require a secure communicating channel between the server and the device requesting the certificate, such as a secure side channel.

Ariadne

The Ariadne protocol [Hu et al., 2005] is an RMP that is based on the Dynamic Source Routing (DSR) protocol [Johnson et al., 2007]. Ariadne authenticate routing messages using one of the following key management schemes: pre-distributed pairwise symmetric keys among all nodes in the ad hoc network, digital signatures, or pre-distributed symmetric keys between communicating nodes with TESLA broadcast authentication[5].

Thus, Ariadne assumes a pre-existing key management scheme, but does not elaborate on possible schemes. However, the options are limited to the set of solutions discussed in SEAD, i.e., one of the following options: a trusted third party, a Web of Trust based solution or relying on a secure side channel for key distribution, which also implies a trusted third party nonetheless.

Conclusions Regarding Secure Ad Hoc Routing Protocols

All five aforementioned secure routing protocols assume a pre-existent key management for the generation and distribution of cryptographic keys or identities among the participants of the ad hoc network, that is either based on a centralized trusted third party that may or may not be available at all times, or on a Web of Trust. Thus, these routing protocols adhere to the classification presented in Section 2.1.2.

[5]The TESLA broadcast authentication is based on loose time synchronization between senders and receivers and one-way key chains [Perrig et al., 2001].

Apart from secure routing, other proposals in the network layer include the already mentioned cryptographically generated addresses and cryptographically verifiable (SUCV) identifiers used to bind IPv6 addresses to public keys [Aura, 2005; Montenegro and Castelluccia, 2002] and host-based IDS for ad hoc network devices [Albers et al., 2002; Marchang and Datta, 2008]. This section follows with an introduction to physical and data link security mechanisms in wireless networks. The term wireless networks is used intentionally here instead of ad hoc networks because security measures in these layers are designed for single-hop networks.

2.3.6 Regarding Physical and Data Link Protection

Long direct sequence spread spectrum (DSSS) codes or long frequency hopping spread spectrum (FHSS) patterns can be used in proprietary wireless technologies to thwart physical layer eavesdropping. However, DSSS codes and FHSS hop patterns are public and standardized in open wireless communication technologies, such as IEEE 802.11 [IEEE 802.11] and Bluetooth [IEEE 802.15.1]. In addition, changing such codes and hop patterns would hinder interoperability among wireless devices. Furthermore, in open standards, the goal of using spread spectrum modulation is to achieve conformance with spectrum band usage rules in the ISM (Industrial, Scientific and Medical) band, and not to enhance security.

The security mechanisms included in the IEEE wireless technology standards are not suitable for ad hoc networking since they are heavily dependant on the continuous presence of centralized services deployed in the wired network. For instance, the authentication and key distribution service for IEEE 802.11 networks requires IEEE 802.1X authentication and therefore a Remote Authentication Dial-In User Service (RADIUS) server is required to be always available [IEEE 802.1X; IEEE 802.11]. Furthermore, the security concerns of such standards are limited according to bounds of their specification, i.e., their security concerns are limited to the data link layer. Thus, data link layer security does not guarantee end-to-end security in a multi-hop scenario.

Other physical layer security approaches for wireless networks include techniques based on information theoretic security that try to exploit capacity-equivocation and secrecy capacity regions in the physical wireless transmission to achieve confidentiality [Liang et al., 2007]. It basically aims to substitute cryptography for secure encoding. However, such approaches usually require the eavesdropper to have a degraded communication channel in relation to a legitimate receiver.

These intrinsic limitations of physical and data link security are an obstacle for the development of pure physical or data link layer security mechanisms for ad hoc networks.

2.4 Enhancing Privacy in Ad Hoc Networks

The same underlying rationale used for security mechanisms, i.e., the existing security mechanisms are not perfectly suitable for ad hoc networks, is also valid for privacy. The existing anonymous communication mechanisms available for wired networks are not suitable or directly applicable for mobile ad hoc networks. These mechanisms rely either on the constant presence of centralized services and/or on a constant network traffic flow, which implies buffering of real traffic data during periods when the current amount of traffic is higher than the expected or the usage of dummy traffic when this current amount of traffic is below the expected. Relying on the assumption of the continuous availability of a centralized service also conflicts with the requirements for mobile ad hoc networks [Corson and Macker, 1999]. Moreover, keeping a constant traffic flow in the network may degrade the network performance or shorten the device lifetime due to excessive transmissions of dummy traffic.

In this section, we limit our scope to anonymous communication mechanisms in ad hoc networks, which are tools designed to protect informational privacy. Anonymous communication mechanisms are usually designed to achieve sender anonymity and sender-destination unlinkability, i.e., relationship anonymity. Anonymity is defined as the state of being not identifiable within a set of subjects, the anonymity set. Sender anonymity is defined in terms of linkability as the impossibility to link a given message to a particular sender that is part of a set of all possible senders, and relationship anonymity is defined as the impossibility to link to senders and recipients, from the sets of all possible senders and all possible recipients [Pfitzmann and Hansen, 2008].

Current proposals for achieving anonymity in ad hoc networks can be classified in two different groups regarding their level of functionality: either in the network layer (i.e., anonymous ad hoc routing protocols) or as a middleware between the application and the transport layer (i.e., overlay anonymous communication mechanisms). Proposed mechanisms to achieve anonymity provide a certain degree of anonymity at the cost of network performance, which can be evaluated in terms of packet delay, packet loss ratio, computational power required, and amount of data delivered versus amount of data transmitted.

The trade-off between the achieved level of anonymity, the cost of network performance and the assumptions regarding key distribution is what differentiate the proposed mechanisms. For instance, an anonymous communication mechanism that offers protection against a global attacker may require dummy traffic and broadcasting to achieve its goal, while another mechanism that does not offer protection against global attackers may provide better performance.

Anonymous ad hoc routing protocols are presented in the Section 2.4.1, while the overlay anonymous communication mechanisms are presented in Section 2.4.2. Anonymity in the physical layer and data link layer are briefly discussed in Section 2.4.3.

2.4.1 Anonymous Ad Hoc Routing Protocols

Anonymous routing protocols offer privacy enhancements by replacing standard routing protocols. The goal of an anonymous routing protocol is to establish an anonymous path in the network layer between the sender and the destination. The functionality of such protocols can usually be divided into two phases: anonymous neighborhood discovery and anonymous route discovery. Some protocols also specify an anonymous data transfer phase, i.e., how is the sending of data through an anonymous path achieved [Andersson et al., 2008b]. During the anonymous neighborhood discovery phase, neighbor devices, i.e., single-hop distance, establish trust relationships, by exchanging public keys for instance, without disclosing their identifiers [Zhang et al., 2005]. Anonymous route discovery is used to establish an anonymous path between source and destination.

The advantages of implementing anonymity in the routing protocol are the complete transparency towards the application layer and possibly better network performance in comparison to overlay anonymous communication mechanisms, but generally worse compared to standard ad hoc routing protocols. Implementing anonymity in the routing protocol allows data to travel directly from the source to the destination, using the route assigned by the anonymous routing protocol—assuming that the routing protocol works as expected by determining an adequate network path.

A major disadvantage on the other hand is the incompatibility with standard ad hoc routing protocols, which may result in a reduced anonymity set containing only the devices running the anonymous routing protocol, since it is not expected that all ad hoc network users would have an anonymous routing protocol running instead of a standard protocol. Although it is technically possible to have several routing protocols running in the same device, the routing priority is given to the protocol with the lowest cost, which is a local defined parameter. Changing such a parameter to force the selection of the anonymous routing protocol would require some sort of upper-layer intervention, which would void the advantage of the transparency property.

In addition, even if a reasonable amount of devices prioritizes the anonymous routing protocol over the standard routing, a set of devices running only standard ad hoc routing protocols may degrade the anonymity of other devices, since they will not be able to reply to packets encoded according to the anonymous routing protocol and force anonymous nodes to disclose information. Furthermore, since messages are directly transferred from the source to the destination, connection information, e.g., for TCP, the connection tuple: IP source address, IP destination address, TCP source port, and TCP destination port, may potentially expose the relationship between two communicating nodes and compromise some anonymity properties, such as the sender anonymity and sender-destination unlinkability, for instance.

Another disadvantage of network layer privacy enhancements, such as anonymous ad hoc routing, is that they do not offer protection against transport layer fingerprinting, such as the fingerprinting based on clock skew analysis discussed in the Section 2.2.3.

The literature on anonymous ad hoc routing protocols is vast. Many proposals were published in the recent years. The main objective of such protocols is to provide anonymity during the establishment of routes in the ad hoc network and, for some protocols, also to provide location privacy. In this section, we briefly introduce a selection of five anonymous routing protocols from the perspective of assumptions and requirements needed in each mechanism, the type of privacy provided, and the required security mechanisms. The five anonymous routing protocols presented in this section are: ANODR [Kong and Hong, 2003], PPR [Čapkun et al., 2004], SDAR [Boukerche et al., 2004], MASK [Zhang et al., 2005], and AnonDSR [Song et al., 2005]. ANODR, SDAR, MASK, and AnonDSR are RMP and PPR is a PMP.

Detailed descriptions regarding the functionality of the aforementioned protocols are not provided in this dissertation. For such information refer to the papers and articles referred in this section. Furthermore, a survey of such protocols regarding their basic functionality can be found in [Andersson et al., 2008b]. A performance evaluation of ANODR, SDAR, MASK, and AnonDSR was published in [Liu et al., 2006].

Anonymous On Demand Routing (ANODR)

The Anonymous On Demand Routing (ANODR) protocol [Kong and Hong, 2003] is an RMP. Its objective is to provide an untraceable routing scheme that offers unlinkability between senders and recipients and location privacy against a global observer[6]. ANODR is built upon Chaumian MIXes [Chaum, 1981], onion routing [Goldschlag et al., 1996; Syverson et al., 1997], broadcasting, dummy traffic, public key encryption, and one-way hash functions [Kong and Hong, 2003]. There are several proposed variants of ANODR, some based only on public key encryption and others on both symmetric and public key encryption. In this dissertation, we describe just some of those variants regarding the assumptions for identification and key distribution. For a more complete description of ANODR, and its variants, refer to [Kong and Hong, 2003] and [Kong et al., 2005].

In the key pre-distribution scheme variant (ANODR-KPS), symmetric keys are distributed beforehand among the ad hoc network devices by a trusted third party [Kong et al., 2005]. Public keys are used in different variants of ANODR, either as one-time only public keys, i.e., freshly generated public keys that are used only once and afterwards discarded, in the ANODR variant [Kong

[6]A global observer is able to eavesdrop all communication channels in the network simultaneously. However, global observers are not able to break public key or symmetric key crypto-systems.

et al., 2005], or as static public keys that are used multiple times, in the protected onion (ANODR-PO) variant [Kong and Hong, 2003]. Although the source of such public keys is not explicitly defined by the authors, the available options are the ones presented in Section 2.1.2, i.e., the public keys are distributed by a trusted third party or are locally generated.

Devices identification in ANODR is assumed to be deployed during a bootstrap phase, since a device must have previous knowledge about the identifiers of possible communicating partners, and must also share cryptographic keys with these other devices. The usage of pre-shared keys implies that ANODR does not provide sender anonymity towards the destination, assuming that each symmetric key is shared only between two devices.

ANODR has several performance constraints due to the high overhead in computing, communication and storage [Yang et al., 2006]. The use of dummy traffic, broadcasting[7], intentional delaying, and data reshuffling has a negative impact in the network performance that is not suitable for low-latency applications. Furthermore, every device must try to decrypt a field of the request message with all the pre-shared symmetric keys it has stored [Seys and Preneel, 2006].

Privacy Preserving Routing (PPR)

The Privacy Preserving Routing (PPR) protocol [Čapkun et al., 2004] is a PMP for ad hoc networks. This protocol differs from the others presented in this section since it was designed for ad hoc networks that are constantly connected to a permanent infrastructure through wireless access points, i.e., a hybrid ad hoc network. The objective of PPR is to provide location privacy and unlinkability between senders and recipients against local observers, and sender anonymity towards the recipient. The design of PPR demands the access points to know both sender and destination identifiers and their location for routing messages accordingly. Therefore, access points are considered trusted in PPR.

A device has two different identifiers in a given instant of time: a transaction pseudonym and a digital certificate signed by a certificate authority. The transaction pseudonym that is generated after a keyed-hash function of the device identifier concatenated with time information. The key used in the keyed-hash function is shared secret between a device and the access point to which this device is connected to. The digital certificate is replaced by a new one every time the transaction pseudonym changes. Thus, a device obtains a set of n certificates from the certificate authority, and has to request more

[7] ANODR has a lack of termination condition causing request messages to be propagated practically into the whole ad hoc network. This is caused by the absence of a maximum propagation limit for a request message, which is usually achieved by a decreasing the time-to-live field (TTL) by 1 every hop. This is an intentional modification to prevent an attacker of knowing how many hops away is the sender of a message [Yang et al., 2006].

certificates before running out of them. The certified public keys are used to establish temporary symmetric secret keys between neighbor devices.

The performance cost of PPR depends on the update frequency of the identifiers used. Increasing the update frequency, results in more frequent request of certificates to the certificate authority [Čapkun et al., 2004].

Secure Distributed Anonymous Routing (SDAR)

The Secure Distributed Anonymous Routing (SDAR) protocol [Boukerche et al., 2004] is a reactive trust-based source routing protocol, and thus an RMP. The objective of SDAR is to provide a secure distributed path construction protocol for anonymous communication in ad hoc networks. The goal of anonymous communication in SDAR is to obtain unlinkability between sender and recipient against a local observer. In SDAR, a sender requires previous knowledge of the public key of the recipient device. The sender identity is encrypted and included in the request messages. Thus, this protocol does not provide sender anonymity towards the recipient.

SDAR is built upon dynamic trust management, broadcasts, public key cryptography, and onion routing in the return path [Goldschlag et al., 1996; Syverson et al., 1997]. In SDAR, a device is identified by a public key that is broadcasted to the neighbor devices. SDAR assumes the existence of a trusted certificate authority, outside the ad hoc network, which issues public and private keys to the ad hoc network devices. Ad hoc network devices are also identified by logical IP addresses.

SDAR requires a fresh public and private key pairs to be generated for every path discovery. In addition, path discovery messages are broadcasted basically forcing every device in the ad hoc network, that fulfills the trust requirements, to perform a public key decryption to verify if the path discovery message is intended to it or should be forwarded (the source of this problem was already discussed in ANODR). Furthermore, before forwarding a path discovery message, every device has to perform a public key encryption and a digital signature using its public and private key pair acquired from the certificate authority. Thus, SDAR has clear performance constraints due to the high overhead in computing and communication [Song et al., 2005].

MASK

MASK is a PMP [Zhang et al., 2005]. The objective of MASK is to provide sender and recipient anonymity, and unlinkability between senders and recipients against a global observer. MASK is built upon an adaptation of the pairing-based key agreement presented in [Balfanz et al., 2003], broadcasting of route request messages, random padding, dummy traffic, intentional delaying of communication data, and data forwarding through multiple paths. The

pairing-based key agreement is used to establish pairwise shared keys and link identifiers between two one-hop neighbor devices. Both the shared key and the link identifier are updated whenever a new message is sent.

In MASK, every ad hoc network device has two identifiers: a permanent identifier and a transaction pseudonym. A set of transaction pseudonyms is issued by a trusted third party to each device in the ad hoc network. The permanent identifiers of recipients are known in advance by potential senders. MASK path discovery demands route request messages to be broadcasted all over the ad hoc network. Route request messages transmit the permanent identifier of the recipient in plaintext. Thus, this identifier is known by all nodes in the ad hoc network, assuming that the route request message, which is broadcasted, is received by all other devices in the ad hoc network. Broadcasting prevents an attacker to associate the permanent identifier to a specific device in the ad hoc network. Thus, the recipient's location privacy is protected, but the recipient anonymity is compromised. MASK does not offer protection against Sybil attacks (see Section 2.2.2, on page 22). A device with multiple network interfaces can potentially have the same amount of transaction pseudonyms at a given instant of time.

MASK sends a single data flow through multiple paths in the ad hoc network from sender to recipient. However, the efficiency of such a measure in terms of anonymity is questionable and may even be harmful against multiple collaborating attackers [Reiter and Rubin, 1998]. Furthermore, the neighborhood discovery process allows malicious users to find out the amount of neighbors a target device has by evaluating the amount of pairing authentication messages sent by the target node. A local attacker can then obtain information regarding the network topology around the target device, and can eventually block messages [Berthold et al., 2000] or, if the attacker is the only neighbor of the target device, it can probe the target with requests regarding some or all identifiers in the network, forcing it to expose their real identity.

Regarding the performance of MASK, the authors compare it to the performance of AODV using simulation results presented in [Zhang et al., 2005]. However, dummy traffic and intentional delaying of communication data are apparently not included in the experiments. Furthermore, the maximum hop distance between nodes in the simulation scenario was 4, which significantly limits the amount of broadcasting needed during the route request mechanism.

Anonymous Dynamic Source Routing (AnonDSR)

The Anonymous Dynamic Source Routing (AnonDSR) [Song et al., 2005] is a PMP, and is based on the Dynamic Source Routing (DSR) protocol [Johnson et al., 2007]. The objective of AnonDSR is to provide unlinkability between sender and recipient against local and global observers. AnonDSR is built upon one-time only public keys, broadcasting and onion routing [Goldschlag et al.,

1996; Syverson et al., 1997].

In AnonDSR, every device in the ad hoc network has one unique identifier. The source of such an identifier is, however, not specified in [Song et al., 2005]. AnonDSR is a protocol suite consisting of three distinct parts: a security establishment parameter protocol, a route discovery protocol, and a data transfer protocol. The security establishment parameter protocol is used to set a shared key between a sender and a destination. The route discovery protocol uses broadcasting and one-time public keys to establish a path between a sender and a destination. Broadcasting and onion routing using symmetric key encryption are used in the data transfer protocol. The symmetric keys are shared between the sender, the recipient and each intermediate device in the path between sender and recipient. Thus, the amount of keys equals the amount of intermediate devices in the path.

The unlinkability property of AnonDSR is compromised if the attacker eavesdrops the messages exchanged in the security establishment parameter protocol because the identifiers of both sender and receiver are transmitted in plaintext [Song et al., 2005]. Thus, the anonymity set of a sender device is reduced to the devices with which it had established a shared key. In principle, the unlinkability property of AnonDSR is kept only if every device in the ad hoc network establishes a shared key with all other devices before any run of the route discovery protocol. In AnonDSR, protection against global observers requires dummy traffic [Song et al., 2005]. The effects caused by the use of dummy traffic were discussed earlier in this section.

AnonDSR requires a fresh public and private key pairs to be generated for every path discovery. In addition, path discovery messages are broadcasted basically forcing every device in the ad hoc network to perform a symmetric key decryption to verify if the path discovery message is intended to it or should be forwarded. Furthermore, before forwarding a path discovery message, every device performs a public key encryption. Thus, AnonDSR has a considerable performance constraint due to the high overhead in communication and computing that might configure a limitation for mobile devices.

Conclusions Regarding Anonymous Ad Hoc Routing

All the aforementioned protocols rely on a trusted third party for the distribution of identifiers. The identifiers used in all the aforementioned protocols are unique identifiers that are made public to allow other devices in the ad hoc network to communicate with them. This is needed since the sender device cannot know the current temporal identifier (if any exist) of the intended recipient. However, such an approach leaks information regarding the presence of a given identifier in the ad hoc network. Thus, an attacker could link two appearances of an identifier in different ad hoc networks in different locations and time periods.

MASK is the only aforementioned protocol that provides some protection against such privacy attacks, since it uses transaction pseudonyms that are issued by a trusted third party. Therefore, MASK provides unlinkability between different appearances as long as the sender is never the recipient of any message, i.e., if it does disclose its presence in the ad hoc network by exposing its permanent identifier. However, one weakness of such a model is that the trusted third party can always associate the temporal pseudonyms issue to a given device with its permanent identifiers. An extended discussion regarding this topic is presented in Section 5.

2.4.2 Anonymous Communication in Ad Hoc Networks

Overlay anonymous communication mechanisms operate over the transport layer and below the application layer. The advantage of these mechanisms is that they are independent from the routing layer since they operate on top of the transport layer. Therefore, overlay anonymous communication mechanisms may be deployed along with standardized ad hoc routing protocols. Other advantages include the possibility of decoupling transport layer information, which prevents transport layer fingerprinting.

The disadvantages, on the other hand, include the non-transparency towards the upper layer, since applications must be diverted from the normal data flow towards the overlay network, e.g., using a local proxy. Furthermore, the network performance might be worse compared to anonymous routing protocols, since messages are routed through a set of intermediary overlay nodes and a number of connections must be established before a message is finally delivered to the destination.

Mix Route Algorithm

The Mix Route Algorithm (MRA) is an overlay anonymous communication mechanism [Jiang et al., 2004]. MRA was designed to offer protection against global observers. MRA is an adaptation of the Chaumian mix concept [Chaum, 1981] to mobile ad hoc networks.

In this proposal, the devices located in the ad hoc network are divided into two sets: the Mix nodes and non-Mix nodes. Mix nodes batches and reorders data traffic to hide the correlation between the incoming and outgoing traffic, and also relies on the usage of bandwidth-consuming dummy traffic between Mix nodes. Obviously, the performance burden is greater in Mix nodes than in non-Mix nodes, since the former set has to execute all the mixing functions, and also relays more data than other nodes. Thus, MRA does not provide fairness among devices, since the workload differs quite much between Mix nodes and non-Mix nodes. In addition, Mix-based solutions heavily rely on public-key encryption, which is a major performance drawback.

Chameleon

Chameleon [Martucci et al., 2006a] is an overlay anonymous communication mechanism designed after the requirements for anonymous communication systems in ad hoc environments described in [Andersson et al., 2005b]. In Chameleon, the sender device never sends a data stream directly to the destination. The data stream is forward through a tunnel that is set using other ad hoc network devices. The underlying functionality of Chameleon is based on the anonymous path setting of the Crowds system [Reiter and Rubin, 1997]. Devices that are part of the tunnel toss of a biased coin to decide if a data stream should be finally forwarded to the destination or if it should be forwarded to yet another relay device instead, which implies extending the tunnel.

Chameleon substitutes the Crowds' blender, i.e., a centralized service that provides a directory of devices that are part of the Crowds network, for a decentralized solution that fits better to an ad hoc network. In Chameleon, any participating device can be a directory server for other devices. If a data forward tunnel is broken, e.g., due to a vanishing node in the ad hoc network, the path is repaired from the point of rupture. Chameleon privacy properties include sender, receiver, and relationship anonymity against a defined set of attackers. Chameleon is further discussed in Chapter 6.

2.4.3 Privacy in Physical and Data Link Layers

The standard identifier in the data link layer is the hardware (MAC) address. In principle, these addresses are unique and permanent identifiers that are bound to a network interface card. However, such addresses can be changed by software. Therefore, hardware addresses might not be unique and are not a permanent identifiers, but pseudonyms. Changing those hardware addresses can enhance privacy in data link layer [Gruteser and Grunwald, 2003]. Other privacy enhancements in the data link layer would require changes in implementation of wireless network card drivers to thwart, for instance, identification of gaps in the sequence numbers of frames or fingerprinting using differences in the implementation of the active scanning algorithm, in the case of IEEE 802.11 wireless network card drivers [Franklin et al., 2006].

Identifiers in the physical layer include geographical location, radio patterns, and transmission characteristics that can be associated with a given device. Physical layer privacy threats need to be individually addressed depending on the type of information gathered by an attacker. RF fingerprinting using signal to noise (S/N) ratio information can be thwarted using noise injection, for instance. However, some physical layer threats to privacy are very difficult to eliminate. For instance, to prevent transient signal detection and modulation domain techniques would require perfectly identical radio transceivers or radio transceivers that could dynamically modify their physical characteristics,

to be produced, which can greatly impact their manufacturing cost [Brik et al., 2008].

Nevertheless, physical and data link layer threats are limited to one-hop distance to the target device. Thus, one attacker relying on such mechanisms would have to either control many devices deployed over a large area or follow the target device. The former option requires a resourceful attacker that could cover a large geographical area with probe devices that could capture RF information regarding passing nodes. In the latter option, assuming a scenario where devices are mobile, the attacker needs to actively stalk the target since the radio propagation range of wireless networks is limited and the radio propagation range changes significantly depending on aspects, such as environmental conditions, geographical characteristics, transmission power, antennas, and physical characteristics of the transceiver.

In conclusion, it is feasible to enhance privacy in the data link layer, but privacy enhancements in the physical layer are difficult to achieve. Nevertheless, these threats are limited to attackers located at one-hop distance from the target device.

2.5 Summary

In this chapter, an introduction to ad hoc networks, including a classification of these networks regarding the assumptions on the availability of external services were presented. The existing security and privacy threats in ad hoc networks were identified, and the security and privacy enhancements were listed and discussed.

In the next chapter, the relationship between the existence of unique and valid identifiers and the provisioning of anonymity is presented. These two concepts are usually understood as opposites, but we argue that valid and unique identifiers are needed for the provisioning of anonymity.

Chapter 3

The Identity-Anonymity Paradox

"...Am I me?
Is Malkovich Malkovich?..."

John Cusack as Craig Schwartz
— Being John Malkovich (1999)

This chapter presents the problem of identification in ad hoc networks and its consequences to security and privacy. The remainder of this chapter is organized in four sections. The first section revisits the definition of ad hoc networks and discusses the provisioning of addressing information in such networks. The objective of the second section is threefold: it shows the connection between the absence of device identifiers in ad hoc network and Sybil attacks, it discusses the relationship between the absence of identifiers and the provisioning of anonymity properties, and it presents the current countermeasures against Sybil attacks in ad hoc networks. The third section introduces the identity-anonymity paradox by presenting the relationship between security, the absence of identifiers and the provisioning of anonymous communications in ad hoc networks. Finally, the last section identifies the consequences of the identity-anonymity paradox.

3.1 Ad Hoc Networks and Unique Identifiers

The RFC 2501 [Corson and Macker, 1999] refers to two operational modes for ad hoc networks: they may operate in isolation, or they may have gateways to and interface with a fixed network (a stub ad hoc network). Thus, in the

operational mode where ad hoc networks may operate in isolation one could assume the absence of a fixed infrastructure during a given period of time, with no central devices controlling the network or providing services such as network routing, security or logical address assignment.

The lack of standardized addressing schemes allows ad hoc network devices to dynamically change their logical and hardware addresses, i.e., IP and MAC addresses, as discussed in the Chapter 2. Moreover, ad hoc network devices can have multiple network interfaces (either real or virtual) with multiple identifiers each. Thus, obtaining unique, persistent and trustworthy identifiers using only information from the layers below the application layer (regarding the TCP/IP model) is not realistic[1]. Therefore, network services that depend on network or data link layer information for authentication purposes cannot offer any guarantees regarding the validity of such identifiers in dynamic environments such ad hoc networks.

The definition of ad hoc networks and the understanding of such a definition are a fundamental aspect in this chapter. If the definition of mobile ad hoc networks stated in RFC 2501 [Corson and Macker, 1999] is understood as that ad hoc networks must be completely set in isolation and without any previous contact with any form of infrastructure or centralized identity provider service, we argue that it is virtually impossible to guarantee the deployment of unique identification in ad hoc networks. This argument is further explained in Section 3.2.

The impossibility to set persistent and unique identifiers in ad hoc networks that are completely isolated, and thus do not have any central services that can issue identifiers, may lead to a hasty conclusion that anonymity is naturally achievable in ad hoc networks, since unique identifiers do not exist in practice below the application layer. In this chapter we expose the incorrect reasoning behind the conclusion that anonymity is naturally achieved without identifiers and show that lack of identification is harmful for the provisioning of anonymity. Furthermore, we explain that the security provisioning in ad hoc network needs unique identifiers.

3.2 The Absence of Identifiers and Sybil Attacks

The lack of reliable network and data link identification might suggest that nodes in mobile ad hoc networks are naturally anonymous, especially if we consider the Sybil attack (see Section 2.2.2 on page 22) as an enabler for achieving

[1]Nevertheless, physical layer fingerprinting (see Section 2.2.3 on page 26) can be used for identifying some consistent features in the radio pattern, and thus be used for device identification. However, such features are not guaranteed to be unique [Barbeau et al., 2006]. In addition, physical layer fingerprinting requires devices to be at one-hop distance from the device being identified and it also may require special hardware, such as oscilloscopes and spectrum analyzers, which is usually not available in every device.

anonymity. A Sybil attack is preceded by one or a small number of network devices counterfeiting multiple identities [Douceur, 2002]. Thus, in theory a device could seek anonymity by having multiple identifiers simultaneously, each with a limited and controlled lifetime that would, for instance, last only for one session and after that be discarded, i.e., each identifier is a fresh and new transaction pseudonym.

Therefore, switching network and data link identifiers frequently, i.e., IP and MAC addresses, could, in principle, enhance privacy since an adversary would be unable to associate current and past identifiers used by a device [Gruteser and Grunwald, 2003]. The goal of such an approach is to eliminate any permanent binding between the device and hardware and logical addresses. The unlinkability between different pairs of hardware and logical addresses used by a device enhance location privacy properties since such strategy does not disclose (or require) any long-term identifier. The use of short-term unlinkable identifiers as the only source of device identification is a provoking indication that ad hoc network devices are intrinsically anonymous as long as the *{IP, MAC}* pairs are frequently switched and no long-term identifiers exist.

However, the apparent benefits of having unlinkable, short-term data link and network identifiers as the only sources of device identification in ad hoc networks are not real. The use of such a technique without any other long-term identifier creates a series of security and privacy problems, as it is further explained along this section. The remainder of this section is divided in two subsections: first the disadvantages of not having long-term identifiers are discussed, and in the last part of this section, Sybil attacks and countermeasures to such attacks are presented.

3.2.1 Disadvantages of the Absence of Trusted Identifiers

In this section the disadvantages of having devices with only short-term identifiers and the absence of long-term trusted identifiers are discussed. Trusted identifiers refer to identifiers that can be acknowledged by other devices as being guaranteed authentic, unique and issued by a third trusted party. Furthermore, a trusted identifier should, in principle, be linked to one, and only one, device in the ad hoc network.

A clear disadvantage of using only short-term *{IP, MAC}* pairs as device identifiers in ad hoc networks with peer-to-peer services running is the correct identification of services located in the network. Since there is no protection against impersonation attacks, attackers can masquerade as any other device. Furthermore, some network services that intrinsically depend on the uniqueness of identifiers, such as ad hoc routing, might be disrupted if an attacker is able to generate false routing information using multiple *{IP, MAC}* pairs. Thus, from the network security perspective, the absence of trusted identifiers

blocks authentication services, and eventually reduces the availability and reliability of the network.

From the privacy perspective, the absence of long-term trusted identifiers does not provide some key privacy properties such as unlinkability between senders and recipients and sender anonymity towards the recipient. These privacy properties are not guaranteed because direct connections are established between senders and recipients [Martucci, 2006]. It is thus straightforward to identify relationships between senders and recipients and to compromise any anonymity property of the communication.

Furthermore, the absence of long-term trusted identifiers does not hinder an attacker to deploy physical layer oriented attacks (as presented in Section 2.2) or to perform traffic analysis on eavesdropped network data. Thus, the geographical location of senders and recipients can in theory be exposed and profiling of users activity may lead to device identification, even with the absence of long-term trusted identifiers.

3.2.2 Sybil Attacks and Countermeasures

A Sybil attack is preceded by one or a small number of network devices counterfeiting multiple identifiers. Thus, a single device can eventually compromise a disproportionate share of the system by controlling these multiple identifiers [Douceur, 2002].

A basic countermeasure against Sybil attacks is to limit the number of identifiers to one per device per time period. This solution can be achieved if we consider that each device has one, and only one, valid identifier in a given instant of time, which is issued by an authority or set of authorities that are trusted by all devices in the ad hoc network. Even though simple in design, this solution is not easily achievable in practice since it demands users to acquire identifiers in advance from a trusted authority, and it also requires the trusted authority to issue no more than one identifier per device.

In the absence of such trusted authorities, there are other countermeasures against Sybil attacks. Such countermeasures include resource testing (computational, communication or storage), radio resource testing, pre-distribution of random key or identifiers, remote code attestation, geographical positioning techniques, recurring costs and fees, and social networks [Douceur, 2002; Levine et al., 2006; Newsome et al., 2004]. However, such countermeasures also have disadvantages in their implementation or feasibility. Next, each of the aforementioned countermeasures is briefly explained and their drawbacks listed:

- *resource testing* — resource testing assumes that devices are limited in resources, either regarding computational power, storage (memory space) or communication capabilities. Thus, if one test demands all the resources of

a given device, a device would not be able to perform two or more complex tests simultaneously [Douceur, 2002]. This countermeasure requires the resource testing to be executed simultaneously by all the other devices. This requirement makes resource testing infeasible in ad hoc networks since devices might leave and join the network at any time. Furthermore, resource testing usually assumes homogeneous devices, i.e., with similar availability of resources. This assumption can be hardly guaranteed in a dynamic and heterogeneous environment such as ad hoc networks;

- *radio resource testing* — radio resource testing is a probabilistic test and is a variant of the aforementioned resource testing. The objective of this test is to evaluate the communication capabilities of neighbor devices, i.e., one-hop distance, and detect Sybil identifiers [Newsome et al., 2004]. In this test, a device assigns to each neighbor a different frequency channel. Thus, if a message is sent on only one channel, the neighbor device that was assigned this channel should be able to hear the message and reply on it. The underlying idea behind this test is that only one channel can be listened in a given instant of time (assuming that each device has one, and only one, radio interface). Therefore, an attacker would not be able to listen to multiple channels simultaneously. However, radio resource testing is not capable of evaluating all neighbor devices simultaneously, since only one channel can be used in a given instant of time to send the message. Thus, part of the devices remains untested. In addition, the number of channels that can be used is limited to limit the cross-interference between channels and also to adhere to local frequency allocation regulations of the radio spectrum[2]. Furthermore, radio resource testing tests only neighbors that are directly connected, i.e., at a one-hop distance. Thus, this test is not able to detect Sybil devices that are located more than one-hop distance away, or even an attacker could claim that its counterfeit identifiers are located more than one-hop away from the device that is running the radio resource test. The delegation of testing and acceptance of results produced by third party devices regarding the presence of Sybil nodes is referred to as indirect identity validation, in contrast to the direct identity validation [Douceur, 2002];

- *pre-distribution of random keys or identifiers* — pre-distribution of identifiers assumes that keys or identifiers are distributed beforehand by a trusted third party to all nodes in an ad hoc network. The trusted third party can, hence, vouch for a one-to-one correspondence between a phys-

[2]The IEEE 802.11 standard specifies up to 14 channels. The standard also specifies a minimum 25MHz of channel separation to prevent cross-interference between channels, which practically limits the IEEE 802.11 to 3 active channels [IEEE 802.11]. Simulation results in [Newsome et al., 2004] indicate that with 5 or less channels, the probability of detection of Sybil devices is basically zero.

ical device and a logical identifier [Douceur, 2002]. Pre-distribution of random keys was originally presented in the context of wireless sensor networks. There are different techniques of random key pre-distribution, such as key pools, single-spaced pairwise key distribution, and multi-space pairwise key distribution [Newsome et al., 2004]. The first scheme associates one identifier to a set of keys. The single-spaced pairwise key distribution is based on public and private information and setting of pairwise keys during a bootstrap phase. The last scheme is a combination of the two aforementioned schemes;

- *remote code attestation* — remote code attestation tries to exploit the fact that the code running in a Sybil device is probably different from the code running in a non-Sybil device [Newsome et al., 2004]. This scheme was originally presented in the context of wireless sensor networks, and may prove useful in networks with homogeneous devices running the same set of programs, such as a sensor network device. However, in networks with heterogeneous devices with heterogeneous computer architectures, running different operating systems and sets of programs, remote code attestation can hardly be useful. Furthermore, even if it is possible to attest the code used for the generation of the device identifier, it might still be possible to launch several instances of the same legitimate code and start multiple threads running the same code in order to bypass remote code attestation;

- *geographical positioning* — geographical positioning techniques try to pinpoint devices in order to verify the position of a node. The underlying basic assumption is that two distinct physical bodies of mass cannot occupy the same space at the same time. Thus, only one device must exist in a particular geographic location. RF triangulation can be used to determine the location of a device with errors in the order of meters (see Section 2.2.3 on page 25). Yet another technique is to verify if a specific geographic location has a density of devices much above than the expected average, which might indicate the presence of Sybil devices [Newsome et al., 2004]. The evaluation of geographical location patterns of clusters of identifiers that are moving together can potentially indicate the presence of a device launching a Sybil attack [Piro et al., 2006]. The use of round-trip time information and beacon devices can be used in static (i.e., non-mobile) networks to determine the geographical position of a device and, thus, the presence of Sybil identifiers [Bazzi and Konjevod, 2005]. Other physical layer techniques, such as statistical analysis of signal strength distribution can also be used to detect Sybil identifiers [Xiao et al., 2006]. The drawback of geographical positioning techniques is clear: in multi-hop wireless ad hoc networks it may be infeasible to verify the geographic position of a particular device, especially without the

aid of other devices, which might be non-legitimate devices, or might be colluding attacker devices;

- *recurring costs and fees* — recurring costs and fees try to impose a cost for creating new identifiers [Levine et al., 2006; Margolin and Levine, 2008]. The recurring costs can be monetary or non-monetary, such as passing a Turing test as a requirement for obtaining a new identifier [Awerbuch and Scheideler, 2004]. Recurring costs provide a disincentive to launch Sybil attacks by imposing a linear cost increase that is directly proportional to the number of controlled identifiers. Such an approach does not guarantee the absence of Sybil identifiers, and may not be suitable for applications that do not tolerate Sybil attacks, such as electronic voting [Douceur, 2002; Margolin and Levine, 2008]. Nevertheless, recurring costs and fees impose a potential high price to an attacker deploying Sybil identifiers;
- *social networks* — social networks are networks based on established relationships between users that trust each other not to launch a Sybil attack. Graphs of trusted identifiers can be built with graph nodes representing different identifiers and graph edges representing trust relationships. The objective of these graphs is to identify and limit the number of possible attack edges, and thus, to limit the number of Sybil identifiers deployed [Yu et al., 2006, 2008]. This approach does not, however, guarantee the absence of Sybil identifiers, and may not be suitable for applications that do not tolerate Sybil attacks. Furthermore, the applicability of solutions based on social networks is limited to cases with significant overlap between real-world social networks and participants in an online application, as discussed in [Margolin and Levine, 2008]. Such a limitation is a result from a different understanding of the term trust, and its implications, in social networks and in the real-world.

In conclusion, only one of the aforementioned countermeasures can actually reliably prevent Sybil attacks: the pre-distribution of random keys or identifiers. A trusted authority or set of trusted authorities that issue one, and only one, identifier to each device in the ad hoc network can prevent Sybil attacks [Douceur, 2002]. The other countermeasures either are not suitable for the characteristics of ad hoc networks or can at most be a disincentive for the deployment of Sybil identifiers. This chapter follows with the introduction of the identity-anonymity paradox in the next section.

3.3 Defining the Identity-Anonymity Paradox

There is an apparent conflict of interests between security and privacy in ad hoc networks regarding the absence of identifiers, uniqueness of identification

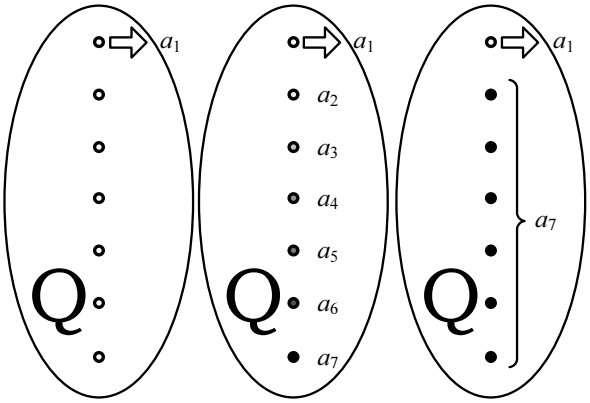

Figure 3.1: This figure illustrates a given anonymity set Q *from three different perspectives. The leftmost set shows the set* Q *from the point-of-view of an outsider or an honest user* a_1 *that had joined such a set, where all the participants of the set are indistinguishable. The figure located in the middle of the figure depicts the configuration of such a set as expected by* a_1*, where each other element of the set corresponds to a different user, referred to as* a_2 *to* a_7*. The rightmost figure depicts the anonymity set from the perspective of the Sybil attacker* a_7*. The attacker contributed with* $(n-1)$ *identifiers to the set* Q*, and, hence, compromises the anonymity properties of the user* a_1*, who is oblivious of the Sybil attack.*

and protection against Sybil attacks. Uniqueness of identification is the ability to associate one logical identifier to one physical device at a specific instant of time. The absence of identifiers prevents the linking of a physical device with a specific logical identifier at a particular instant of time and thus protects privacy. On the other hand, unique and trusted identifiers are required for authentication services, for detection and prevention of Sybil attacks, and for the provisioning of security in ad hoc networks in general.

Unique identification of network devices is required to prevent basic network services, such as ad hoc routing, and other applications, such as electronic voting or reputation systems, to be compromised or disrupted by one or few malicious users deploying Sybil attacks. Thus, achieving privacy with the absence of identifiers is not optimal, since authentication services require trusted identifiers by definition.

The definition of anonymity states that a subject is anonymous if it cannot be sufficiently identified within an anonymity set, from an attacker's perspective [Pfitzmann and Hansen, 2008]. Thus, anonymity properties require confidence in the validity of the identifiers that constitute the anonymity set. Thus, in the absence of trusted identifiers a Sybil attacker can poison an anonymity

set if it contributes to $(n-1)$ identifiers, of the overall n identifiers existing in such a set. The Figure 3.1 depicts a Sybil attack poisoning an anonymity set.

The complete absence of trusted identifiers is thus not a solution for achieving privacy in ad hoc networks, since it is not capable of guaranteeing a device the ability of not being identifiable within an anonymity set, since the anonymity set is not reliable. The provisioning of privacy therefore depends on the construction of a trusted anonymity set, i.e., an anonymity set that guarantees that each identifier corresponds to one, and only one, physical device. Privacy properties, such as anonymity, can then be achieved with privacy enhancing technologies, such as anonymous communication mechanisms.

Anonymous communications mechanisms require a set of trusted and unique identifiers to be able to define an anonymity set. Likewise, security models for ad hoc networks also require trusted and unique identifiers to prevent the deployment of Sybil attacks and thus can provide an anonymity set that is free of Sybil nodes. Therefore, network security is required for the provisioning of anonymity. Furthermore, the provisioning of anonymity needs trusted and unique identifiers.

In conclusion, even though the concepts of anonymity and identification are often understood as two opposites and conflicting ideas, trusted identification is required for guaranteeing the security properties of the anonymity set, i.e., an anonymity set free of Sybil identifiers. Thus, since these two concepts initially seem to contradict each other, the need for trusted identification for the provisioning of anonymity is called the *identity-anonymity paradox*.

3.4 Identity-Anonymity Paradox Consequences

The identity-anonymity paradox points out a direct relationship between privacy and security regarding the need for trusted identifiers. Trusted identifiers are thus necessary for both security and privacy. The underlying reason of such a paradox is the Sybil attack, i.e., the possibility of one device controlling multiple logical identifiers. The identity-anonymity paradox leads to a new interpretation of the basic assumptions regarding the independence of ad hoc networks operating in isolation in RFC 2501 [Corson and Macker, 1999] on one hand, and security and privacy requirements on the other hand.

When analyzing the definition of ad hoc networks included in the RFC 2501, the classification of ad hoc networks presented in Section 2.1.2, the security and privacy enhancements in ad hoc networks presented in Sections 2.3 and 2.4, and the identity-anonymity paradox, a list of conclusions can be devised:

- the total absence of trusted identifiers does not guarantee some fundamental privacy properties in ad hoc networks. Privacy properties such as relationship anonymity and sender anonymity are not fulfilled. Furthermore, the absence of trusted identifiers makes detection of Sybil attacks

in ad hoc networks much harder, if not impossible, in the absence of a trusted third party, as discussed in Section 3.2.2. Other basic network services, such as secure ad hoc routing, also require device identification. Other disadvantages regarding the absence of identifiers were listed in Section 3.2.1;

- security schemes for ad hoc networks demand some sort of device identification for the detection of Sybil attacks. Identification is achieved with trusted identifiers, such as digital certificates. To be trusted, these identifiers must be issued by a device that is believed to be trusted by all other ad hoc network devices, such as a third trusted party, that can be either centralized or distributed, as presented in Section 2.3;

- regarding the definition of the operational mode in isolation of ad hoc networks mentioned in RFC 2501, we can conclude that if security is a requirement, the total and complete isolation of all devices from any infrastructure is not possible. If security and privacy are required, trusted identification is a prerequisite. As trusted identification requires previous contact with a trusted device, such as a certification authority, total and complete isolation from any infrastructure[3] is not compatible with the security and privacy requirements. Thus, if security and privacy are a requirement, the passage regarding the operational mode in isolation from the hardwired network, in RFC 2501, should not be interpreted as that the ad hoc network is isolated from any infrastructure during the entire lifetime of the devices that are part of such an ad hoc network. Isolation should not imply the absence of all trust relationships, or the nonexistence of a trusted third party. An ad hoc network can, of course, be constructed in isolation from any infrastructure, but it should not mean that the devices that are part of such an ad hoc network do not have an existing trust relationship.

The aforementioned list of conclusions is based on the need for security and privacy in ad hoc networks. Clearly, if security and privacy are not requirements, the understanding of the operational mode in isolation can mean total isolation from any kind of infrastructure during the entire lifetime of all devices that are part of such an ad hoc network. Obviously, if network security is not guaranteed, the functionality of the ad hoc network can be compromised by a small number of malicious devices, as mentioned in Section 2.2. Moreover, according to the identity-anonymity paradox presented in the Section 3.3, there are no privacy guarantees in the absence of trusted identifiers.

[3]A trust third party can certainly be a device deployed in the ad hoc network, but since it has a special and unique role for the functionality of the network, we refer to it as part of the network infrastructure.

3.5 Trusted Identification and Unlinkability

One intrinsic problem of long-term trusted identifiers is the linkability between different shows of the same identifier, i.e., if the same identifier is presented several times, it is possible to profile the different locations and instants of the appearances of such identifiers. Thus, preventing the linkability of distinct appearances of long-term trusted identifiers is a privacy issue.

The solution is to turn long-term trusted identifiers into short-term trusted identifiers that have the functionality of transaction pseudonyms. There are basically two different strategies for achieving both unlinkability and trusted identification based on that solution: the trusted third party issues a chain of one-time only certificates instead of only one long-term certificate; or a single long-term certificate is issued and this certificate is used to generate other short-term certificates that are unlinkable between each other and also unlinkable to the long-term identifier. A basic problem of such solutions is how to limit or how to make it possible to detect if a device uses more than one certificate at a single instant of time. Hence, it must be possible to prevent such mechanisms to be used to launch Sybil attacks. In Chapter 5 we further discuss this problem, present existing solutions, and introduce a framework for providing trusted identification, unlinkability between identifiers and detection of Sybil attacks.

3.6 Summary

In this chapter the relationship between the absence of trusted identifiers in an ad hoc network and the Sybil attack have been presented. Countermeasures to Sybil attacks were listed and the disadvantages of the absence of trusted identifiers were enumerated. The intrinsic relationship between the need for trusted identifiers and the provisioning of anonymity was introduced and such a relationship was called the identity-anonymity paradox. Finally, a linkability problem regarding long-term trusted identifiers and the profiling of multiple appearances of such long-term identifiers was presented. Some solutions for the problem of linkability of long-term identifiers and the risks created by such solutions regarding the deployment of Sybil attacks were briefly listed.

In the following chapter, the requirements for security and privacy in ad hoc networks regarding the need for protection against Sybil attacks and the different categories of privacy-friendly identifiers are reviewed. Moreover, a set of requirements for anonymous communication mechanisms in ad hoc networks is introduced.

Chapter 4

Security and Privacy Requirements for Ad Hoc Networks

אֶֽהְיֶה אֲשֶׁר אֶֽהְיֶה

— *Exodus 3:14*

This chapter reviews the requirements for security and privacy in ad hoc networks. Most of the requirements listed in this chapter were already mentioned in Chapters 2 and 3. The objective of this chapter is to outline the security and privacy requirements that are the basis for the design of privacy-friendly identifiers and for defining the trade-offs between the offered degree of anonymity and the network performance parameters, such as end-to-end delay, for anonymous communication mechanisms that are suitable for ad hoc networks. The former objective is further discussed in Chapter 5, and the latter objective is the focus of Chapters 6, 7, and 8 through the Chameleon protocol. The remainder of this chapter is divided in two sections: the first section outlines the network security requirements in ad hoc networks and the last section summarizes the privacy requirements for anonymous communication mechanisms and privacy-friendly identifiers.

4.1 Security Requirements

In general, the security requirements for ad hoc networks are the same for networks connected to an infrastructure in terms of data confidentiality, integrity

and availability. This section focuses on the security requirements that are specific to ad hoc networks and are relevant for designing and constructing of privacy-friendly identifiers and for setting up the foundations for the deployment of anonymous communication mechanisms.

Ad hoc networks were, in Section 2.1.2, categorized in three distinct groups regarding assumptions on the availability of external services and the conditions of such availability: intermittently connected to an established infrastructure, one or more privileged devices in the ad hoc network, and fully independent and self-organized ad hoc networks. This classification was reapplied in Section 2.3 in the context of the way identifiers are generated, obtained, and, eventually, transferred in such networks.

The importance of such classification is related to the need of trusted identifiers and the source of such identifiers. Trusted identifiers are required in ad hoc networks for the detection and prevention of Sybil attacks [Douceur, 2002], as concluded in Section 3.4. The detection and prevention of Sybil attacks is fundamental for achieving both network security and privacy, since Sybil attacks can be used both as a general attack vector to disrupt services running in an ad hoc network, such as routing, and to poison anonymity sets with Sybil identifiers, as presented in Chapters 2 and 3. Therefore, trusted identifiers are a security requirement in ad hoc networks.

Yet another conclusion from Section 3.4 is that trusted identifiers are obtained from trusted devices. Therefore, from the three possible sources for identifiers listed in Section 2.3, only the first two sources can provide trusted identifiers and, thus, the detection and prevention of Sybil attacks. These two sources are: ad hoc network devices that are intermittently connected to an established infrastructure, which host one or more issuers of trusted identifiers, and ad hoc networks with one or more privileged devices that can issue trusted identifiers to other devices.

In conclusion, trusted identifiers are needed, and such identifiers can be obtained either from ad hoc network devices that are intermittently connected to an established infrastructure, which host one or more issuers of trusted identifiers, or ad hoc networks with one or more privileged devices that can issue trusted identifiers to other devices. The detection and prevention of Sybil identifiers are not guaranteed for fully independent and self-organized ad hoc networks.

4.2 Privacy Requirements

The privacy requirements for ad hoc networks can be divided into requirements for anonymous communications mechanisms and requirements for privacy-friendly identifiers. This section is therefore divided in two parts: the first discusses general recommendations for anonymous communication mechanisms

and the second summarizes the requirements for privacy-friendly identifiers assuming that the aforementioned security requirements are fulfilled, i.e., that a trusted authority exists and it is able to issue trusted identifiers.

4.2.1 Anonymous Communication Mechanisms

Considering the characteristics of ad hoc networks presented in Sections 2.1 and 2.2, it is possible to list some general recommendations for anonymous communication mechanisms in ad hoc network environments in terms of anonymity properties, fairness, network performance, network architecture, mobility, and scalability. An anonymous communication mechanism shall [Andersson et al., 2005b]:

- *provide strong anonymity properties* — the anonymous communication mechanism shall provide adequate privacy protection against malicious users, such as local and global attackers;
- *fair distribution of workload among the participating devices* — no participating device should be required to spend more resources, such as computational, network or battery resources, than others devices to obtain the same quality of service, regarding the level of privacy obtained. Nevertheless, incentives should be given to devices that accept an disproportional share of the total workload;
- *provide acceptable performance* — the anonymous communication mechanism should avoid, for instance, unnecessary data transmissions in the wireless interface, to save the battery power drained by such process, and minimize the use of expensive or complex cryptographic operations, in terms of required computational power;
- *employ a peer-to-peer model during its operational phase* — following the characteristics of ad hoc networks presented in Sections 2.1 and 2.2, ad hoc networks should work independently of online services that are constantly reachable and available. Thus, a peer-to-peer service model is more adequate for anonymous communication mechanisms in ad hoc networks;
- *handle a dynamic topology* — in an ad hoc network, nodes might unpredictably enter or leave the network. Such behavior might certainly impact the functionality and performance of anonymous communication mechanisms that set paths in the ad hoc network. An anonymous communication mechanism that is designed for such networks, shall be able to handle a dynamic topology with minimum impact on the users' privacy and network performance, and;

- *provide good scalability* — the anonymous communication mechanism shall work regardless the number of participating devices.

The items in the aforementioned requirement list are orthogonal since each item of the list addresses a specific objective of an anonymous communication mechanism designed for ad hoc networks. However, some of the requirements might conflict with others, depending on the resources used to implement them, e.g., privacy protection against global observers usually requires the use of dummy traffic, which reduces the amount of user data transmitted per packet in the ad hoc network, and thus, reducing the overall performance of the network.

Even though the items in the list are orthogonal, it is not possible to affirm that all items have the same weight in the evaluation of an anonymous communication mechanism. Different application scenarios have different acceptable trade-offs regarding the aforementioned list of recommendations. Thus, the aforementioned list should not be used for evaluation purposes without a close inspection of the application scenario on which the anonymous communication mechanism is going to be deployed.

4.2.2 Privacy-Friendly Identifiers

The main requirements for privacy-friendly identifiers is that they need to be constructed to provide unlinkability between different shows by a specific user. Such functionality can be achieved with either short-term or long-term trusted identifiers issued by a trusted third party, such as a certification authority.

Privacy-friendly short-term identifiers are, for instance, chains of unlinkable one-time only unique certificates, i.e., certificates that should be used only once, that are discarded after being used. The main advantage of short-term identifiers is the simplicity of the solution. A device just needs to obtain such chains from a trusted third party and request more certificates every time it runs out of identifiers. However, the disadvantages are threefold. First, a device needs to store and manage the chain of one-time only certificates. Second, every time it runs out of certificates the device needs to contact a trusted third party, which might not be always available. Third, it might allow the use of multiple certificates at once. The last disadvantage can lead to the deployment of Sybil identifiers in an ad hoc network, as presented in Section 3.5.

Privacy-friendly long-term identifiers, on the other hand, are unique and trusted identifiers issued by a trusted third party, such as a certification authority, and are expected to be used multiple times. A fundamental requirement for privacy-friendly long-term identifiers is that it needs to provide unlinkability between different shows of the same identifier as discussed in Section 3.5. An advantage of such solution is that certificates are issued only once, during an initialization phase of the device preferably. Moreover, in such so-

lution there is no need to contact the trusted third party after the long-term identifier is obtained, unless for eventually reporting attacks carried out by devices holding privacy-friendly identifiers, as further discussed in Chapter 5. The disadvantage of such solution is that it cannot be implemented with standard X.509 digital certificates [ITUT X.509] and it requires the design of new privacy-friendly identifiers.

Therefore, privacy-friendly long-term identifiers are preferable over privacy-friendly short-term identifiers in ad hoc networks, since they do not depend on the continuous presence of a trusted third party and they can also be designed to offer protection against Sybil attacks.

4.3 Summary

In this chapter we reviewed the security and privacy requirements for ad hoc networks, including the need of Sybil-free anonymity sets, and requirements for privacy-friendly identifiers. We concluded that long-term privacy-friendly identifiers are more suitable since the presence of a trusted third party is only obligatory during the set up phase of a device. Moreover, certificate chains of one-time short-term identifiers obtained from a trusted third party can be used to launch Sybil attacks, if several identifiers are used at once. The chapter also listed a set of recommendations for anonymous communication mechanisms for ad hoc networks.

In the following chapter we will introduce a framework for providing trusted identification, unlinkability between identifiers and detection of Sybil attacks. We will show that short-term identifiers can be generated from long-term identifiers and it is possible to detect if a malicious device issues two or more short-term identifiers to be used in the same application or context in an ad hoc network.

Chapter 5

Self-Certified Sybil-Free Identifiers

"It's always the same fish, isn't it?"
"I don't know, I have trouble recognizing the fish."
"What does the fish remind you of?"
"Other fish."
"And what does other fish remind you of?"
"Other fish."

Major Sanderson and Capt. John Yossarian
— Catch 22 (1961), Joseph Heller

This chapter presents a framework for the provisioning of identifiers that are bound to a group and are Sybil-free and self-certified, i.e., they are issued and locally signed by the device that holds it and supports the detection of devices that issue more than one identifier in a given group. Moreover, this framework provides unlinkability between different identifiers issued to different groups by the same device. The objective of the chapter is to present this framework and the self-certified Sybil-free identifiers.

The remainder of this chapter is organized in four sections. Section 5.1 outlines the solution and introduces the main components of the framework for the provisioning of self-certified Sybil-free identifiers. The objective of Section 5.2 is twofold. First, it presents the basic structure of the self-certified Sybil-free identifiers. Second, it describes the algorithms used in this framework for generation, use, and revocation of such identifiers. Section 5.3 provides the security analysis of such a framework. Finally, the related work and other ap-

plications that potentially could benefit from self-certified Sybil-free identifiers are presented in Section 5.4.

5.1 A Self-Certified Sybil-Free Framework

In this section we first define the key terms for describing the framework around the self-certified Sybil-free identifiers. Then, we provide an overview of the solution, the underlying assumptions, the framework objectives, and the attacker model. The objective of the framework proposed in the chapter is to construct sets of identifiers that provide detection of Sybil identifiers. A Sybil identifier is defined as every i^{th} identifier generated by a single device to be used in an instantiation or context of a given application, where $i > 1$, $\forall i \in \mathbb{N}$.

The elements of the sets constructed using the framework proposed in this chapter are privacy-friendly identifiers. Such identifiers are referred to as privacy-friendly because two or more locally generated identifiers, issued for distinct instantiations or contexts of a given application, are not linkable, i.e., it is not possible for an attacker to distinguish whether two identifiers from two distinct sets of identifiers are related or not. We refer to those sets as identity domains in the remainder of the chapter.

5.1.1 Identity Domains

In brief, self-certified Sybil-free identifiers are pseudonyms that are bound to one, and no more than one, identity domain. A pseudonym is an identifier of a subject other than one of subject's real names [Pfitzmann and Hansen, 2008]. Self-certified Sybil-free identifiers can be used in different applications and for different tasks. Therefore, the pseudonym class will depend on the purpose of the application. The proposed identifiers can be used as transaction pseudonyms in electronic voting applications or as role pseudonyms in applications that implement reputation schemes.

The term identity domain was coined to represent a collection of distinct identifiers. An identity domain corresponds to the set of all possible subjects, i.e., participating devices, of an instantiation or context of a given application. Therefore, in the context of the self-certified Sybil-free framework, the role of an identity domain is to define an anonymity set, i.e., a set of identifiers in which a user is not identifiable, as presented further in the chapter.

A device has as many pseudonyms as the number of identity domains that the device is enrolled with. An identity domain is used to uniquely specify the application and its context, i.e., characteristics of such application in which a set of identifiers is used, such as name, location and validity time.

Identity domains can be either secure or insecure. A secure identity domain provides an environment absent of Sybil identifiers, while an insecure identity

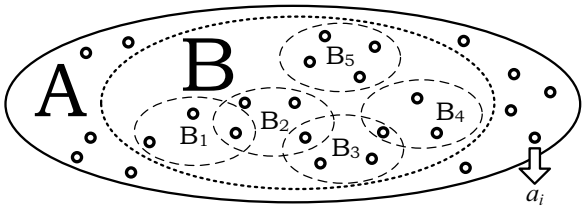

Figure 5.1: The relationship $B_n \subseteq A, \forall n \in \mathbb{N}^*$ presented in Equation 5.1 is illustrated in this figure that highlights five possible subsets (B_1, B_2, B_3, B_4, and B_5) of the set $A = \{a_1, \ldots, a_n\}$. The circles represent the elements a_i of the set A.

domain does not provide mechanisms for detection of Sybil identifiers. The framework described in the chapter requires the setting of an initial secure identity domain A, i.e., an identity domain that is free of Sybil identifiers.

The framework provides the propagation of the Sybil-freeness property of the initial secure identity domain[1] A to arbitrary many (n) subsets B_n of A:

$$B_n \subseteq A, \forall n \in \mathbb{N}^* \tag{5.1}$$

The sets $B_n, \forall n \in \mathbb{N}^*$ are included in a superset B. The relationship between the sets A, B and the subsets B_n is illustrated in Figure 5.1. In the set B are included subsets of the set A, excluding the empty set ($\varnothing$). Hence, the maximum cardinality of the set B is:

$$|B| = \sum_{i=1}^{|A|} \binom{|A|}{i} = 2^{|A|} - 1 \tag{5.2}$$

Pseudonyms are produced from a long-term identifier obtained from the trusted third party. Nevertheless, pseudonyms generated for different sets B_i are unlinkable and are only valid for the set B_i that they were created for. These locally produced unlinkable identifiers are called self-certified Sybil-free pseudonyms, since the pseudonyms are issued by the device that will own them, and they allow the detection of Sybil identifiers in an identity domain. These pseudonyms are further presented in Section 5.1.3.

The aforementioned context of an identity domain is used to uniquely identify distinct identity domains. The context of an identity domain, z, is defined by the device that creates the identity domain, i.e., the domain initiator. Any device can create new identity domains and be a domain initiator. The z information is used as a unique identifier.

[1]The term propagation is used in this dissertation to indicate that one or more properties from a given set are being passed along to subsets obtained from the original set.

```
<ctx>
  <applicat>   Application Name   <\applicat>
  <valid_fr> 2009-03-07 12:00 GMT <\valid_fr>
  <valid_to> 2009-22-06 12:00 GMT <\valid_to>
  <location> SE65188, KAU, Sweden <\location>
  <rand_non> 0F59765E74D3E67C1A2E <\rand_non>
  <init_pbK>      Public Key      <\init_pbK>
<\ctx>
```

Figure 5.2: Example of a context information z. This hypothetical z information has 6 fields: the application name, starting time, expiration time, the location, and a random nonce, which is used to increase the entropy of the context information to prevent accidental collisions of identity domain identifiers, and the public key associated with the identity domain initiator's self-certified pseudonym generated for this identity domain.

The context information z of an identity domain should specify the unique characteristics of such set, e.g., validity time, purpose, and application. Thus, some identity domains can be either short-lived or long-lived, depending on the validity time set to the identity domain. For instance, the lifetime of an identity domain used for an election is limited to a couple of hours or days, and thus, is a short-lived identity domain. Long-lived identity domains, such as an identity domain of a discussion group, are not limited in their lifetime. As a heuristic[2], the context information z of an identity domain should follow some kind of URI-like (Uniform Resource Identifier) scheme, since this is a de-facto standard that provides a simple and extensible means for identifying resources in a network environment [Berners-Lee et al., 2005]. Moreover, a short-lived identity domain with context information z must include the identity domain's validity time. Long-lived identity domains do not have a limited validity time.

The context information may contain a user-friendly name for the identity domain and other information, such as the public key of the domain initiator, or a contract that all users who join the identity domain should agree on. From a practical point of view, there is no limit on the size of z. It can be hashed down to a constant size value before being used in the cryptographic algorithms. Appending the hash to the validity time makes the uniqueness of z independent from the collision resistance of the hash function. Figure 5.2 illustrates an example of a context information z written in XML. Moreover, the uniqueness of the domain identifiers can be guaranteed under three conditions:

- a device must keep a list of all identity domains that it belongs to and removes records from the list only if the corresponding identity domains have expired. Only short-lived identity domains can expire. Identity do-

[2]A heuristic is a set of rules or hints to aid discovery or invention [Chalmers, 1999].

mains are identified according to their context information z;

- a device must only join identity domains once and such identity domains must have a valid z information, i.e., that have not yet expired. Thus, a device must check its list of identity domains that it belongs to before producing a new self-certified Sybil-free pseudonym. This is required to prevent honest devices to be lured to produce more than one pseudonym for an identity domain. Producing more than one pseudonym for an identity domain can be confused with an attempt to deploy Sybil identifiers and can eventually result in the identification of the device;
- a device must not turn back its clock. Turning back the clock could result in a device joining an expired identity domain since such expired identity domain might be valid again from the perspective of the device that turned back its clock. Joining an expired identity domain can result in a device producing more than one pseudonym for that identity domain, if it had joined before and later removed it from its list of identity domains that it is part of. Moreover, joining an expired identity domain can also result in the degradation of privacy properties since the set is reduced to the device that joined the expired domain and eventually, an attacker.

There is no theoretical limit for the number of identity domain identifiers z that can be associated to a set B_i. If the set $Z = \{z_1, \ldots, z_n\}$ is defined as the set of all possible identity domain identifiers, it is possible to affirm that there exists a surjective function f that relates the set Z to the set B of all possible subsets of A, i.e., for every element $b = B_i$ of the codomain B, there is at least one element z of the domain Z such that $f(z) = b$. Thus:

$$f : Z \rightarrow B \tag{5.3}$$

The function f is surjective since one element of the set B, B_i, can have one or more identity domain identifiers z associated with it. This function is illustrated in Figure 5.3. For instance, two different applications, each one with its own z identifier, can have as clients the same set of identifiers B_i. The unlinkability property provided by the self-certified Sybil-free pseudonyms guarantees that it is not possible for any observer to associated the elements of the set Z to the elements of the set B, or affirm that two elements of the set Z are associated with a same element of the set B, as further presented in Section 5.1.3.

The identity domain initiator does not need to be trusted by the other devices. Thus, any device (or several devices) can initiate an identity domain. Nevertheless, a trusted third party can assume the role of identity domain initiator for some applications in which the domain initiator ideally should not or must not be part of the identity domain, such as a voting application.

The identity domain initiator does not possess any control over the devices that join the identity domain, and cannot prevent any other device from joining

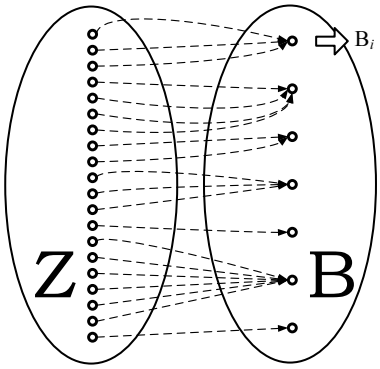

Figure 5.3: This figure represents the function $f : Z \rightarrow B$ presented in Equation 5.3. Every identity domain identifiers z, i.e., the elements of the set Z, is associated with one or more elements B_i of the set B.

it. Moreover, any device can disseminate the context information z to other devices that may or may not join this identity domain. Furthermore, the identity domain initiator cannot arbitrarily tear down an identity domain once it had been created, and, thus, this identity domain will remain existing until it is has reached its validity time, if any.

The integrity of the context information z is kept by signing it using the temporary public key generated associated with the domain initiator's self-certified pseudonym, which is part of the context information z. The signature is appended to the context information. In any case, changing the parameters of a context information z means, in practice, to setup a new identity domain.

5.1.2 Membership Certificates and Trusted Third Party

Access to the trusted third party, such as a certificate authority, is required only for acquiring a membership certificate a_i from which self-certified Sybil-free pseudonyms are produced from. The set of all issued membership certificates is referred to set $A = \{a_1, \ldots, a_n\}$, where n is equal to the number of unique devices that acquired a membership certificate from a trusted third party. Thus, the trusted third party is used for establishing the initial Sybil-free identity domain A.

A user that wants to benefit from self-certified Sybil-free pseudonyms first enrolls with a trusted third party to acquire exactly one unique membership certificate a_i, where $i \in \mathbb{N}^*$. A membership certificate is used for issuing m distinct self-certified Sybil-free pseudonyms for arbitrarily many m distinct identity domains as further clarified in Section 5.1.3. Moreover, these generated pseudonyms cannot be linked back to the membership certificate that was used

to produce them, not even by the trusted third party.

5.1.3 Self-Certified Sybil-Free Pseudonyms

Every self-certified Sybil-free pseudonym is bound to an identity domain, thus, each device can have at most one self-certified Sybil-free pseudonym per identity domain. Therefore, if a device with membership certificate a joins a set X of distinct identity domains identifiers z, where $X \subseteq Z$, there exists a bijective function $g(x)$, i.e., a one-to-one correspondence, that relates the set X to a set $P = \{p_1, \ldots, p_n\}$ of self-certified Sybil-free pseudonyms generated by this device such that $g(x) \leftrightarrow p$ for $p \in P$ and $x \in X$. The bijective function $g(x)$ is presented in Equation 5.4 and illustrated in Figure 5.4.

$$g : X \leftrightarrow P \mid X \subseteq Z \tag{5.4}$$

Moreover, all possible identity domain identifiers are bound to distinct sets of pseudonyms Q_i. The elements of a set Q_i are self-certified Sybil-free pseudonyms p_i generated from distinct membership certificates a_j. Since the self-certified Sybil-free pseudonyms are unique, the intersection of any two sets Q_i and Q_j is always the (empty) set $\varnothing$, as long as $i \neq j$. Therefore, the union of all sets Q_i must contain all self-certified Sybil-free pseudonyms. This union set of subsets Q_i is referred to as set Q:

$$Q = \bigcup_{i=1}^{n} Q_i \quad \text{where } n = |Z| \text{ and} \tag{5.5}$$

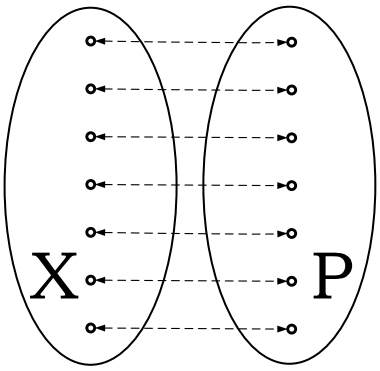

Figure 5.4: This figure represents the function $g : X \leftrightarrow P \mid X \subseteq Z$ presented in Equation 5.4. It illustrates that each identity domain z that a given device joins there is one, and only one, pseudonym p associated with it.

$$Q_i \cap Q_j = \varnothing \mid i \neq j, \forall i, j \in \mathbb{N}^* \tag{5.6}$$

The cardinality of the set Q is equal to the cardinality of the set Z since there is a set Q_i for every identity domain identifier z_i. Furthermore, there exists a bijective function h that relates the set Z to the set Q such that $h(z_i) \leftrightarrow Q_i$. Thus:

$$h : Z \rightarrow Q \qquad \therefore \qquad h^{-1} : Q \rightarrow Z \tag{5.7}$$

The composition $(h^{-1} \circ f)$ of functions h^{-1} (Equation 5.7) and f (Equation 5.3) is a surjective function that relates the set Q, whose elements are sets Q_i of self-certified Sybil-free pseudonyms, to the set B of all possible subsets of the initial secure identity domain A. Thus:

$$h^{-1} \circ f : Q \rightarrow B \tag{5.8}$$

Each subset Q_i of set Q can be mapped to one set B_i. In addition, a set B_i corresponds to one or more elements of the set Q. Moreover, assuming that a device that possess a membership certificate generates at most one pseudonym per identity domain identifier z, each element of a subset Q_i has a one-to-one mapping to the elements of a set B_i.

Moreover, if we consider only the produced self-certified Sybil-free pseudonyms, i.e., the pseudonyms that were so far generated by all participating devices for all existing identity domains, there exists a set $Q' \subseteq Q$ of all possible sets Q'_i that contain all generated pseudonyms. Naturally, the set Q' is equal to the union set of all sets P_j containing all the generated pseudonyms from distinct membership certificates $a_j \in A$:

$$Q' = \bigcup_{i=1}^{n} Q'_i = \bigcup_{j=1}^{m} P_j \quad \text{where } n = |Z| \text{ and } m = |A| \tag{5.9}$$

The self-certified Sybil-free pseudonyms are produced through a mechanism of self-certification. This mechanism uses different cryptographic building blocks and primitives, such as anonymous credentials and group signatures, for generating an arbitrary number of pseudonyms from one initial identifier, the membership certificate a_i, which is obtained from a trusted third party.

The aforementioned membership certificates are required for the bootstrapping of the initial Sybil-free domain as presented in Section 5.1.2. The generation of the self-certified Sybil-free pseudonyms also produces a certificate $cert_{(a,z)}$ associated with the self-certified Sybil-free pseudonym that has the following uses:

- to bind a freshly generated public key to the self-certified Sybil-free pseudonym. This operation is similar to the binding of public keys to X.509 certificates [ITUT X.509];

- to verify the self-certified Sybil-free pseudonym and its binding to the aforementioned public key, and;
- to disclose the device identifier obtained from the trusted third party and revocation of its certificates, if this device creates more than one self-certified Sybil-free pseudonym, and, thus, more than one certificate, for a given identity domain z.

Periodic n-times spendable e-tokens [Camenisch et al., 2006] are used as a base for the instantiation of the self-certified Sybil-free pseudonyms described in this chapter. Nevertheless, there are other cryptographic primitives that can be used to create such pseudonyms. Further details regarding the cryptographic building blocks and primitives are described in Appendix A.

In the remainder of this chapter we refer to the self-certified Sybil-free identifiers as self-certified Sybil-free pseudonyms, or just pseudonyms, to simplify the notation and also increase the accuracy of the definition.

5.1.4 Objective and Assumptions

In this section we formalize the objectives of the self-certified Sybil-free pseudonyms and also summarize the assumptions. The objectives of the self-certified Sybil-free framework are:

- to conceal the relationship between a device's membership certificate a_j and the set P of distinct pseudonyms generated from this membership certificate. Thus, an attacker cannot link a domain identifier z from set Z to a specific device or find out the set X, of the enrolled domain identifiers, associated with a device;
- to provide unlinkability between the elements p_i of the set P. Therefore, in this framework, it is not possible for any device to make explicit the relationship between the sets Z, of z identity domain identifiers, and B, of all possible subsets of the secure identity domain, as long as $B_i \subsetneq A$[3], and;
- to prevent the deployment of Sybil identifiers in a set Q_i of pseudonyms. Therefore, the owner of a membership certificate $a_i \in A$ is allowed to have one, and only one, pseudonym p in a set Q_i, i.e., per identity domain z_i. If there is more than one pseudonym p associated only one membership certificate $a_i \in A$, it must be possible to detect it.

Regarding the assumptions, parts of them were already presented in Section 5.1.1. In this section, these aforementioned assumptions are summarized and complemented. The self-certified Sybil-free framework assumes that:

[3]For the case B_i is equal to A, it is possible for an attacker to affirm that every element of A is represented in the set B_i, and, thus, is possible to link B_i to a z identity domain identifier. In any case, the anonymity set remains being the set A.

- a trusted third party is able to establish the initial secure identity domain A by distributing no more than one membership certificate per participating device, i.e., to setup an identity domain absent of Sybil identifiers;
- the identity domain identifiers z are unique, and a participating device keeps a list, i.e., the set X, of all valid, i.e., non-expired, identity domains that it had joined. Short-term identity domains x_i that had expired must be removed from the set X. Furthermore, a device must not turn back its clock, and;
- all devices are capable of performing the necessary cryptographic functions mentioned in Section 5.2.

5.1.5 Attacker Model

In the context of the self-certified Sybil-free framework, a malicious user has the following objectives:

- for a given identity domain, with an identity domain identifier z_i, where $z_i \in Z$, an attacker wants to deploy multiple Sybil identifiers in the set Q_i of pseudonyms that is associated with z_i and remain undetected, i.e., the result of this operation should not be noticeable to the other devices that also joined the identity domain z_i. Therefore, an objective of the attacker is to turn the function g, in Equation 5.4, into a non-injective and surjective function g', i.e., there are one or more generated self-certified pseudonyms by a single membership certificate a_i that can be mapped to an identity domain identifier z. Thus, the attacker objective is to construct the surjective and non-injective function g' and remain undetected:

$$g' : P \rightarrow X \mid X \in Z \tag{5.10}$$

- identify a relationship between two self-certified Sybil-free pseudonyms, p_i and p_j, that are associated with two different identity domains, z_i and z_j. The objective of the attacker is to verify if those two pseudonyms were generated from the same membership certificate a;
- identify relationships between two identity domains z_i and z_j to verify if there exists an overlap between the sets B_i and B_j that are associated with these identity domains. Thus, the objective of the attacker is to obtain the function $f(z) \rightarrow B_i$, presented in Equation 5.3;
- lure a device with a membership certificate a to generate more than one self-certified Sybil-free pseudonym for an identity domain z_i. Thus, the objective of the attacker is to convince an honest device to construct the surjective and non-injective function g' presented in Equation 5.10. Thus,

the malicious user can compromise the anonymity properties of such honest user, as presented next in Section 5.2;

- partition an identity domain z. A malicious device can provide partial information regarding the members enrolled in an identity domain z. Such an attack is particularly harmful for some applications, such as anonymous communication protocols, since it can be used to decrease the cardinality of the anonymity set, i.e., the identity domain z. This attack is further explained in Section 5.3.3.

As a general assumption regarding the attacker model, we assume that the attackers are able to eavesdrop all communication data being exchanged between the participating devices. A limitation of the attacker that is connected to the aforementioned assumptions presented in Section 5.1.4 is that the attacker is allowed to have at most one membership certificate a.

5.1.6 Notation

In this section we summarize the aforementioned notation used in this section and introduce some new notation that is useful and more precise for describing the mechanisms used for the setup of the self-certified Sybil-free framework.

The characteristics of the self-certified Sybil-free framework were so far described using sets and set theory. The six sets used are summarized here:

- A is the set that represents the secure identity domain whose elements are the n membership certificates a_i that were issued by the trusted third party. Thus, $A = \{a_1, \ldots, a_n\}$, where $n \in \mathbb{N}^*$;

- B is the set of subsets B_i constructed from the set A and, thus, $B_i \subseteq A$, and $B_i \in B$, $\forall i \in \mathbb{N}^*$. The elements of a subset B_i are membership certificates a_i;

- Z is the set of identity domain identifiers z_i. An identity domain identifier is a unique information that is associated with an identity domain. The elements of the set $Z = \{z_1, \ldots, z_n\}$, where $n \in \mathbb{N}^*$ are identity domain identifiers z, such as there is a one-to-one relationship between an identity domain and an identity domain identifier z_i. Moreover, there exists a surjective and non-injective function f that associates an element of the set Z to an element of the set B, such that $f : Z \rightarrow B$;

- Q is a set of subsets Q_i, where Q_i are sets of all possible self-certified Sybil-free pseudonyms that are produced by distinct membership certificates a_i, i.e., the elements of Q_i are pseudonyms generated by distinct devices. For each element of Q, i.e., a subset Q_i, there exists a bijective function h that associates it with an identity domain identifier z_i, such that $h : Z \leftrightarrow Q$. Moreover, the cardinality of a set Q_i, $|Q_i|$, is bounded by the cardinality

of the set A, such that $|Q_i| \leq |A|$. Moreover, the set Q' is the set of all the generated, in contrast to all possible, self-certified Sybil-free pseudonyms;

- X is a subset of Z, $X \subseteq Z$. A set X is associated with a single device, i.e., to a single membership certificate. It is used to indicate the identity domains that were joined by this device. Thus, the elements of the set $X = \{x_1, \ldots, x_n\}$, where $n \in \mathbb{N}^*$, are the identifiers of the joined identity domains z.

- P is a set of self-certified Sybil-free pseudonyms produced by a device to the set X of identity domains that it is part of. Thus, there exists a bijective function g that maps the elements of the domain X and the codomain P, such that $g : X \leftrightarrow P$. Moreover, the elements of $P = \{p_1, \ldots, p_n\}$, where $n \in \mathbb{N}^*$, are unique and the union set of the all sets P is equal to the set Q'.

There are just two types of entities that are part of the presented framework: a trusted third party that issues the membership certificates a_i, and the devices that receive those membership certificates. The information regarding the possible roles that these entities may play in the framework, and if they are assumed to be trusted or not by the other participating devices, are summarized in Table 5.1.

The generation of a self-certified Sybil-free pseudonym $p_{(a,z)}$, which is produced by a device that possess the membership certificate a, $a \in A$, to the identity domain identified by z, $z \in Z$, also produces a new public key $pk_{(a,z)}$ and a certificate $cert_{a,z}$ that links this public key to the produced pseudonym.

Section 5.2 details the mechanisms used in the presented framework for issuing the self-certified Sybil-free pseudonyms and detecting Sybil identifiers in a given identity domain. The base for the instantiation of the self-certified Sybil-free pseudonyms are the periodic n-times spendable e-tokens [Camenisch et al., 2006]. Section 5.2 also highlights the similarities and differences between the self-certified Sybil-free pseudonyms and the periodic n-times spendable e-tokens. Table 5.2 provides a summary of the notation used in this chapter and the equivalent notation used in the periodic n-times spendable

Table 5.1: A summary of framework entities and their possible roles. The role of a verifier is further discussed in Section 5.2. The trusted third party can also be a domain initiator, but since this is a special case, it is not listed in the table.

Framework entities	Trusted	Possible assigned roles
trusted third party	yes	issuer of membership certificates
participating devices	no	user, verifier, or domain initiator

Table 5.2: The relationship between the notation used to detail the self-certified Sybil-free framework and the notation used in [Camenisch et al., 2006]. The identity domain identifiers are also included in this table for the sake of completeness, even though there is no equivalence for the identity domain identifiers in [Camenisch et al., 2006]. Therefore, the last row is illustrated spanning both columns to emphasize this point.

Notation used in this dissertation	Notation in [Camenisch et al., 2006]
membership certificate a, $a \in A$	e-token dispenser $\mathbb{D}$
newly generated public key $pk_{(a,z)}$	message m
pseudo-random pseudonym $p_{(a,z)}$	serial number S
pseudonym certificate $cert_{(a,z)}$	transcript τ
trusted third party	issuer $\mathcal{I}$ of e-token dispensers
identity domain identifier z, $z \in Z$	

e-tokens. Even though this chapter could have used basically the same terminology of the periodic n-times spendable e-tokens, such a terminology is not didactic to present the self-certified Sybil-free pseudonyms, and, thus, we had defined such a more intuitive notation. Nevertheless, the terminology used in the periodic n-times spendable e-tokens is also partially used in Section 5.2 to introduce the algorithms used in the e-token based signature scheme. A self-certified Sybil-free pseudonym is implemented as the tuple (*pseudo-random pseudonym*, *pseudonym certificate*, *newly generated public key*).

5.2 *k*-Spendable E-Tokens and Algorithms

The pseudonym certificates are created using a special signature scheme originally introduced for periodic n-times spendable e-tokens [Camenisch et al., 2006]. In such a proposal, sensors spend e-tokens whenever they report some data. Yet, it is only possible to compute k different e-tokens per time period. Consequently, the sensors can anonymously file at most k reports per previously specified time period. Otherwise if a sensor spends an e-token twice, all other participants can compute the sensor's identity from these two e-token show transcripts τ. While k-spendable e-tokens provide the necessary main functionality for the self-certified Sybil-free framework, the periodic n-times spendable e-tokens solution is adapted in several ways. These adaptations are:

- while the *show* protocol is interactive in [Camenisch et al., 2006], the self-certified Sybil-free framework has a non-interactive publicly verifiable *show* protocol for signature verification;
- in the self-certified Sybil-free framework a newly generated public key is

bound to the e-token *show*. This fresh public key replace the message m that is signed in [Camenisch et al., 2006];

- instead of time periods used in [Camenisch et al., 2006], the self-certified Sybil-free framework limits the number of generated e-tokens per identity domain z. An identity domain may also have a validity period, in the case of a short-term identity domain. Moreover, it may also have other parameters that identify it, as presented in Section 5.1.1, and;
- the self-certified Sybil-free framework uses a version of the protocol optimized for $k = 1$, i.e., at most one self-certified Sybil-free pseudonym is produced for a given identity domain. If more than one pseudonym is generated for a given identity domain from a given membership certificate $a \in A$, then it is possible to identify such a malicious user.

The first two properties are obtained by applying the Fiat-Shamir heuristic [Fiat and Shamir, 1987], a cryptographic trick that turns certain interactive identification protocols into signature schemes. The Fiat-Shamir heuristic is briefly explained in the Appendix A. Instead of a time period t, the self-certified Sybil-free framework uses an arbitrary identifier domain identifier z. The value z can be understood as an identification of the context in which a signer is allowed to sign only once.

5.2.1 Algorithms

The e-token based signature scheme consists of eight algorithms: *IKg* and *UKg*, which are used to produce public and private key pairs; *Obtain*, which is used to request a membership certificate; *Issue*, which is used to issue a membership certificate; *Sign*, which is used to generate self-certified pseudonyms; *Verify*, which is used verify the validity of a pseudonym; *Identify*, which is used to identify a device that generates multiple pseudonyms to a given identity domain and; *Revoke*, which is used by the trusted third party to revoke a membership certificate.

These algorithms are executed by the entities that are part of the self-certified Sybil-free framework: the trusted third party, which is the issuer $\mathcal{I}$ of e-token dispensers, and the devices that own a membership certificate or are requesting one. Moreover, the algorithm *Verify* can be executed even by devices that do not possess a membership certificate, and just monitor an identity domain to detect the presence of Sybil identifiers, i.e., the aforementioned verifiers. The algorithms are introduced using the notation of the periodic n-times spendable e-tokens. Further details regarding the algorithms used in the self-certified Sybil-free framework are presented below:

- $IKg(1^k)$ and $UKg(1^k, pk_{\mathcal{I}})$ — these two algorithms are used to create the issuer's, i.e., the trusted third party's, public and private key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$,

and the user's, i.e., a device's, public and private key pair (pk_a, sk_a), respectively. The value k is the security parameter, where k is in unary, and 1^k denote the unary representation of integer k [Goldwasser et al., 1988];

- $Obtain(pk_{\mathcal{I}}, sk_a)$ and $Issue(pk_a, sk_{\mathcal{I}})$ — these two algorithms are related and define a protocol between a user and the e-token issuer $\mathcal{I}$, i.e., the trusted third party. The algorithm *Obtain* is executed by a user, while the algorithm *Issue* is executed by the trusted third party. At the end of this protocol, the user obtains an e-token dispenser $\mathbb{D}$, i.e., a membership certificate a, that can be used to create one e-token based signature per identity domain identifier z. The trusted third party stores the public key pk_a of the user and the revocation information $r_{\mathbb{D}}$ under the user's identity;

- $Sign(m, \mathbb{D}, pk_{\mathcal{I}}, z)$ — a user produces an e-token from dispenser $\mathbb{D}$ for the identity domain z to sign a message m, i.e., a fresh public key $pk_{(a,z)}$. The outputs of this algorithm are: a token serial number S, i.e., a pseudo-random pseudonym, a transcript τ, i.e., the pseudonym certificate, and an updated e-token dispenser $\mathbb{D}'$. The triplet (m, S, τ), i.e., $(pk_{(a,z)}, p_{(a,z)}, cert_{(a,z)})$, corresponds to a self-certified Sybil-free pseudonym generated for an identity domain z;

- $Verify(m, S, \tau, pk_{\mathcal{I}}, z)$ — this algorithm is designed for checking that the pseudo-random pseudonym S and the pseudonym certificate τ, were created by a valid e-token dispenser $\mathbb{D}$ to sign a message m for the identity domain identifier z;

- $Identify(pk_{\mathcal{I}}, S, \tau, \tau', m, m')$ — given two records of self-certified Sybil-free pseudonyms (S, τ) and (S, τ'), created by a dispenser $\mathbb{D}$ when signing two different messages m and m', $m \neq m'$, for the same identity domain identifier z, the algorithm *Identify* computes the public key pk_a of the owner of the e-token dispense $\mathbb{D}$. Thus, if a device generates more than one public key and, thus, more than one public key certificate, i.e., more than one pseudonym, for a given identity domain identifier, it is possible to compute the public key pk_a that was used by this device when requesting its e-token dispenser $\mathbb{D}$, i.e, its membership certificate, to the issuer $\mathcal{I}$ and;

- $Revoke(sk_{\mathcal{I}}, pk_{\mathcal{I}}, r_{\mathbb{D}})$ — takes as input the issuer's public and private key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$ and the revocation information $r_{\mathbb{D}}$ that is related to a particular user (see the *Obtain* algorithm). The *Revoke* algorithm outputs an updated issuer public key $pk'_{\mathcal{I}}$. The dispenser $\mathbb{D}$ is revoked and can no longer be used to create signatures that verify this updated issuing key.

In the rest of the chapter, it is assumed that all participating devices use the most up-to-date issuer's public key $pk_{\mathcal{I}}$ for signing and verification. Such assumption is a common assumption regarding security systems that require

a trusted third party. Further details regarding the cryptographic construction can be found in Appendix A and in [Camenisch et al., 2006].

5.2.2 Instantiation Based on E-Token Signatures

This section describes how to implement Sybil-free self-certified pseudonyms using e-token signatures. The interaction model of the self-certified Sybil-free framework consists of two distinct phases.

The first phase is related to the request of membership certificates a from a trusted third party from devices that want to benefit from self-certified Sybil-free pseudonyms. The objective of this phase is to setup a secure identity domain A. Thus, this first step is referred to as the *setup* phase.

In the second phase, the devices that possess a membership certificate $a \in A$ can buildup identity domain identifiers $z \in Z$ and issue self-certified Sybil-free pseudonyms for a set $X \subseteq Z$ of identity domains. Thus, the second step, which is named the *operation* phase, includes two important tasks referred to as: the identity domain buildup task and the pseudonym generation task.

Setup Phase

The setup phase involves the participating devices and one issuer $\mathcal{I}$, i.e., the trusted third party. The objective of this step is to set the secure identity domain A. In the set A, every device has one, and only one, membership certificate.

First, the trusted third party $\mathcal{I}$ generates an e-token issuing public and private key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$ using the algorithm *IKg*. A device that wants to acquire a membership certificate creates a public and private key pair (pk_a, sk_a) using the algorithm *UKg*. The public part of the device's key pair, pk_a, is sent to the trusted third party $\mathcal{I}$ using a secure channel and it is authenticated under the device's identity for setting up the Sybil-free identity space. In turn, the trusted third party $\mathcal{I}$ and the device that generated the public and private key pair (pk_a, sk_a) interact using the protocol $Obtain(pk_{\mathcal{I}}, sk_a)$ and $Issue(pk_a, sk_{\mathcal{I}})$. The result of this process is that the device obtains an e-token dispenser $\mathbb{D}$, i.e., a membership certificate a.

Operation Phase

In this phase, any participating device with a membership certificate device may create an identity domain identifier z. Moreover, participating devices may join such an identity domain and, thus, issue self-certified Sybil-free pseudonyms that are associated with one, and only one, identity domain. The operation phase consists of four steps, where users may take different roles.

Table 5.1 summarized the roles that may be assigned to a participating device. The steps are as follows:

i. *setting up new identity domain identifiers* z — any participating device may set up a new identity domain, i.e., publish a unique identity domain identifier $z \in Z$. A device that publishes a new identity domain identifier is referred to as an identity domain initiator. As presented in Section 5.1.1, the identity domain initiator does not have any control over the devices that join the identity domain and its set up, and it cannot arbitrarily tear down an identity domain once it is created;

ii. *generation of self-certified Sybil-free pseudonyms* — registration at an identity domain is done using the triplet $(pk_{(a,z)}, p_{(a,z)}, cert_{(a,z)})$, i.e., a fresh generated public key, a pseudo-random pseudonym, and a pseudonym certificate. The triplet corresponds to the self-certified Sybil-free pseudonym and it is generated following the procedure presented below:

   (a) the device with a membership certificate a wants to certify a new application specific, i.e., specific to an identity domain z, and hitherto uncertified public and private key pair, $(pk_{(a,z)}, sk_{(a,z)})$ and;

   (b) the device creates a pseudo-random pseudonym $p_{(a,z)}$ for a given identity domain identifier z using the e-token to sign the new application specific public key $pk_{(a,z)}$. The $Sign(pk_{(a,z)}, a, pk_I, z)$ algorithm outputs an e-token-based signature (S, τ). The e-token's serial number S is used as the pseudo-random pseudonym $p_{(a,z)}$ and the transcript τ is used as the pseudonym certificate $cert_{(a,z)}$, as presented in Table 5.2;

iii. *verification of self-certified Sybil-free pseudonyms* — every device can verify the correctness of the pseudonym certificate $cert_{(a,z)}$ using the algorithm $Verify(pk_{(a,z)}, p_{(a,z)}, cert_{(a,z)}, pk_I, z)$. The uniqueness of the pseudonym can be checked by comparing the pseudo-random pseudonym $p_{(a,z)}$ with the pseudo-random pseudonyms of the other pseudo-random pseudonyms produced for the same identity domain z;

iv. *identification of misuse and revocation* — by executing the algorithm *Identify*, it is possible to extract the public key pk_a that is associated with the membership certificate a, of a device from two self-certified Sybil-free pseudonym registrations $(pk_{(a,z)}, p_{(a,z)}, cert_{(a,z)})$ and $(pk'_{(a,z)}, p'_{(a,z)}, cert'_{(a,z)})$ if $p_{(a,z)} = p'_{(a,z)}$ and $pk_{(a,z)} \neq pk'_{(a,z)}$. The underlying cryptographic foundation assures that all pseudo-random pseudonyms produced for a same identity domain z using the same membership certificate a are identical. The algorithm $Identify(pk_I, p_{(a,z)}, cert_{(a,z)}, cert'_{(a,z)}, pk_{(a,z)}, pk'_{(a,z)})$, outputs the public key pk_a. The membership certificate a can be revoked by the trusted third party using the algorithm *Revoke*. Nevertheless, a device that issues the

same public key $pk_{(a,z)}$ twice for a given identity domain z is not a Sybil attacker, since it is the same pseudonym that is being generated twice and not a Sybil identifier.

These four steps define the operation phase of self-certified Sybil-free framework. The operation phase is basically independent of a trusted third party, which is required only to revoke membership certificates. Nevertheless, the revocation of membership certificates can be easily postponed until the device that detected the deployment of Sybil identifiers contacts a trusted third party. Moreover, once detected, the Sybil identifiers can be removed from the set of pseudonyms Q_i associated with a domain identifier z.

5.3 Security Analysis

The objective of this section is to provide an analysis of the security and privacy properties provided by the self-certified Sybil-free framework. This section is divided in three parts. The Sybil-proof and unlinkability properties of the self-certified Sybil-free framework are assessed in the first part. In the second part, the sharing and theft of membership certificates are discussed. The presence of malicious identity domain initiators is analyzed in the third part.

5.3.1 The Sybil-Proof and Unlinkability Properties

The self-certified Sybil-free framework has a network security property and a privacy-friendly property. The security property is the possibility to detect Sybil identifiers in an anonymity set where elements of this set are devices identified by pseudonyms produced from their long-term identifiers. The privacy property is the provisioning of unlinkability between two or more self-certified Sybil-free pseudonyms produced from the same long-term identifier, i.e., no observer, including the trusted third party, can associate n self-certified pseudonyms generated for different n identity domains to the same membership certificate a.

The Sybil-Proof Property

The cryptographic properties of e-token signatures ensure that for each valid membership certificate a there can exist only one unique pseudo-random pseudonym $p_{(a,z)}$ per identity domain identifier z, as seen in Section 5.2. However, as there is no inherent trust in any other device that is part of this identity domain, including the identity domain initiator, users have to check the correctness of the pseudonym certificates $cert_{(a_j,z)}$ of all other users in the set Q_k associated with the identity domain identifier z, by locally running the algorithm *Verify*. After an honest device a_i has finished this verification and has

checked the uniqueness of a pseudo-random pseudonym $p_{(a_j,z)}$, the honest device is assured that its communication partner does not have a Sybil identifier and has a fresh generated public key $pk_{(a_j,z)}$, provided that such a public key is authenticated by proving the possession of the private key $sk_{(a_j,z)}$.

The Unlinkability Property

The self-certified Sybil-free framework has strong unlinkability properties as the cryptographic properties of the e-token signatures ensure the algorithmic unlinkability of two pseudonym certificates $cert_{a,z_i}$ and $cert_{a,z_j}$ generated for different identity domains z_i and z_k (see Section 5.2). However, the attacker may still be able to make an educated guess on whether two arbitrary pseudonym certificates from different identity domains are related or not, since information that may identify a device can be acquired from different sources in the TCP/IP stack, such as the network or application layers, as seen in Section 2.4. In a real world scenario, additional information sources could help the attacker to make such a guess, such as the location parameter of the identity domain identifier or the geographical location of the user. Traffic analysis of each setting is required to assess the concrete threats to the users' privacy.

5.3.2 Membership Certificate Sharing and Theft

In order to deploy an identity-based attack, an attacker must either forge a membership certificate a, create multiple pseudonym certificates for the same identity domain identifier z, or misuse other users' membership certificates a_i through theft or sharing. The first two options are infeasible, as they would force the attacker to break the cryptographic properties of the underlying e-token scheme, as seen in Section 5.2. Thus, the remaining viable strategies are sharing or theft of membership certificates a_i.

An attack can be launched by sharing c membership certificates among c malicious users. Still, in contrast to a Sybil attack where one attacker injects c forged identities into a network, an attacker must now inject c certified (yet misused) pseudonyms in one identity domain, which is notably more difficult than a Sybil attack, and far less effective, since this attack is bounded by the number of cooperating attackers. Sharing can be hindered by equipping the devices with Trusted Platform Modules or a similar tamper-proof hardware token, and then storing the secrets related to the membership certificate in the Trusted Platform Module. Another option is to include personal information in the membership certificate that an attacker would not be willing to share with the other c malicious users, e.g., a credit card number, that is automatically disclosed in case of sharing, for instance.

A malicious user can also try to steal membership certificates from honest users. The magnitude of such an attack is similar to the aforementioned

membership certificate sharing among malicious users. There is, however, a possibility for the trusted third party to revoke stolen membership certificates, given that the attack is detected, i.e., the malicious user and the honest user generate self-certified pseudonyms for the same identity domain, and the public key associated with the membership certificate is recovered by running the algorithm *Identify*.

Malicious users sharing membership certificates can be identified if they generate different pseudonym certificates with different public keys associated with them for a given identity domain z. Thus, the malicious users could agree on using identical public key for a given identity domain, e.g., by connecting the generation of the public key $pk_{(a,z)}$ to the identity domain z, following some deterministic function f, $f(z) = pk_{(a,z)}$. Attackers applying this strategy cannot be identified, since the pseudonym certificates are the same. However, the c malicious users can only generate c pseudonym certificates.

The prevention of membership certificate sharing would require some form of interactivity during the generation of the fresh public key that is associated with the pseudonym certificate, e.g., the domain initiator or other online parties could contribute with randomness to the generation of the public key pair associated to the identity domain. A drawback of sharing membership certificates is that malicious devices have to trust each other, since they will have to share the secret keys $sk_{(a,z_i)}$ associated with the public keys $pk_{(a,z_i)}$ and their e-token dispensers $\mathbb{D}$.

5.3.3 Malicious Identity Domain Initiators

An identity domain initiator crafts an identity domain identifier $z_i, \in Z$, by setting the parameters that are included in context information of the identity domain z_i. An example of a context information can be found in Figure 5.2.

This section evaluates the impacts of actions performed by malicious identity domain initiators. The purpose of a malicious identity domain initiator is to compromise the security and privacy of the self-certified Sybil-free framework by, e.g., luring honest users to generate multiple self-certified Sybil-free pseudonyms for a given identity domain or partitioning the identity domain into smaller domains to reduce the size of the anonymity set. Thus, a malicious identity domain initiator may:

- setup an identity domain identifier z' that is equal to a pre-existing identity domain identifier z, such that $z = z'$. The objective of the attacker is to lure an honest device to produce a pseudonym certificate $cert_{a,z'}$ for this identity domain identifier z'. Thus, if the honest device has already produced another pseudonym certificate $cert_{a,z}$ for the pre-existing identity domain z, such that $cert_{a,z} \neq cert_{a,z'}$, the malicious domain initiator can execute the algorithm *Identify* to retrieve the public key pk_a associated with

this honest device. This attack is thwarted if the honest devices keep a list of identity domain identifiers z that it had already joined. Thus, a device is able to recognize an identity domain identifier z that it had already joined and simply retrieve the pseudonym certificate associated with this identity domain;

- protect other malicious devices that share membership certificates from being identified. This might be possible if we assume an identity domain initiator that acts as a directory service, i.e., an identity domain initiator that is responsible to publish a list of pseudonym certificates that are part of the identity domain. In this case, the identity domain initiator can remove pseudonym certificates that are identical from such a list, i.e., pseudonym certificates generated from a same membership certificate, but associated with two or more fresh generated public keys. Therefore, the malicious identity domain initiator can prevent other devices to identify malicious devices sharing membership certificates.

Moreover, assuming a malicious device strategically located in between the subset of honest devices B_i and an honest identity domain initiator that produced an identity domain identifier z. This malicious device is, thus, acting as the only path connecting this subset and the identity domain initiator. Therefore, it can prevent the dissemination of the identity domain identifier to the subset B_i and reduce the anonymity set associated with the identity domain identifier z.

Furthermore, if this malicious device is used as the only possible path connecting two subsets of honest devices B_i and B_j, the malicious device can protect two other malicious devices sharing membership certificates if they are not located in different subsets by not propagating the pseudonym certificates of the malicious users from one subset to the other. Nevertheless, such attacks are highly dependent on the network topology and may not be feasible to deploy in practice, since they depend of series of highly improbable conditions in an ad hoc network.

While the listed attacks should be considered relevant, none of the aforementioned attacks allow a malicious device to break the Sybil-proof and unlinkability properties of the self-certified Sybil-free framework presented in Section 5.3.1.

5.4 Sybil-Free Applications and Related Work

This section has two objectives, and, is thus divided in two parts. The first part discusses the applicability of the proposed self-certified Sybil-free framework and outlines applications that can benefit from privacy-friendly Sybil-free

identity domains. The second part of this section presents other solutions for the generation of privacy-friendly identifiers.

5.4.1 Privacy-Friendly Sybil-Free Applications

The objective of this section is to present applications and application scenarios that would largely benefit from the self-certified Sybil-free framework, i.e., applications that demand identity domains free of Sybil identifiers, but also have privacy requirements, or at least, can profit from a privacy-friendly setting.

The number of applications where a group of users interact electronically is endless: instant messaging, chat rooms, forums, and e-commerce platforms are only a few examples of widely used applications. Often, such applications allow users to slip into different roles, and behave accordingly. Nevertheless, with the growing size and sophistication of such communities and applications, the amount of required administration tasks increases. Misbehaving users need to be excluded, users' contributions need to be evaluated based on their reputation, and tasks need to be distributed. In short, such applications and communities develop their own social dynamics, and there is a need to make decision processes work in a more automated way. Such decisions could, for instance, be based on majority voting, seniority, or reputation.

Truly anonymous or pseudonymous applications are currently debated, partly because they can enable misbehaving users to create social problems within their communities. Although these users can be banned from such applications, it is often easy for the wrongdoers to simply re-register using a new name. To change IP addresses using proxies or similar techniques is enough to thwart most existing countermeasures. Reputation systems also break under such an attack as users can register multiple times to collaboratively increase the reputation of all of their pseudonyms. Furthermore, the allocation of resources and the distribution of work can be potentially manipulated by misbehaving users in applications that use reputation information. Malicious users can also choose names similar to other users to abuse their reputation. Users that control multiple identities can also more easily spread rumors and influence voting results to their own advantage.

The separation between real world identities and different virtual worlds that allows the support of pseudonymous and anonymous users is a valued feature since every activity performed in a networked environment may be logged and stored for further analysis. This separation decreases the privacy risks associated with interacting in a computer network environment. Many papers have been dedicated to various types of pseudonymity, e.g., [Bhargav-Spantzel et al., 2006; Borcea-Pfitzmann et al., 2005; Franz and Borcea-Pfitzmann, 2006] or to the graceful degradation of anonymity towards full identification [Andersson et al., 2005a] using existing approaches. Numerous applications would thus benefit from the presented self-certified Sybil-free framework, such as:

- *peer-to-peer systems* — some of these systems need to manage users' reputation and electronic voting schemes, and, thus, would benefit from the self-certified Sybil-free framework. Other peer-to-peer systems implement dummy e-currencies and might require a distributed scheme for detecting double-spending;
- *online communities* — some platforms, such as the ones used for electronic auctions, would benefit from protection against self-ranking, i.e., artificially increase of a user's reputation. Moreover, if a user deletes its account and joins the platform again to reset its reputation score, such actions can be linked. Other online communities, such as social networks, can be protected in a way that only one profile can be posted per membership certificate;
- *anonymous communication systems* — these systems require a portion of the participating devices to be honest. Such systems usually assume that devices on the path between the sender and the recipient are distinct, belong to different users and do not cooperate. If such assumptions are not followed, the anonymity properties can be compromised.

The Sybil-free self-certified pseudonyms can be used in admission control schemes [Kim et al., 2003; Saxena et al., 2003, 2005] to aid applications or to manage anonymous or pseudonymous users in a secure and privacy-respecting manner. Privacy-friendly admission control allows the creation of several identity domains z simultaneously, whose participating devices cannot be linked, i.e., it is not possible for any device to affirm that another participating device has joined none, one or more than one identity domain with a probability greater than the one achieved by guessing. Thus, a user can be part of multiple identity domains simultaneously and the identifiers used in different identity domains are unlinkable.

5.4.2 Other Privacy-Friendly Identifiers

In this section other solutions for privacy-friendly identifiers are outlined and discussed. Nevertheless, the [Camenisch et al., 2006] e-tokens on which the self-certified Sybil-free framework is based on is not discussed in this section, since the cryptographic foundation of both schemes is the same and the similarities and differences between these two solutions were already presented in Section 5.2.

Group Signatures, Group Key Distribution, and Key Agreement

Anonymous authentication that provides unlinkability between multiple shows of the same identifier can be implemented using group signatures [Boneh et al.,

2004; Chaum and van Heyst, 1991]. Group signature schemes support escrowed anonymity, i.e., a trusted third party that can open a group signature and reveal the identity of the signer. A privacy-preserving protocol using such scheme was proposed, for instance, in [Lin et al., 2007] for signing messages in the context of vehicular communications. However, group signature schemes alone do not provide any protection against a signer generating any two group signatures, i.e., the deployment of Sybil identifiers [Defrawy and Tsudik, 2007; Tsudik and Xu, 2006].

Nevertheless, the combination of a group signature scheme, a centralized group key distribution scheme, and a distributed key agreement scheme into a secure secret handshake scheme can provide unlinkability, anonymous authentication, and detection of Sybil identifiers. Such a solution demands a group controller and a group manager service, which might be executed in a single device, for distributing identities to the participating devices, admission control, and updating system state information, i.e., rekeying.

The framework presented in [Tsudik and Xu, 2006] is cryptographically sound, but it has some practical drawbacks if deployed in an ad hoc network scenario. The framework requires the continuous presence of the group controller, for admitting new users into the group and, in addition, it requires a rekeying process every time a device joins a group. The proposed self-certified Sybil-free pseudonyms do not have such requirements since domain initiators cannot prevent devices from joining an identity domain and pseudonyms are self-certified, i.e., they are locally produced.

Moreover, a distributed group key agreement followed by the broadcasting of an authentication tag to all group participants precedes the detection of Sybil identifiers. The requirement of a group key agreement incurs that the detection of Sybil identifiers can only be achieved in the group of devices that are online and joining the group key agreement. Such a requirement does not exist in the proposed self-certified Sybil-free framework, since self-certified pseudonyms can be generated asynchronously and the detection of Sybil identifiers can be performed even if the Sybil identifiers were generated at distinct time instants. In addition, the broadcast distribution of authenticated tags causes a heavier traffic load than the self-certification approach used in the self-certified Sybil-free framework. Small network traffic overhead is especially important in scenarios with resource constraints, such as an ad hoc network scenario.

Furthermore, the group manager can always trace the users involved in a handshake session, while in the self-certified Sybil-free framework, the identification of a participating device can only be performed if a device deploys Sybil identifiers. Nevertheless, the cryptographic framework presented in [Tsudik and Xu, 2006] can be more appropriate than the self-certified Sybil-free framework for some applications in which it is important to identify users and the group manager is trusted not to abuse its rights.

Identity-based Encryption, Pairing, and Blind Signatures

Identity-based encryption schemes can be used to construct pseudonyms. The pseudonym-based encryption scheme proposed in [Huang, 2007] is based on pairings and constructed on top of an identity-based encryption scheme [Boneh and Franklin, 2003] and short signatures from the Weil pairing [Boneh et al., 2001]. A device's public key is assumed to be its initial identifier of a device in the pseudonym-based encryption scheme.

The pseudonym-based encryption scheme was designed for achieving anonymous communication in ad hoc networks. Thus, the pseudonym-based encryption scheme allows the local generation of pseudonyms, i.e., without the presence of a trusted third party. Nevertheless, a group manager is required for setting up groups of anonymous devices and admitting anonymous devices to these groups. The group manager generates certificates that are bound to pseudonyms that join a group. These certificates are issued using a blind signature scheme [Boldyreva, 2003].

There is no identity escrow in this scheme, therefore, it is not possible for any device to obtain the initial public key that was used to generate a pseudonym [Huang, 2007]. However, the group manager can revoke the certificate of a pseudonym or of a group of pseudonyms.

The main disadvantage of the pseudonym-based encryption scheme is that it is vulnerable to a Sybil attack. Any device with an initial identifier, i.e., a public key is used as the initial identifier in this scheme, can generate an arbitrary number of pseudonyms. Since the group managers do not have any mechanism to distinguish if two or more pseudonyms were generated from the same public key, Sybil identifiers can also obtain certificates from the group managers. Thus, the pseudonym-based encryption scheme should not be deployed in application scenarios that require a Sybil-free group of identifiers.

Extensions to X.509 Attribute Certificates

X.509 attribute certificates [ITUT X.509] can be made privacy-friendly by assigning a pseudonym in the *holder* field instead of binding it directly to an identity certificate [Benjumea et al., 2007]. Such schemes assume the existence of an Attribute Authority, which is responsible for issuing the attribute certificates, and a Source of Authority [Benjumea et al., 2004], which is a root trusted authority of delegation chains, i.e., a root entity in a tree structure that binds the Attribute Authorities under it. In addition, other entities can be included in the system, such as Judges and Judge Agents [Benjumea et al., 2006], which are authorities responsible for deciding upon revocation of anonymity and tracing user activities.

Anonymous X.509 attribute certificates can be constructed using different signature schemes, such as fair blind signatures, traceable signatures, and

ring signatures [Benjumea et al., 2007]. Attribute certificates created with fair blind signatures [Stadler et al., 1995] were presented in [Benjumea et al., 2004]. However, such schemes do not provide unlinkability between multiple shows of a same attribute certificate. A traceable signature scheme [Kiayias et al., 2004], which is basically group signatures schemes with additional tracing capabilities [Benjumea et al., 2007], was used as a cryptographic primitive to set up privacy-friendly X.509 attribute certificates that can provide unlinkability between different shows of a same attribute certificate [Benjumea et al., 2006].

The main benefit of these solutions is that the structure of the standard attribute certificate remains unaltered and, thus, can benefit from the existing infrastructure provided by the X.509 framework. Nevertheless, some new entities are introduced in the framework for the supporting of privacy-friendly applications, such as the Source of Authority and the Judge. However, the main disadvantage of such systems is that Sybil identifiers can be easily deployed. Thus, for certain applications that demand interaction between devices, the current proposed extensions of X.509 attribute certificates are not enough to prevent or detect the deployment of Sybil identifiers. Moreover, the proposed extensions of the X.509 framework require some entities to be constantly available, such as Source of Authority and a trusted third party. Thus, such a solution may not be adequate for ad hoc networks.

5.5 Summary

In this chapter, we have presented a framework for producing self-certified Sybil-free pseudonyms starting from a secure identity domain, i.e., a set of devices where each device has one, and no more than one, long term identifier, a so-called membership certificate a. Self-certified Sybil-free pseudonyms are associated with identity domains z, which are subsets of the initial secure identity domain A. Such pseudonyms are locally produced from this membership certificate and, in addition, the generated pseudonyms cannot be linked back to the membership certificate. Moreover, two pseudonyms generated for two different identity domains using the same membership certificate cannot be linked, either. Nevertheless, if two or more distinct pseudonyms are generated for the same identity domain using a single membership certificate, it is possible for any device, not only to detect the presence of a Sybil identifier, but also to recover the public key associated with the membership certificate.

In this chapter, we have also provided an analysis of the security properties provided by the self-certified Sybil-free pseudonyms regarding the detection of Sybil identifiers, the unlinkability between different pseudonyms, the sharing and theft of membership certificates, and the presence of malicious identity domain initiators. Finally, we presented a list of applications that would cer-

tainly benefit from privacy-friendly Sybil-free identifiers, and other initiatives for constructing identifiers with privacy-friendly properties.

In the following chapter, we present Chameleon, an anonymous overlay communication mechanism for ad hoc networks. The goals of Chameleon are to provide sender anonymity against recipients and relationship anonymity against local observers with a reasonable performance cost. Chameleon was designed taking into account the requirements for anonymous communication mechanisms presented in Chapter 4. Moreover, Chameleon can benefit from the self-certified Sybil-free pseudonyms, as any other anonymous communication mechanism, to set up an anonymity set that is free from Sybil identifiers and the members of such a set cannot be linked to members of other anonymity sets.

Chapter 6

The Chameleon Protocol

"Escravos de Jó jogavam caxangá.
Tira, põe, deixa ficar...
Guerreiros com guerreiros fazem zigue zigue zá.
Guerreiros com guerreiros fazem zigue zigue zá."

— *Brazilian nursery rhyme*

This chapter presents Chameleon, an overlay anonymous communication mechanism designed according to the requirements for anonymous communication mechanisms presented in Section 4.2.1. Chameleon is tailored for ad hoc networks and provides sender anonymity against recipients and relationship anonymity against local observers with reasonable performance costs. In addition, Chameleon provides conditional anonymity against malicious Chameleon users, as well as protection against single attackers trying to compromise large portions of a network by assuming multiple identities, i.e., a Sybil attack. Chameleon builds on a flexible design that provides isolation and independence from both the application and transport layers, allowing the usage of standardized mobile ad hoc routing protocols. To the best of our knowledge, Chameleon was the first low-latency anonymous overlay network designed for an ad hoc network setting.

Chameleon was designed with the characteristics of ad hoc environments in mind. The key characteristics of those environments, such as user mobility and vanishing devices, were thus taken into account in Chameleon. The core functionalities are inspired by the Crowds system for anonymizing HTTP traffic [Reiter and Rubin, 1997]. The decision to base the design of Chameleon on the Crowds system was made according to the evaluation of peer-to-peer anonymous overlay networks in the context of ad hoc networks presented in [Andersson et al., 2005b]. Although none of the peer-to-peer anonymous communi-

cation protocols assessed in such an evaluation were fully compliant with the characteristics of ad hoc networks, the Crowds system was deemed as an appropriate choice for a foundation upon which Chameleon could be developed, as briefly explained in Section 6.2. Nevertheless, a number of modifications were made to Crowds, such as including end-to-end encryption between the sender and the recipient and the use of credentials to hinder attackers from having multiple identifies, such as the self-certified Sybil-free pseudonyms presented in Chapter 5. Moreover, Chameleon is a general overlay network accepting any messages from the application layer, independently from the application that uses it.

The remainder of this chapter is organized in four sections. In Section 6.1, general strategies for setting up anonymous communication networks are presented along with a short description of the Crowds anonymous communication mechanism. The Chameleon protocol, its architecture and assumptions are outlined in Section 6.2. In Section 6.3, the framework around the Chameleon protocol is presented. Such a framework includes the different classes of devices, the distinct types of messages, and the relay tables used to support the Chameleon protocol. Moreover, Section 6.3 presents a detailed description of Chameleon using state-transition diagrams. Finally, the last section presents the theoretical analysis of the Chameleon protocol.

6.1 Anonymous Communication Networks

This section is divided in two parts. The first part presents the alternatives for setting up anonymous paths that are implemented by different anonymous communication mechanisms. The second part provides a short description of the Crowds anonymous communication mechanism.

6.1.1 Anonymous Communication Network Strategies

An anonymous path routes encrypted messages through chains of devices. To protect against traffic analysis, the appearance of the messages is changed at each device in the path through encryption. There are basically two main strategies for constructing anonymous paths in anonymous overlay networks:

- in the first strategy, the sender selects all the intermediary devices at the application layer, i.e., the sender decides the whole anonymous path between the sender and the recipient [Syverson et al., 1997]. There are two main methods to implement such a strategy:
  - the sender selects the entire anonymous path by wrapping a message in several layers of encryption, one for each intermediary device along the path. These layers are thereafter peeled off by decryption,

one by one, at each subsequent device on the anonymous path until the message arrives in the recipient device. Such an approach is applied in layered encryption systems;

- the sender to selects the entire anonymous path using an incremental (telescopic) path establishment. In telescopic path building the initiator negotiates cryptographic session keys with each successive hop in the path and extends it hop by hop. Such an approach is applied in anonymous communication systems such as Tor [Dingledine et al., 2004];

- in the second strategy, intermediate devices select their respective successor device in an anonymous path. Such a strategy works as follows. The sender selects its successor device from all possible devices that are part of the anonymity set and forwards the message to it. Next, the chosen intermediate device decides, following some criteria, if the message received from the previous device should be delivered to the destination or if the message should be forwarded to another intermediate device, which is also chosen from the anonymity set. If the intermediate device decides to forward the message to another intermediate device the process is repeated until an intermediate device in the anonymous path decides to deliver the message to the destination device. Such an approach was first proposed and implemented in the peer-to-peer anonymous communication system Crowds.

To deal with high mobility and to enable efficient path repairing in case of disappearing devices, Chameleon employs the latter strategy for establishing anonymous paths.

6.1.2 The Crowds System

Crowds is a peer-to-peer anonymous communication mechanism originally designed for anonymous web browsing on the Internet [Reiter and Rubin, 1997]. Moreover, Crowds is an overlay protocol and, thus, operate over the transport layer and below the application layer.

Crowds implements the strategy of letting intermediate devices to select their successor in an anonymous path in case it decides not to deliver the message directly to the destination device. In Crowds, a sender device S forwards application data, such as an HTTP request message[1], to an intermediate device J_1, which is randomly selected from the anonymity set and is also the first intermediate device in the anonymous path initiated by S. An intermediate device J_i, where $i \in \mathbb{N}^*$, makes a random choice to forward the received application

[1]The Hypertext Transfer Protocol, or HTTP, is a stateless application-level protocol [Fielding et al., 1999].

data to another intermediate device, which results in the extension of the anonymous path by one more hop, or in the delivery of the message directly to the destination device D. The choice of extending or ending the anonymous path is decided according to the outcome of a toss of a biased coin. The probability of extending the path is called the probability of forwarding p_f, where p_f is bounded by the open interval $]0.5, 1[$. The probability of delivering message to the destination device D is, naturally, $(1 - p_f)$. Further messages sent towards the destination D by the sender S are forwarded through the same, i.e., already constructed, anonymous path.

The reply message from D to S is sent backwards along the anonymous path, with each intermediate device sending the reply message to its predecessor device in the path. All the communication data between the sender S and J_1, as well as all communication between any two intermediate devices J_i and J_k is encrypted using pre-distributed symmetric keys that are shared between the devices that are part of the anonymity set.

The anonymity set in Crowds is controlled by a centralized directory server, the blender. The role of a blender is threefold: it is responsible for admission control in the anonymity set, it distributes the list of devices in the anonymity set, and it distributes the symmetric keys that are used in the encryption of the communication data between two devices in an anonymous path.

In Crowds, an intermediate device J_i on an anonymous path cannot distinguish whether its predecessor on the path J_{i-1} is the source of the application data, i.e., that J_{i-1} is the sender S, or is just forwarding the data on behalf of another device. Thus, no intermediate device on the anonymous path knows which device is the sender device S. The sender anonymity for S against the destination D is *beyond suspicion*[2] since from the perspective of the destination D all devices are equally likely to be the sender of a message and D obtains no further information regarding who initiated the given anonymous path. Crowds also offers receiver anonymity against a local eavesdropper that can observe all communication of a given device as long as the anonymity set is sufficiently large. A more complete anonymity analysis of Crowds is provided in [Reiter and Rubin, 1997].

6.2 Chameleon Anonymous Overlay Network

The objective of the Chameleon protocol is to hide one user's action within the actions of many other users. By sending messages through anonymous paths, a user can participate in a communication session while at the same time hiding his identity among the identities of the other users in the mobile ad hoc network.

[2]The anonymity metric of Crowds is further explained in Section 7.1, on page 118.

This section is divided in three parts. The first part outlines the anonymous path establishment process and the general characteristics of such paths. The second part highlights the main differences between Chameleon and the Crowds protocol. The assumptions considered during the design of Chameleon are presented in the last part of this section.

6.2.1 Anonymous Paths in Chameleon

During path establishment, the decision of an intermediate device to extend or terminate the anonymous path is determined by the probability of forwarding p_f, where p_f is bounded by the open interval $]0.5, 1[$. With the probability $(1 - p_f)$, the path is ended and a connection is established with the destination. Otherwise the path is extended to another randomly chosen device, at which the same process is repeated. The path length L is thus probabilistic and denotes the sum of device appearances on the path (excluding the destination device). The shortest anonymous path has only the sending device and one intermediate device. Thus, $L \geq 2$. The expected path length, L_{exp}, is given in Equation 6.1 [Reiter and Rubin, 1997], and the curve produced from this equation is graphically illustrated in Figure 6.1.

$$L_{exp} = \frac{p_f}{(1 - p_f)} + 2 \quad | \quad 0.5 < p_f < 1, \ \forall\, p_f \in \mathbb{R} \tag{6.1}$$

Anonymous paths are bidirectional, meaning that messages can travel forward, i.e., towards the destination, or backward, i.e., towards the sender. As in Crowds, the destination's IP address is known only to the devices belonging to the path, and path rebuilding is performed in the forward direction only[3]. To provide better protection against local observers, link encryption is employed between the devices in the anonymous path. Unlike Crowds, conditionally on the destination type, end-to-end encryption may also be applied between the sender and destination.

6.2.2 Chameleon and the Crowds Protocol

In this section, the main differences between the Chameleon and Crowds are outlined. The difference are:

- Chameleon does not rely on a blender for admission control, distribution of cryptographic keys, and distribution of the list of participants of the anonymity set, as the Crowds protocol does. Instead, the distribution of

[3]To allow path rebuilding also in the backward direction, intermediate devices would require greater knowledge about the other devices that anonymous path and, eventually, would be able to identity of the sender.

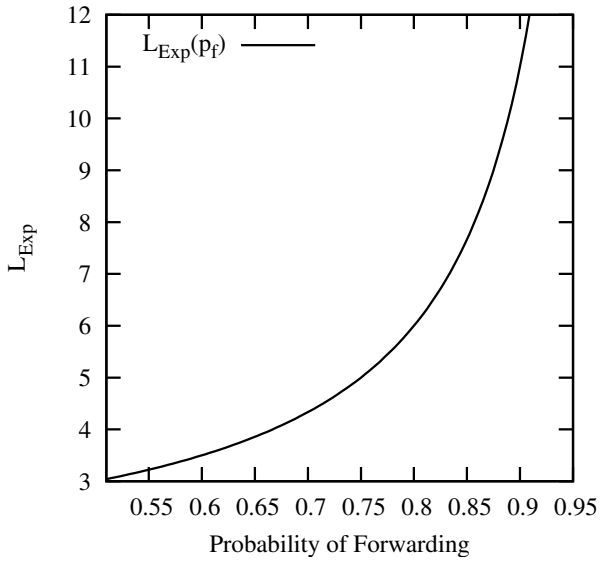

Figure 6.1: This curve illustrates the expected path length L_{exp} as function of the value associated with the probability of forwarding p_f. There is a direct relationship, i.e., positive relationship, between the expected path length and the probability of forwarding.

list of elements in the anonymity set and the public-keys associated to such elements can be performed by any participating device, and, thus, multiple directory servers may be running simultaneously. Moreover, self-certified Sybil-free pseudonyms are used to differentiate the elements of an anonymity set. Any device that can produce a self-certified Sybil-free pseudonym can join an anonymity set in Chameleon.

- Chameleon was proposed and analyzed for a wireless ad hoc network scenario, while Crowds was designed to be implemented and used in a network with deployed infrastructure, such as the Internet. Thus, the attacker model and the anonymity analysis of both protocols differ according to the network scenario. The attacker model of Chameleon is presented in Section 6.4, whereas the anonymity analysis of Chameleon is presented in Chapter 7.

- Chameleon provides end-to-end encryption, if such a feature is supported by the destination device. The Crowds protocol does not offer native end-to-end encryption. However, Crowds operates as a web proxy and can thus forward not only HTTP requests, but also establishes end-to-end

secure connections to a destination device using a standard secure transport protocol, such as DTLS over UDP [Rescorla and Modadugu, 2006] or TLS [Dierks and Rescorla, 2008], since the messages exchanged to set up a secure end-to-end tunnel are treated as any other application data by Crowds.

6.2.3 Assumptions

Chameleon relies on a set of assumptions regarding the identifiers used in the set up of the anonymity set, the establishment of secure sessions, and the characteristics of the ad hoc network. The assumptions are as follows:

- it is expected that identifiers, such as the membership certificates described in Chapter 5, are obtained a priori from a third trusted party, which is, most likely, located in a fixed network. Temporary availability of a trusted third party is also present in other papers dealing with the problem of anonymity in ad hoc networks, as discussed in Chapter 2;
- Chameleon assumes that it is possible to establish secure sessions at the transport layer, with mutual authentication using anonymous credentials followed by the establishment of symmetric keys, and;
- since network and hardware addresses are not necessarily unique identifiers, and, thus, might not constitute a long-term one-to-one relationship with a given device, it is assumed that the ad hoc network is also a service-based network, such as a Jini [Jini] or UPnP [UPnP] network. Therefore, all network services, including the anonymity services, are announced through a directory service, such as Jini's Lookup Server [Jini].

6.3 The Chameleon Framework

In this section, the components of the Chameleon framework are detailed. Beyond the aforementioned Chameleon protocol, the Chameleon framework also defines different classes of devices, distinct types of messages, and the message relay table. The components of the Chameleon framework are presented in the following order. First, the device classes are presented, followed by the message types, the message relay table, and, finally, the Chameleon protocol.

6.3.1 Device Classes and Anonymous Paths

The Chameleon framework is composed of devices that are organized in non-exclusive classes, i.e., a device might be part of one or more classes simultaneously, according to their role and running services. There are four non-

exclusive classes of devices in Chameleon. The following notation is used for describing them:

- Ψ denotes the set of all devices in the ad hoc network, where the elements of the set $\Psi = \{\psi_1, \psi_2, \ldots, \psi_n\}$ are distinct devices, and the cardinality of the set Ψ, $|\Psi| \in \mathbb{N}$;
- Γ denotes the set of all Chameleon users, $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, where $\Gamma \subseteq \Psi$. Regarding the cardinality of the set Γ, $|\Gamma|$, it is assumed that it is equal or greater than 3, i.e., $|\Gamma| \geq 3$. This is the minimum amount of Chameleon users required in the framework to provide a certain level of anonymity, as presented in the Section 7. Thus, in relation to the set Γ of all Chameleon users:

$$\Gamma \subseteq \Psi,\ 3 \leq |\Gamma| \leq |\Psi| \ \wedge\ |\Gamma| \in \mathbb{N} \tag{6.2}$$

- D denotes the set of all destination devices, $D = \{d_1, d_2, \ldots, d_n\}$. The set D is the union of three disjoint subsets, $D_{\overline{sec}} \cup D_{sec} \cup D_\Gamma = D$ and $D_{\overline{sec}} \cap D_{sec} \cap D_\Gamma = \varnothing$, and each of those subsets are associated with a different class of destination devices. These classes are:
  - $D_{\overline{sec}}$ is a set of destination devices that accept only unencrypted request messages;
  - D_{sec} is a set of destination devices that accept secure requests using a standard secure transport protocol between an element of the set D_{sec} and the last Chameleon user in an anonymous path, and;
  - D_Γ is a set of destination devices that understand Chameleon protocol messages, which allows the use of end-to-end encryption between an element of the set D_Γ and the sender, which is a Chameleon user. Thus, $D_\Gamma \subseteq \Gamma$;
- Φ denotes the set of all devices running a directory service, where $\Phi \subseteq \Gamma$ and $\Phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$. A device ϕ_i running a directory service announces a set of network addresses, IP_Γ of the available elements in Γ, i.e., other Chameleon users that are part of the anonymity set.

To reveal as little information as possible to any element of the set Φ, each device in Γ requests the set of network addresses IP_Γ at regular time intervals. Restricting $\Phi \subseteq \Gamma$ decreases the likelihood of corrupted directory services announcing false information, since they can be detected and identified as malicious devices and filtered out by other Chameleon users.

The announcement of IP_Γ follows one of the principles of zero configuration networking working group [IETF zeroconf], which assumes the existence of a service discovery system in network environments such as ad hoc networks. The devices in Φ act as a decentralized version of the blender service in Crowds.

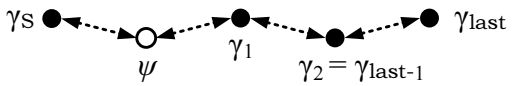

Figure 6.2: An illustration of an anonymous path that extends from a sending device $\gamma_s \in \Gamma$, which is the source of the application data θ, to the device $\gamma_{last} \in \Gamma$. There are two intermediary devices γ_1 and $\gamma_2 \in \Gamma$ in the anonymous path connecting γ_s to γ_{last}. In this example, the data sent by the device γ_s towards γ_1 is routed, in the network layer, through a device in $\psi \in \Psi$, which is not in Γ.

In the Chameleon framework, an anonymous path is defined as a path connecting the sender, $\gamma_s \in \Gamma$, with the last device before the destination, $\gamma_{last} \in \Gamma$, where γ_s and γ_{last} are interconnected by zero or more γ_i devices, where $\gamma_i \in \Gamma$. When describing the protocol in the following sections, γ_i denotes the device where the current message is being processed. In the Chameleon framework, multiple anonymous paths may naturally exist during the same time slot, and a device in Chameleon can be part of multiple anonymous paths simultaneously. Moreover, a device in Chameleon might be a sender, γ_s, an intermediary node, or the last device in an anonymous path, γ_{last}, for one or more anonymous paths that such a device is belonging to. Figure 6.2 illustrates an anonymous path that extends from a sending device $\gamma_s \in \Gamma$, which is the source of the application data θ, to the device $\gamma_{last} \in \Gamma$.

6.3.2 Chameleon Message Types

The Chameleon framework has three distinct types of messages. Two of them are used for the communication between two consecutive devices in an anonymous path, while the third one is used in the communication between the Chameleon overlay and the application layer. The following notation is used regarding the types of messages in Chameleon:

- θ denotes the application data that is passed from the application layer to the Chameleon overlay;
- m_{γ_i,γ_j} denotes a message that is transmitted between two consecutive devices, γ_i and γ_j, that are part of an anonymous path and are running the Chameleon protocol. This message m_{γ_i,γ_j} is encrypted between γ_i and γ_j using a symmetric encryption key $E_{k_{\gamma_i,\gamma_j}}$. This symmetric encryption key is established using a secure transport layer protocol, such as DTLS over UDP [Rescorla and Modadugu, 2006], DTLS over DCCP [Phelan, 2008], and TLS [Dierks and Rescorla, 2008]. If the destination device $d \in D_{sec}$ or $d \in D_{\overline{sec}}$, the payload of the message m_{γ_i,γ_j} includes:
  - IP_d, which is the logical address of the destination d;

- a path identifier $p\#_{\gamma_i,\gamma_j}$. The path identifier is a randomly generated value that uniquely identifies a packet stream between two devices γ_i and γ_j. This identifier is used to discriminate different packet streams being forwarded between these two devices, and;
- the data payload θ.

The payload of the message m_{γ_i,γ_j} for a destination device $d \in D_{sec}$ or $d \in D_{\overline{sec}}$ is illustrated in Equation 6.3, where the symbol "·" denotes the operation of concatenation.

$$m_{\gamma_i,\gamma_j} = E_{k_{\gamma_i,\gamma_j}}[p_{\#\gamma_i,\gamma_j} \cdot IP_d \cdot \theta] \tag{6.3}$$

If the destination device $d \in D_{\Gamma}$, the payload of the message m_{γ_i,γ_j} has two additional fields that are used to achieve end-to-end encryption and data integrity. These additional fields are:

- a symmetric key $k_{\gamma_s,d}$, which is encrypted with the destination's public key, Pu_d. The symmetric key $k_{\gamma_s,d}$ is used to create an end-to-end secure channel between γ_s and the destination device d.
- the output of a keyed-hash function, such as an HMAC [HMAC]. The input of this keyed-hash function is the application data θ and the cryptographic key used in this operation is $k_{\gamma_s,d}$.

The payload of the message m_{γ_i,γ_j} for a destination device $d \in D_{\Gamma}$ is illustrated in Equation 6.4.

$$m_{\gamma_i,\gamma_j} = E_{k_{\gamma_i,\gamma_j}}[p_{\#\gamma_i,\gamma_j} \cdot IP_d \cdot E_{k_{\gamma_s d}}[\theta] \cdot E_{Pu_d}[k_{\gamma_s d}] \cdot hash_{k_{\gamma_s,d}}(\theta)] \tag{6.4}$$

- $ack_{\gamma_{i+1},\gamma_i}$ denotes an acknowledgment message that is transmitted between two consecutive devices, from γ_{i+1} to γ_i. This message is produced in the last device of the anonymous path, γ_{last}, and sent towards γ_s through the anonymous path to inform γ_s that the application data θ has reached its destination $d \in D$. The acknowledgement message $ack_{\gamma_{i+1},\gamma_i}$ is shown in Equation 6.5.

$$ack_{\gamma_{i+1},\gamma_i} = E_{k_{\gamma_{i+1},\gamma_i}}[p_{\#\gamma_{i+1},\gamma_i}] \tag{6.5}$$

6.3.3 Chameleon Relay Table

Each device in Chameleon maintains a relay table. This table is used to associate incoming and outgoing packet streams with their path identifiers. The Chameleon relay table has several entries, where each entry is associated with a unique packet stream. An entry of the Chameleon relay table has the following mandatory fields:

IP_d	$IP_{\gamma_{i-1}}$	$p_{\#\gamma_{i-1},\gamma_i}$	$IP_{\gamma_{i+1}}$	$p_{\#\gamma_i,\gamma_{i+1}}$	TTL

Figure 6.3: An entry in the Chameleon relay table. The 1st field is the destination's logical address. The 2nd field is the logical address of the preceding device. The 3rd field is the backward path identifier. The 4th field is the logical address of the succeeding device. The 5th field is the forward path identifier. Finally, the 6th field is the time-to-live (TTL) counter.

- the destination's logical address, IP_d;
- the logical address of the preceding device in the anonymous path, $IP_{\gamma_{i-1}}$;
- the backward path identifier, $p_{\#\gamma_{i-1},\gamma_i}$, which is associated with an incoming packet stream;
- the logical address of the succeeding device in the anonymous path, $IP_{\gamma_{i+1}}$;
- the forward path identifier, $p_{\#\gamma_i,\gamma_{i+1}}$, which is associated with an outgoing packet stream, and;
- a time-to-live (TTL) counter. The TTL is a decremental counter that indicates the remaining lifetime of an entry in the table. It is used to remove inactive path entries from the Chameleon relay table. This counter is reset if a new packet is transmitted in the anonymous path associated with this entry.

Figure 6.3 graphically illustrates an entry of the Chameleon relay table and its fields. The path identifiers are managed in the same way as the *path_id* in Crowds [Reiter and Rubin, 1997]. In Chameleon, the tuple $[IP_{\gamma_i}, IP_{\gamma_{i+1}}, p_{\#\gamma_i,\gamma_{i+1}}]$ identifies a path connection between two devices γ_i and γ_{i+1}.

6.3.4 Chameleon Protocol Description

A Chameleon device γ_i is a local proxy server that follows the state transition diagram presented in Figure 6.4. The role of a Chameleon device is threefold:

- a Chameleon device γ_i can serve as the user's local proxy to which the user's applications forward their data, θ. In such a case, the device is the first device on the anonymous path, thus $\gamma_i = \gamma_s$. This condition is represented by the "Handle Forward θ" state in Figure 6.4, which in turn can be expanded to the diagram in Figure 6.5.
- a Chameleon device γ_i can serve as an intermediary peer and thus forwards messages m_{γ_i,γ_j} in one or more anonymous paths. This situation

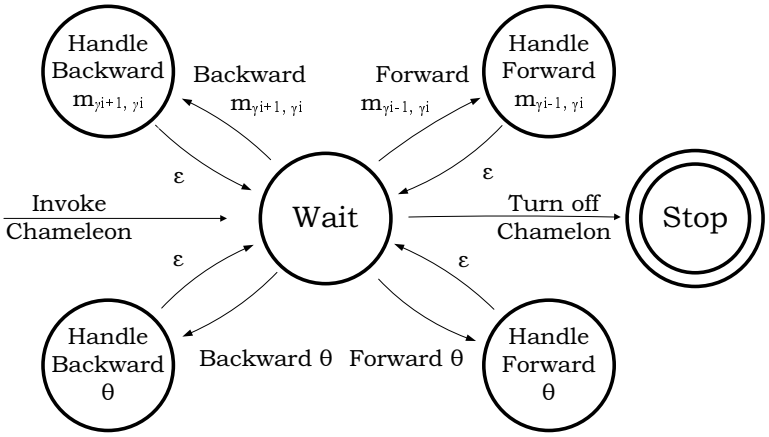

Figure 6.4: The Chameleon main state transition diagram for each device in the Chameleon framework. A device in Chameleon may be the first device of an anonymous path, γ_s, an intermediary device, γ_i, or the last device of an anonymous path, γ_{last}, depending on the type of the incoming message.

is represented by the "Handle Forward $m_{\gamma_{i-1},\gamma_i}$" and "Handle Backward $m_{\gamma_{i+1},\gamma_i}$" states in Figure 6.4. The former case can be expanded to the state transition diagram in Figure 6.6, which refers to messages m_{γ_i,γ_j} being forwarded towards the destination γ_d, and the latter case to the diagram in Figure 6.9, which refers to messages being transmitted in the backward direction, i.e., towards the sender γ_s.

- a Chameleon device can act as the last peer in an anonymous path, γ_{last}. In this case, it acts as a proxy server towards the destination device $d \in D$. This circumstance is represented by the "Handle Backward θ" state in Figure 6.4. This state can be expanded into the state transition diagram presented in Figure 6.8.

In the remainder of this section, the protocol details are keyed out by first outlining the anonymous path establishment, followed by the description of how data is sent from $\gamma_s \in \Gamma$ to $d \in D$, and, finally by depicting how anonymous paths are repaired in the event of the rupture of such a path. The state transition diagrams are also detailed in the remainder of this section.

The Establishment of Anonymous Paths

In the Chameleon framework, anonymous paths are established as described in this section. It is initially assumed that the Chameleon relay table is empty, i.e., there is no entry in the relay table for the designated logical address IP_d of

the destination device $d \in D$. The process of establishing an anonymous path works as follows:

i. The process of establishing an anonymous path for a given device $\gamma_s \in \Gamma$ is initiated when the Chameleon overlay receives application data θ from the application layer.

   After receiving the application data θ, the Chameleon overlay randomly selects a device $\gamma_1 \in \Gamma$, as depicted in the state "Select γ_1" from the state transition diagram presented in Figure 6.5. The information regarding the set Γ is obtained from any device running a directory service $\phi_i \in \Phi$. If the device γ_s has fresh information regarding the set Γ, it may use this information instead of contacting a directory service. The devices γ_s and γ_1 establish a symmetric encryption key k_{γ_s,γ_1} through a secure transport protocol. All further communication between the devices γ_s and γ_1 is performed through the established secure session.

   After the establishment of the secure session, the sender γ_s assembles the message m_{γ_s,γ_1} and forwards it to γ_1, as shown in the state "Send m_{γ_s,γ_1} to γ_1" in Figure 6.5. If γ_s is not able to send the message m_{γ_s,γ_1} to γ_1, γ_s replaces γ_1 with a randomly selected device from the Γ set. This new randomly selected device assumes the role of γ_1, and the process of establishing a secure session and sending the message m_{γ_s,γ_1} is repeated until γ_s succeeds to send m_{γ_s,γ_1};

ii. After receiving a message $m_{\gamma_{i-1},\gamma_i}$, an intermediary device $\gamma_i \, \forall i \in \mathbb{N}^*$ follows the state transition diagram presented in Figure 6.6. The device γ_i first decrypts the message $m_{\gamma_{i-1},\gamma_i}$ and checks its Chameleon relay table.

   If there is no corresponding entry in the Chameleon relay table for the pair $[IP_{\gamma_{i-1}}, p_{\#\gamma_{i-1},\gamma_i}]$ corresponding to the logical address of the preceding device and the previous path identifier associated with the incoming message $m_{\gamma_{i-1},\gamma_i}$, a biased coin is tossed. This procedure is depicted in the "Toss biased coin" state in the state transition diagram presented in Figure 6.6. After the toss of the biased coin, there are two possible outcomes:

   (a) The result indicates the termination of the anonymous path. Thus, the application data θ is extracted from the message $m_{\gamma_{i-1},\gamma_i}$ and the data is forwarded to the destination device $d \in D$. In such a case, γ_i becomes the last device in the anonymous path, and, thus, $\gamma_i = \gamma_{last}$. This step is concluded by updating the Chameleon relay table to indicate the end of this anonymous path. The resulting entry in the Chameleon relay table is shown in Figure 6.7.

   (b) The result does not indicate the termination of the anonymous path. Instead, the anonymous path is extended by one hop and a new device γ_{i+1} is randomly selected from the set Γ. A new message $m_{\gamma_i,\gamma_{i+1}}$

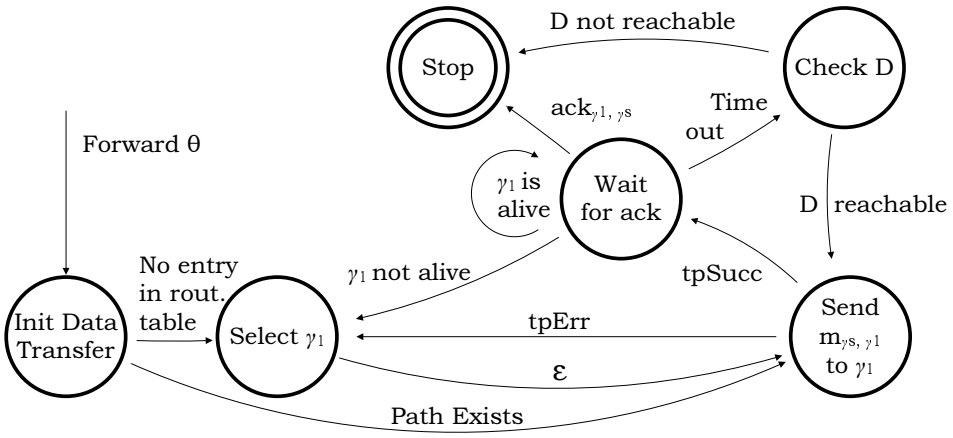

Figure 6.5: State transition diagram for a device $\gamma_s \in \Gamma$, which is the initiator of an anonymous path. The device γ_s receives the application data θ from the application layer sitting above the Chameleon overlay. The acronyms tpSucc *and* tpErr *used in this section denote transitions indicating whether the sending of a message was accomplished successfully (*tpSucc*) or not (*tpErr*). Such a functionality might be implemented by the transport layer positioned below the Chameleon overlay, if such transport protocol is connection-oriented, such as TCP. In the case of a connectionless transport protocol, and, thus, in the absence of acknowledgment messages in the transport layer, all transmissions are assumed to be accomplished successfully.*

is assembled and forwarded to γ_{i+1}, and a path identifier is associated with this connection. This procedure results in the update of the Chameleon relay table to reflect the extension of the anonymous path to the device γ_{i+1}. Moreover, the arrival of message $m_{\gamma_i,\gamma_{i+1}}$ at the device γ_{i+1} causes the described procedure to be repeated.

The aforementioned procedure sets up an anonymous path that begins at source of the application data θ, γ_s, and ends at γ_{last}, where γ_s and γ_{last} are interconnected by zero or more intermediary devices in Γ.

Sending and Forwarding Data

Assuming that there is an already established anonymous path linking γ_s to γ_{last} that was established according to state transition diagrams presented in Figures 6.5 and 6.6, the application data θ is forwarded from $\gamma_s \in \Gamma$ to the destination device $d \in D$ as follows:

- The Chameleon overlay of the device γ_s receives the application data θ

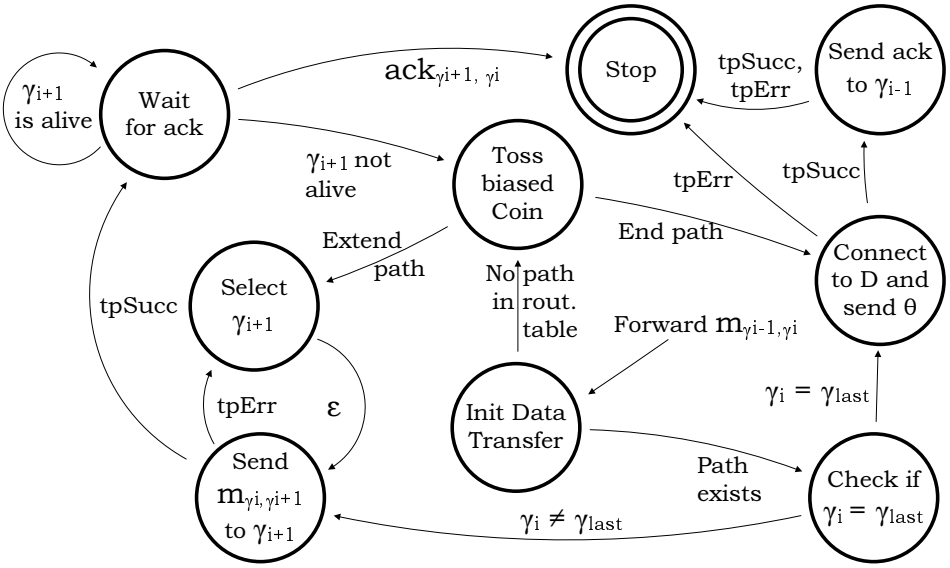

Figure 6.6: State transition diagram for a device γ_i that receives a message $m_{\gamma_{i-1},\gamma_i}$ from a device γ_{i-1}. The device γ_i tosses a biased coin and the result of this toss determines if the anonymous path should be further extended or if the application data should be forwarded to the destination device $d \in D$. The process of anonymous path repairing is also depicted in this diagram. The anonymous path repairing is triggered if the device γ_{i+1} becomes unavailable, which results in a new toss of the biased coin.

from the application layer. The final destination of the application data θ is the destination device $d \in D$. The device γ_s checks the Chameleon relay table and verifies which entry of the table, i.e., anonymous path, is associated with the logical address of the destination device. This entry of the Chameleon relay table also has the logical address and path identifier of the next device in the anonymous path, γ_1. The device γ_s assembles a message m_{γ_s,γ_1} and sends it to the next device of the anonymous path, γ_1, as depicted in the "Send Message m_{γ_s,γ_1} to γ_1" state in Figure 6.5;

- An intermediary device γ_i $\forall i \in \mathbb{N}^*$ is positioned in the anonymous path between γ_s and γ_{last}. An incoming message $m_{\gamma_{i-1},\gamma_i}$ is handled according to the state transition diagram shown in Figure 6.6. The intermediary device γ_i decrypts the message $m_{\gamma_{i-1},\gamma_i}$. The device γ_i checks the Chameleon relay table and verifies which entry of the table, i.e., anonymous path, is associated with the pair $[IP_{\gamma_{i-1}}, p_{\#\gamma_{i-1},\gamma_i}]$, i.e., the logical address of γ_{i-1} and the previous path identifier. This entry of the Chameleon relay table also

IP_d	$IP_{\gamma_{i-1}}$	$p_{\#\gamma_{i-1},\gamma_i}$	NULL	NULL	TTL

Figure 6.7: An entry in the Chameleon relay table indicating the end of the anonymous path. The 4th and the 5th fields of this entry, corresponding to the logical address of the succeeding device and the forward path identifier, are empty, i.e., NULL, to indicate the end of an anonymous path.

contains the logical address of the next device in the anonymous path, γ_{i+1}, and the path identifier associated with this anonymous path. The device γ_i assembles the message $m_{\gamma_i,\gamma_{i+1}}$ and forwards it to the next device in the anonymous path, γ_{i+1}.

Eventually, $\gamma_i = \gamma_{last-1}$, and a message $m_{\gamma_{last-1},\gamma_{last}}$ will be received by the last device in the anonymous path, γ_{last}. This device then sends the application data θ to the destination device $d \in D$. This data is sent to the destination device either encrypted or unencrypted, depending on the destination type, as presented in Section 6.3.1. Provided that the connection with the destination device d was successful, an acknowledgment message $ack_{\gamma_{last},\gamma_{last-1}}$ is sent backwards along the anonymous path to acknowledge the first device in the anonymous path, γ_s, that the application data θ was successfully delivered to the destination device d;

- To send application data in the backward direction, the device $d \in D$ sends the reply message through the already opened connection to $\gamma_{last} \in \Gamma$ during the forwarding data procedure, where γ_{last} is a device situated on one end of an anonymous path used to forward data from γ_s to the destination device. Such a process is shown in the state transition diagram presented in Figure 6.8.

  The device γ_{last} receives the application data θ from d, verifies which anonymous path and which connection are associated with the destination device d, and assembles a message $m_{\gamma_{last},\gamma_{last-1}}$. This message is sent to

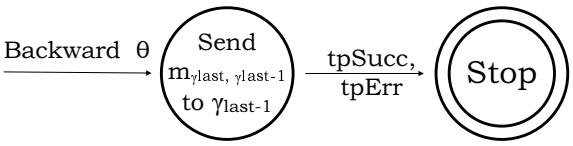

Figure 6.8: State transition diagram invoked in the Chameleon device γ_{last}, which is positioned in the end of an anonymous path, to send application data θ in the backward direction, i.e., towards the Chameleon device γ_s, which is located in the other end of the anonymous path in relation to γ_{last}.

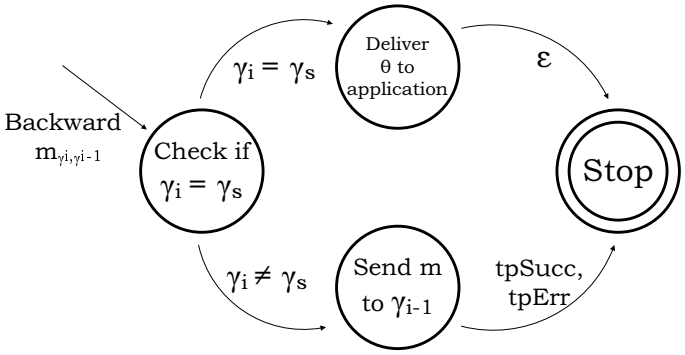

Figure 6.9: State transition diagram invoked by an intermediary device γ_i, which is located in the anonymity path in between γ_s and γ_{last}, or a device γ_s, if $\gamma_i = \gamma_s$. After receiving a message $m_{\gamma_{i+1},\gamma_i}$, the device γ_i verifies if the application data θ should be delivered to the application layer, and, thus, $\gamma_i = \gamma_s$, or a new message $m_{\gamma_i,\gamma_{i-1}}$ has to be assembled and sent to a device γ_{i-1}, and, thus, $\gamma_i \neq \gamma_s$.

γ_{last-1} on the anonymous path in the backward direction, i.e., in the opposite direction in relation to the set up of the anonymous path. There is no acknowledgment for messages travelling in the backward direction. Acknowledgment messages are important for repairing anonymous paths, but to repair the anonymous path for messages travelling in the backward direction would require a device γ_i to possess an extended knowledge about on the anonymous path. Thus, the finite state machine in Figure 6.4 returns to the "Wait" state, independent of whether or not it was possible to send the message to γ_{last-1}. This process is repeated at each intermediary device γ_i until the message is received at the other endpoint of the anonymous path, i.e., by the device γ_s, as depicted in Figure 6.9. If a timeout threshold is exceeded, the "Check D" state is entered, as shown in Figure 6.5. In this state, the device γ_s checks the status of the destination device $d \in D$. Such a verification should be possible since the ad hoc network is assumed to be a service-based network. The retransmission timeout should on one hand be large enough to allow intermediary devices in the anonymous path to repair the path if necessary, but on the other hand it should not be too large, since a large value might result in a negative impact in the overall protocol performance.

Repairing Anonymous Paths

In the Chameleon protocol, there are two cases that trigger the process of anonymous path repairing. The description of these cases are presented as follows:

- if an intermediary device $\gamma_i \mid \gamma_i \neq \gamma_s \wedge \gamma_i \neq \gamma_{last}$ does not manage to successfully send a message $m_{\gamma_i,\gamma_{i+1}}$ to the device γ_{i+1} positioned after it in a given anonymous path, or;

- if an intermediary device $\gamma_i \mid \gamma_i \neq \gamma_{last}$ is waiting for an acknowledge message $ack_{\gamma_{i+1},\gamma_i}$ from a succeeding device γ_{i+1} in a given anonymous path, and realizes that such a device γ_{i+1} is not responding[4]. Such an operation is illustrated in the state transition diagrams presented in Figures 6.5 and 6.6. The state "Wait for *ack*" of these diagrams asserts that the next device in the anonymous path, γ_{i+1}, is still responding. The implementation of such path repairing functionality is built by regularly polling the device γ_{i+1} by the previous device in the anonymous path γ_i.

The transition "γ_{i+1} not alive" in the state transition diagrams presented in Figures 6.5 and 6.6 indicates that the process of path repairing has been triggered. Thus, a device $\gamma_i \mid \gamma_i \neq \gamma_s \wedge \gamma_i \neq \gamma_{last}$ tosses the biased coin again and either forwards the application data θ directly to the destination device $d \in D$ or extends the anonymous path by selecting a device $\gamma_{i+1} \in \Gamma$ as its successor in such an anonymous path. Thus, the anonymous path is reestablished from the point of breach rather than from the beginning.

No explicit anonymous path destruction is conducted after a communication session via the anonymous path has ended. Instead, the TTL field, which is a decremental counter, ensures that inactive anonymous path entries are removed from the table when the counter reaches zero. The TTL field is part of the Chameleon relay table, presented in Figure 6.3.

6.4 Attacker Model

The Chameleon attacker model assumes that all participating devices (including the attackers) have similar omnidirectional transponders, and that the transmission and reception range of such transponders is also similar. Such an assumption is important for the anonymity analysis of the Chameleon protocol presented in Section 7. The following attacker types are included in the Chameleon attacker model:

- *local observers* — a local observer $\psi_{obs} \in \Psi$ is a passive observer that can eavesdrop the radio communication of the initiator of an anonymous path, $\gamma_s \in \Gamma$, i.e., ψ_{obs} is within the radio reception range of γ_s;

- *malicious insiders* — a malicious insider $\gamma' \in \Gamma' \mid \Gamma' \subset \Gamma$, where Γ' is the set whose elements are malicious insiders that may collaborate and try

[4]In practice, if $\gamma_i = \gamma_s$, the whole anonymous path is rebuild. Thus, the process of repairing is exactly the same process of setting up a completely new anonymous path.

to occupy all positions of an anonymous path. If a subset of collaborating devices of the set Γ' succeeds in occupying all positions in an anonymous path ahead of γ_s, such malicious insiders can compromise the anonymity properties of γ_s;

- *malicious outsiders* — a malicious outsider $\psi' \notin \Gamma$ is a malicious device whose objective is to place itself between a pair of devices, $\gamma_i \in \Gamma$ and $\gamma_{i+1} \in \Gamma$, that exchange data through a given anonymous path, i.e., to be part of the routing path in the network layer that is connecting those two devices that are part of the set Γ;
- *destination* — a destination device $d' \in D$ whose objective is to identify the original source of an incoming application data θ, and, thus, uniquely identify the device γ_s from all possible devices that are part of the set Γ, and;
- *malicious directory servers* — a malicious directory server $\phi' \in \Phi' \mid \Phi' \subseteq \Phi$, where Φ' is the set of whose elements are malicious devices running a directory service. This set of attackers misuse the distribution of information regarding the set Γ. They may collaborate and announce different subsets of Γ in an attempt to deploy a partition attack. Elements of the set Φ may also distribute a reduced set Γ in order to increase the percentile of malicious insiders $\gamma' \in \Gamma'$ in the set announced to other participants of the Chameleon framework.

The Chameleon attacker model differs from the one used in the Crowds protocol, since it also includes the malicious directory servers and malicious outsiders. The Chameleon attacker model is suitable for ad hoc networks since it includes devices routing and forwarding data at the lower layers that are not part of the set Γ.

6.5 Theoretical Analysis

In Section 4.2.1, a list of security and privacy requirements for anonymous communication mechanisms in ad hoc network environments was presented considering the characteristics of ad hoc networks presented in Sections 2.1 and 2.2 [Andersson et al., 2005b]. Such a list of requirements was defined in terms of anonymity properties, fairness, network performance, network architecture, mobility, and scalability. In this section, we list such requirements and discuss to what extent the Chameleon framework meets them:

- *strong anonymity properties* — this requirement states that an anonymous overlay network should provide adequate protection against malicious users and different types of eavesdroppers. The Chameleon framework provides sender and relationship anonymity against local observers

$\psi_{obs} \in \Psi$. Unlike the Crowds protocol [Reiter and Rubin, 1997], Chameleon enables both link-to-link and end-to-end encryption for certain destination types, as presented in Section 6.3.1. However, the Chameleon framework does not provide protection against global observers, since such protection would inevitably result in a negative impact on the performance of the Chameleon protocol. The anonymity analysis of the Chameleon protocol is further detailed in the Section 7;

- *fair distribution of workload among the participating devices* — an anonymous overlay network should be fair regarding the distribution of workload among the participants. A possible source for unfairness in the Chameleon protocol is the additional workload that has to be performed by the elements of the set Φ, i.e., the devices running a directory service. Nonetheless, the fairness property could be improved by allocating directory servers dynamically and introducing some rewarding system for the directory servers [Buttyán and Hubaux, 2003], or having all the devices in the set Γ to announce their presence to the other devices of the set Γ in the ad hoc network using a controlled flooding protocol for instance;

- *acceptable performance* — in order to reduce computational overhead and increase battery lifetime, an anonymous overlay network should generate as few control messages as possible and perform as few public key operations as possible. The Chameleon protocol uses public key encryption sparsely and does not use layered encryption. Assuming that a device $\gamma_s \in \Gamma$ has knowledge about the set Γ and L denotes the length of the anonymous path that connects γ_s to γ_{last}, a total of $(2 \times L)$ public key operations and $((2 \times L) - 1)$ Chameleon protocol messages are needed for establishing such an anonymous path. The Mix Route Algorithm [Jiang et al., 2004] presented in Section 2.4.2 uses layered public key encryption for the establishment of the anonymous path and for the exchange of messages through the anonymous path. In contrast, the Chameleon protocol makes use of public-key encryption only for the establishment of a secure tunnel between two consecutive devices of an anonymous path[5]. The protocol overhead is therefore low in comparison to the Mix Route Algorithm. Finally, the Chameleon framework avoids the use of dummy traffic, since the performance cost of such a mechanism is high, especially for ad hoc environments and battery-driven devices;

- *peer-to-peer model during its operational phase* — mobile ad hoc networks are most often assumed to operate without the aid of central services [Corson and Macker, 1999]. Unlike e.g., Crowds, Chameleon is a peer-to-peer compliant protocol, even though it is required that all devices that

[5]Assuming the knowledge of the set Γ, obtained from a device $\phi \in \Phi$ running a directory service.

are part of the set Γ to agree on the value of the probability of forwarding p_f. The use of different values for the probability of forwarding may result in the degradation of the expected anonymity properties, since the value p_f is directly related to the expected length of the anonymous path;

- *dynamic topology* — in most proposed ad hoc network scenarios, it is assumed that devices frequently enter and leave the network. The Chameleon framework is compliant to dynamic topologies since, among other properties, it provides an optimized path repairing process in the forward direction. An anonymous path is repaired only from the point of rupture, in contrast to approaches that require the setting up of an entirely new anonymous path, and;

- *scalability* — the workload on each participant in Chameleon remains constant as the number of participants grows, in a similar matter as in Crowds. It is proved by Reiter and Rubin [1997] that for each device in the network, the expected number N of anonymous paths a given device will be participating in at a particular instant of time is given by Equation 6.6:

$$N = \frac{1}{(1 - p_f)^2} \cdot \left(1 + \frac{1}{|\Gamma|}\right) \tag{6.6}$$

6.6 Summary

This chapter introduced the Chameleon framework, which includes the Chameleon protocol, which is an anonymous communication mechanism. Chameleon is a low-latency anonymous overlay network tailored for ad hoc networks and provides a reduced amount of control messages in comparison to other anonymous overlay networks and anonymous path repairing. Chameleon does not use dummy traffic or layered encryption to provide an acceptable compromise between the anonymity properties and the performance cost as to be presented in the following chapters. The Chameleon protocol is based on Crowds. Still, the Chameleon framework differs from Crowds in a number of ways, including: the possibility of end-to-end encryption between the sender and recipient devices, a distributed service discovery mechanism, and an attacker model consistent with ad hoc network environments.

The combination of the Chameleon framework and privacy-friendly identifiers can deliver protection against Sybil attacks in the anonymity set. Sybil identifiers in the anonymity set can be detected if the self-certified Sybil-free pseudonyms presented in the Chapter 5 are used.

In the following chapter, the anonymity analysis of the Chameleon protocol is presented. The following aspects are evaluated in such an analysis:

sender anonymity, receiver anonymity and relationship anonymity, i.e., sender-receiver unlinkability. The attacker model used in the anonymity analysis was defined in this chapter.

Chapter 7

Anonymity Analysis of the Chameleon Protocol

"Who are you?"
"Who? Who is but the form following the function of what, and what I am is a man in a mask."
"Well, I can see that."
"Of course you can. I'm not questioning your powers of observation. I am merely remarking upon the paradox of asking a masked man who he is."

Natalie Portman as Eevy Hammond and Hugo Weaving as V
— V for Vendetta (2006)

In the analysis of the Chameleon protocol presented in this chapter[1], the following aspects of anonymity are evaluated: sender anonymity, receiver anonymity, and relationship anonymity, i.e., unlinkability between the sender and the recipient.

The chapter is organized in six sections. Section 7.1 outlines the metric used to measure anonymity and Sections 7.2 to 7.6 present the anonymity analysis of the Chameleon protocol. The attacker model considered in the anonymity analysis consists of the following five types of attackers: local observers, malicious insiders, malicious outsiders, destination devices, and malicious devices hosting a directory service. The considered attacker model was detailed in Section 6.4.

[1]The anonymity analysis of Chameleon presented in this section previously appeared in [Martucci et al., 2006a] and [Andersson, 2008].

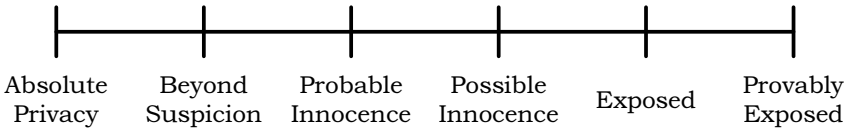

Figure 7.1: The degrees of anonymity according to anonymity metric introduced by the Crowds protocol [Reiter and Rubin, 1997].

7.1 Measuring Anonymity

The metric used to measure anonymity in this section is the same metric applied for evaluating the anonymity properties of the Crowds protocol [Reiter and Rubin, 1997]. In that metric each device is considered separately and the resulting range of values is a function of the cardinality of the anonymity set, $|\Gamma|$ and the cardinality of the set of malicious insiders $|\Gamma'|$, $\Gamma' \subset \Gamma$, among other parameters. In this section, the term "subject" is used to refer to a device γ_i that is part of the anonymity set Γ.

The degree of anonymity for a subject $\gamma_i \in \Gamma$ is expressed as $A_{\gamma_i} = 1 - P_{\gamma_i}$, where P_{γ_i} is the probability that γ_i is the originator of a particular message. The degree of anonymity A_{γ_i} is measured on a discrete scale ranging from *absolute privacy* to *provably exposed*, as presented in the Figure 7.1. This scale has the following points of interest:

- *absolute privacy* — the probability that a given subject $\gamma_i \in \Gamma$ is linked to a particular message is zero, and, hence, the degree of anonymity $A_{\gamma_i} = 1$;
- *beyond suspicion* — a device $\gamma_i \in \Gamma$ that is part of the anonymity set $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ is *beyond suspicion* if it appears no more likely than any other subject in the anonymity set of being linked to a particular message, that is, $A_{\gamma_i} = \min\{A_{\gamma_1}, \ldots, A_{\gamma_n}\}$;
- *probable innocence* — the probability that a given subject $\gamma_i \in \Gamma$ is linked to a particular message is less than $\frac{1}{2}$, and, thus, the degree of anonymity $A_{\gamma_i} \geq 0.5$;
- *possible innocence* — there is a non-trivial chance that a particular subject $\gamma_i \in \Gamma$ is not the originator of a given message such as the degree of anonymity $A_{\gamma_i} > \nabla_{limit}$, where $0 < \nabla_{limit} < 0.5$);
- *exposed* — a given subject $\gamma_i \in \Gamma$ can be unambiguously linked to a given message, and, hence, the degree of anonymity $A_{\gamma_1} = 0$, and;
- *provably exposed* —- the degree of anonymity $A_{\gamma_1} = 0$ as above and, furthermore, it could be proved to a third party that the subject $\gamma_i \in \Gamma$ is linked to the given message.

The next section presents an analysis of the degree of anonymity obtained by Chameleon users considering the attacker model defined in Section 6.4.

7.2 Anonymity Against a Local Observer

This section quantifies anonymity according to the metric presented in Section 7.1. The degree of sender anonymity, receiver anonymity, and relationship anonymity is evaluated against an attacker acting as a local observer $\psi_{obs} \in \Psi$. A local observer ψ_{obs} is a passive observer that can eavesdrop the radio communication of the initiator of an anonymous path, $\gamma_s \in \Gamma$. This means that γ_s is within the radio range of ψ_{obs}, as defined in Section 6.4.

Sender Anonymity

Since the device γ_s is within the radio range of local observer ψ_{obs}, ψ_{obs} can theoretically eavesdrop all information transmitted from the device γ_s. However, except during periods of low network traffic, ψ_{obs} cannot tell whether γ_s was the originator of such messages, as the device γ_s could instead be forwarding messages that were originated from another device $\gamma_i \in \Gamma$. Further, the local observer ψ_{obs} is not capable of recognizing an already observed data flow that eventually reappear within its radio reception range since all data flows are link encrypted between each pair of Chameleon devices.

In periods of low network traffic there is, however, a nontrivial risk that a ψ_{obs} may suspect that γ_s is the originator of the observed messages, e.g., by using traffic analysis. Still, the local observer ψ_{obs} cannot know for certain whether γ_s is the original source of the application data being forwarded in an anonymous path, as the device γ_s might be communicating with a device that is hidden from the local observer ψ_{obs}, i.e., a hidden terminal. The hidden terminal problem is depicted in Figure 7.2. Thus, the degree of sender anonymity amounts to the degree of *possible innocence*.

Receiver Anonymity

To compromise receiver anonymity the local observer ψ_{obs} must be within the radio range of the destination device $d \in D$ as well as the last device of the anonymous path $\gamma_{last} \in \Gamma$. In such a case, ψ_{obs} can conclude that a given message m is intended for a given destination device d. However, the larger the ad hoc network is, the less the likelihood of γ_s, d, and γ_{last} being subsumed by the radio range of the local observer ψ_{obs}. Thus, the degree of receiver anonymity approaches the degree of *beyond suspicion* for ad hoc networks where the physical size of the network is larger than the radio range of the attacker, which is a reasonable assumption given our attacker model.

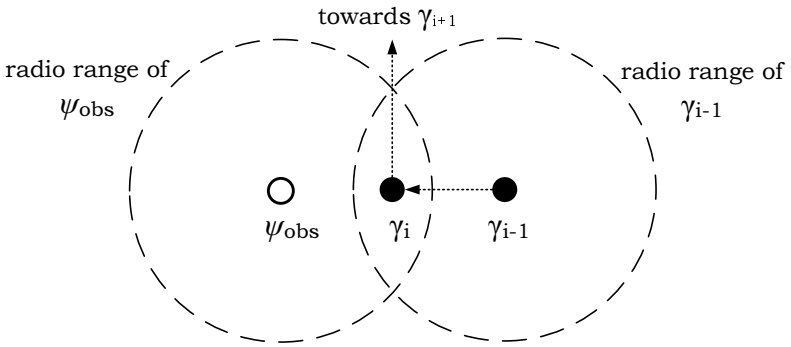

Figure 7.2: The hidden terminal problem. In this example, the local observer device $\psi_{obs} \in \Psi$ is not able to determine for sure whether a message m, being transmitted from a device $\gamma_i \in \Gamma$ to another device $\gamma_{i+1} \in \Gamma$, was originated in the device γ_i or in another device $\gamma_{i-1} \in \Gamma$, which is located outside the radio range of ψ_{obs}. In the latter case, the device γ_i is just an intermediary device in an anonymous path connecting the devices γ_{i-1} and γ_{i+1}.

Relationship Anonymity

The local observer $\psi_{obs} \in \Psi$ is not able to link γ_s to the destination device $d \in D$, since ψ_{obs} does not have a complete knowledge regarding the network topology and the anonymous path. In addition, the link encryption between two consecutive devices in an anonymous path results in the change of the appearance of a message being forwarded. Except for the special case where the local observer ψ_{obs} can eavesdrop on the entire anonymous path connecting γ_s to γ_{last}, and the destination device $d \in D$, the degree of relationship anonymity amounts to *beyond suspicion* for a relatively large ad hoc network.

7.3 Anonymity Against a Malicious Insider

This section provides a quantification of the degree of anonymity, against a malicious insider $\gamma' \in \Gamma' \mid \Gamma' \subset \Gamma$, where Γ' is the set of malicious insiders that may collaborate and try to occupy all positions of an anonymous path. If a subset of collaborating devices of the set Γ' succeeds in occupying all positions in an anonymous path ahead of γ_s, such malicious insiders can compromise the anonymity properties of γ_s.

Sender Anonymity

A consequence of the probabilistic nature of the anonymous path setup is that a malicious insider $\gamma' \in \Gamma'$ on a given path will not be able to determine for certain if the previous device in such a path is the sender or not, that is if $\gamma_{i-1} = \gamma_s$ or $\gamma_{i-1} \neq \gamma_s$. This situation for the malicious insiders in Chameleon is similar to that of "collaborating jondos" in the Crowds protocol. Thus, the degree of sender anonymity against a malicious insider is *probable innocence*, provided that Equation 7.1 holds [Reiter and Rubin, 1997], where $|\Gamma'|$ denotes the cardinality of the set Γ'.

$$|\Gamma| \geq \frac{p_f}{(p_f - 0.5)} \cdot (|\Gamma'| + 1) \quad | \quad 0.5 < p_f < 1, \ \forall\, p_f \in \mathbb{R} \tag{7.1}$$

As result of Equation 7.1, a tradeoff between the anonymous path lengths and the probability of forwarding is defined. This relationship implies that by making the probability of forwarding high, i.e., closer to 1, the fraction of malicious insiders that can be tolerated increases in relation to the total number of devices in the anonymity set, eventually approaching half of the elements of the set Γ [Reiter and Rubin, 1997]. This equation can be rewritten to define the same inequality in terms of the fraction $(|\Gamma'|/|\Gamma|)$. The resulting equation is presented in Equation 7.2 and sets a minimum bound for the probability of forwarding p_f in relation to the fraction $(|\Gamma'|/|\Gamma|)$ [Andersson et al., 2004].

$$p_f \geq \frac{1}{2 \cdot \left(1 - \left(\frac{|\Gamma'|}{|\Gamma|}\right)\right)} \quad | \quad |\Gamma| \gg 1, \ 0 \leq \left(\frac{|\Gamma'|}{|\Gamma|}\right) < 0.5, \ \ \forall \left(\frac{|\Gamma'|}{|\Gamma|}\right) \in \mathbb{R} \tag{7.2}$$

Naturally, a high p_f implies longer expected path lengths. The relationship between the fraction $(|\Gamma'|/|\Gamma|)$ and the expected path length L_{exp} can also be derived from Equations 7.1 and 6.1. The resulting equation is then expressed in terms of the expected path length L_{exp} and the cardinalities of the sets Γ and Γ', as shown in Equation 7.3, for $|\Gamma| \gg 1$ and assuming that the probability of forwarding p_f has the minimum value allowed by the Equation 7.2.

$$L_{exp} = \frac{1}{1 - 2 \cdot \left(\frac{|\Gamma'|}{|\Gamma|}\right)} + 2 \quad | \quad |\Gamma| \gg 1, \ 0 \leq \left(\frac{|\Gamma'|}{|\Gamma|}\right) < 0.5, \ \ \forall \left(\frac{|\Gamma'|}{|\Gamma|}\right) \in \mathbb{R} \tag{7.3}$$

The Equations 7.2 and 7.3 are plotted in Figure 7.3. This figure presents the expected path length L_{exp} and p_f in relation to the fraction $(|\Gamma'|/|\Gamma|)$.

It can further be noted that although not affecting the degrees of anonymity per se, privacy-friendly identifiers, such as the self-certified Sybil-free pseudonyms, makes it more costly for malicious insiders to take control of a sufficiently large portion of the network.

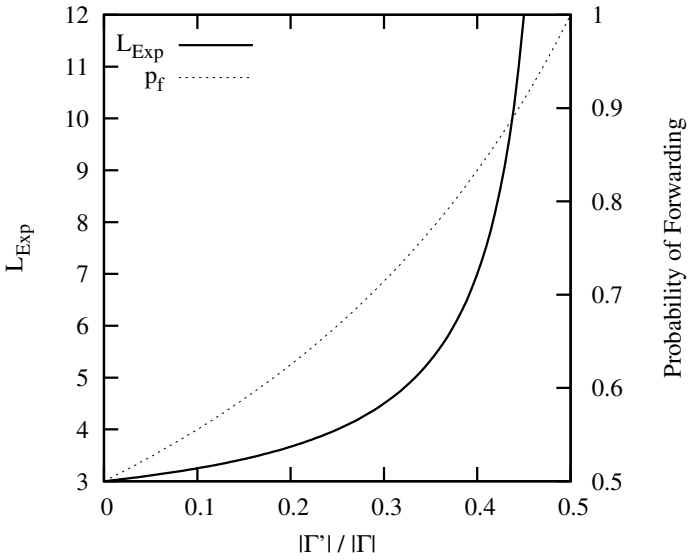

Figure 7.3: These two curves presents the expected path length L_{exp} and the probability of forwarding p_f in relation to the ($|\Gamma'|/|\Gamma|$) ratio. The p_f curve refers to the minimum p_f for a given ($|\Gamma'|/|\Gamma|$) ratio, and the L_{exp} curve illustrates the expected path length related to the ($|\Gamma'|/|\Gamma|$) ratio.

Receiver Anonymity

A malicious insider γ' that is part of an anonymous path always knows the logical address of the destination device IP_d for the current path, since this information is encapsulated in the message $m_{\gamma_i,\gamma_{i+1}}$. In such cases, the degree of anonymity is *exposed*. However, if none of the $|\Gamma'|$ malicious insiders are part of the anonymous path, the degree of anonymity is *absolute privacy*. The probability that none of the $|\Gamma'|$ malicious insiders are part of a particular path, and, thus, that the degree of receiver anonymity is *absolute privacy* is given by the Equation 7.4, where L_{exp} denotes the expected length of the anonymous path.

$$P(\textit{absolute privacy}) \;=\; 1 - P(\textit{exposed}) \;=\; \left(\frac{|\Gamma| - |\Gamma'|}{|\Gamma|}\right)^{L_{exp}-1} \tag{7.4}$$

Relationship Anonymity

Assuming that γ' is part of the anonymous path and knows the logical address of the destination device $d \in D$, a malicious insider γ' can only break the properties of relationship anonymity by breaking the properties of sender anonymity.

Thus, the degree of relationship anonymity is *probable innocence* provided that Equation 7.1 holds.

7.4 Anonymity Against a Malicious Outsider

A malicious outsider $\psi' \notin \Gamma$ is a malicious device whose objective is to place itself between a pair of devices, $\gamma_i \in \Gamma$ and $\gamma_{i+1} \in \Gamma$, that exchange data through a given anonymous path, i.e., to be part of the routing path in the network layer that is connecting two devices in Γ.

Sender Anonymity

To evaluate sender anonymity against a malicious outsider ψ', the following events need to be defined:

- E_{route} denotes the event that a malicious outsider $\psi' \in \Psi$ is selected to route messages being transmitted between $\gamma_i \in \Gamma$ and $\gamma_j \in \Gamma$. The probability that the malicious outsider ψ' can force the event E_{route} to occur is likely to be low, since ψ' needs to possess information about the physical locations of γ_i and γ_j, as well as their radio ranges, to be used as an intermediary routing link between γ_i and γ_j. Alternatively, the malicious outsider ψ' could misuse the underlying ad hoc routing protocol to deceive γ_i and γ_j so that it appears that ψ' constitute an intermediary path between γ_i and γ_j;

- E_{dir} denotes the event that a malicious outsider $\psi' \in \Psi$ can conclude that γ_i precedes γ_j in a given anonymous path. A malicious outsider ψ' eavesdropping on the wireless communication might conclude from a message m_{γ_i,γ_j} transmitted in the wireless medium that a device γ_i precedes another device γ_j in a given anonymous path. However, the mobile behavior of the devices in an ad hoc network prevents the malicious outsider ψ' from knowing for certain that the observed message m_{γ_i,γ_j} was not preceded by a number of other messages being transmitted in the opposite direction, i.e., from γ_j to γ_i;

- $E_{\gamma_i=\gamma_s}$ denotes the event that a malicious outsider $\psi' \in \Psi$ concludes that a device $\gamma_i = \gamma_s$.

Although the probability of the events E_{route} and E_{dir} occurring concurrently, that is $E_{route} \wedge E_{dir}$, is likely to be low, it is still necessary to consider such a probability for calculating a lower bound for the degree of sender anonymity. In such a case, the probability of the event $E_{\gamma_i=\gamma_s}$, given the events $E_{route} \wedge E_{dir}$, can be expressed as the inverse of the expected number of hops H_{exp}, i.e., the

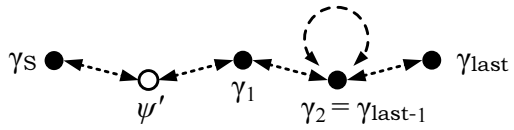

Figure 7.4: An illustration of an anonymous path that extends from a sending device $\gamma_s \in \Gamma$, which is the source of the application data θ, to the device $\gamma_{last} \in \Gamma$, and has a local loop in the device $\gamma_2 = \gamma_{last-1} \in \Gamma$ that precedes the device γ_{last} in the anonymous path. In this figure, the devices γ_s and γ_1 that are part of the anonymous path are connected through a malicious outsider $\psi' \in \Psi$, which routes and forwards messages being transmitted between γ_s and γ_1. This figure illustrates the event $E_{\gamma_i=\gamma_s} \mid E_{route} \wedge E_{dir}$.

expected path length $L_{exp} - 1$, since the attacker could be situated at any hop between two consecutive devices in Γ that are part of a given anonymous path.

The degree of anonymity for a device γ_i is $A_{\gamma_i} = 1 - P_{\gamma_i}$. In addition, the degree of anonymity against a malicious outsider $\psi' \in \Psi$ can be denoted as $1 - P(E_{\gamma_i=\gamma_s} \mid E_{route} \wedge E_{dir})$. The probability of the event $E_{\gamma_i=\gamma_s} \mid E_{route} \wedge E_{dir}$, is given in Equation 7.5, where R_L denotes the expected reduction in the actual number of hops that are caused by local loops. A local loop occurs if a device selects itself as its successor, as depicted in Figure 7.4.

$$P(E_{\gamma_i=\gamma_s} \mid E_{route} \wedge E_{dir}) = \frac{1}{H_{exp}} = \frac{1}{(L_{exp} - 1) - R_L} \tag{7.5}$$

For an expected path length $L_{exp} \geq 4$ and an anonymity set, i.e., the cardinality of the set Γ, $|\Gamma| \geq 3$, it is possible to affirm, in relation to the expected number of hops, that $H_{exp} > 2$. Therefore, the malicious outsider $\psi' \in \Psi$ can expect that there are at least two different hops that it could be situated on, assuming $E_{route} \wedge E_{dir}$. In conclusion, the degree of anonymity is *probable innocence*, according to the Equation 7.5. The proof regarding this conclusions is presented in [Martucci et al., 2006a] and [Andersson, 2008].

According to Equation 6.1, an expected path length $L_{exp} \geq 4$ is obtained for a probability of forwarding $p_f \geq 2/3$. Nevertheless, for a large anonymity set $|\Gamma|$, the degree of sender anonymity is likely to approach *beyond suspicion*, since the probability of the occurrence of the events $E_{route} \wedge E_{dir}$ is low, and such events are a requirement, which is assumed to be fulfilled in the analysis presented in this section, for any malicious outsider ψ' to compromise the anonymity of a device $\gamma_s \in \Gamma$.

Receiver Anonymity

The malicious outsider $\psi' \in \Psi'$ is not able to obtain the logical address of the destination device, IP_d, directly from an eavesdropped message m_{γ_i,γ_j}, that is being transmitted from a device $\gamma_i \in \Gamma$ to a device $\gamma_j \in \Gamma$ since such a message is link encrypted using a symmetric shared key that is possessed only by the devices γ_i and γ_j. Thus, the degree of receiver anonymity is *possible innocence* if the expected path length $L_{exp} \geq 4$.

Relationship Anonymity

The Chameleon protocol assures that $\gamma_s \in \Gamma$ never communicates directly with the destination device $d \in D$. Thus, the degree of relationship anonymity is *beyond suspicion*.

7.5 Anonymity Against a Destination Device

A malicious destination device $d' \in D$ has as objective to disclose the identity of the source of an incoming application data θ, and, thus, uniquely identify the device γ_s from all possible devices that are part of the set Γ.

From the perspective of the malicious destination device $d' \in D$, γ_s could be any device $\gamma_i \in \Gamma$, since the length L of an anonymous path is always $L \geq 2$. Thus, both the degrees of sender anonymity and relationship anonymity are *beyond suspicion*. Naturally, the concept of receiver anonymity does not exist for the destination device, since the destination is the receiver.

7.6 Anonymity Against a Directory Server

A malicious directory server $\phi' \in \Phi' \mid \Phi' \subseteq \Phi$, where Φ' is the set of malicious devices running a directory service. This set of attackers misuse the distribution of information regarding the set Γ. Directory servers distribute the logical addresses of devices $\gamma_i \in \Gamma$. However, such a knowledge alone is useless for breaking the anonymity properties of devices in Γ, and, thus, the degree of anonymity against a malicious directory server is *absolute privacy*.

A malicious directory server ϕ' may, however, collaborate with other attacker types to increase their chance of succeeding with an attack. Thus, the malicious directory server ϕ' can distribute false information regarding the set Γ to the users of the Chameleon protocol, i.e., the elements of the set Γ. For instance, a malicious directory server ϕ' can distribute a set $\Gamma'' \subset \Gamma$ which is comprised mostly of malicious insiders $\gamma' \in \Gamma'$. The specification and evaluation of secure and efficient mechanisms that hinder malicious directory servers from performing such partitioning attacks is out of the scope of this dissertation,

but such mechanisms are usually comprised of one or more of the following strategies:

- *redundancy* — the more the directory servers in the set Φ, the stronger the protection against malicious directory servers in theory, since the probability that a device in Γ chooses a non-malicious directory server increases with a growing $|\Phi|$;
- *distributed reputation metrics* — this relates to mechanisms that assign trust values to the devices in the set Φ, so that a misbehaving directory server could be found a filtered out. A trust-based service discovery protocol for ad hoc networks that suits the Chameleon framework is described in [Martucci et al., 2004b], for instance, and;
- *cycling through different directory servers* — the continuous use of a given directory server $\phi_i \in \Phi$ for obtaining the list of the available elements of the set Γ is potentially harmful and, thus, should be avoided. As a general recommendation, Chameleon users $\gamma_i \in \Gamma$ should contact different directory servers, so that significant differences in the distributed instances of the set Γ could be detected.

7.7 Summary

This section presented the anonymity analysis of the Chameleon framework. The Crowds-based anonymity metric was used to quantify the anonymity provided by Chameleon. The anonymity of the Chameleon protocol was evaluated against the attacker model presented in Section 6.4. The attacker model that was considered consisted of the following five types of attackers: local observers, malicious insiders, malicious outsiders, destination devices, and malicious devices hosting a directory service. Table 7.1 summarizes the anonymity analysis presented in this section.

In the following chapter, the network performance of the Chameleon protocol is evaluated using a network simulator. The performance of Chameleon is measured by simulating an ad hoc network with 30 devices and at one-hop distance. The objective of such a simulation is to identify the amount of packets lost and the extra transmission delay introduced by the Chameleon protocol running in an ad hoc environment and to isolate such delays from other transmission delays caused by ad hoc routing protocols. The performance impact is also evaluated according to multiple values attributed to the probability of forwarding p_f.

Table 7.1: This table summarizes the degrees of sender anonymity, receiver anonymity, and relationship anonymity in the Chameleon protocol against: local observers, malicious insiders, malicious outsiders, and destinations.

	Sender Anonymity	Receiver Anonymity	Relationship Anonymity
Local observer	possible innocence	beyond suspicion for large networks	beyond suspicion for large networks
Malicious insider	probable innocence if $\|\Gamma\| \geq \frac{p_f}{(p_f - 0.5)} \cdot (\|\Gamma'\| + 1)$	absolute privacy $= \left(\frac{\|\Gamma\| - \|\Gamma'\|}{\|\Gamma\|}\right)^{L_{exp}-1}$	probable innocence
Malicious outsider	probable innocence if $L_{exp} \geq 4$ and $\|\Gamma\| \geq 3$	probable innocence if $L_{exp} \geq 4$ and $\|\Gamma\| \geq 3$	beyond suspicion
Destination	beyond suspicion for $\|\Gamma\| \geq 3$	—	beyond suspicion

Chapter 8

Anonymity and Performance Trade-offs

> "Wait a moment," William said. "I do not know why, but I have never seen a machine that, however perfect in the philosophers' description, is perfect in its mechanical functioning. Whereas a peasant's billhook, which no philosopher has ever described, always function as it should. . . ."
>
> *Brother William of Baskerville*
> *—The Name of the Rose (1980), Umberto Eco*

This chapter evaluates the network performance of the Chameleon protocol and elaborates on the trade-off between anonymity and performance. In the evaluation, a network simulator was used to simulate an ad hoc network with 30 devices equally distributed in a 44100 m^2 square area. Simulation results include the amount of packets lost, and the extra transmission delay introduced by the Chameleon protocol running in an ad hoc environment. In order to isolate such delays from other transmission delays caused by the underlying ad hoc routing protocol all devices are within one-hop distance from any other device in the ad hoc network. The performance impact is simulated using different values for the probability of forwarding, which determines the degree of anonymity protection, as presented in Chapter 7.

The remainder of this chapter is organized as follows. The simulation objectives are outlined in Section 8.1. In Section 8.2, the simulation environment is introduced, including the modifications in the used network simulator. The details regarding the simulation parameters, e.g., number of devices in the ad hoc network, area, data flow, the probability of forwarding p_f used, and number of

simulation runs, are given in Section 8.3. Finally, the results of the simulation runs are aggregated, presented, and analyzed in Section 8.4.

8.1 Objectives

The objective of the chapter is to analyze the results from the simulation runs and to identify the trade-off between anonymity, in terms of the resistance against malicious insiders, and performance, in terms of the expected cumulative distribution function of the end-to-end delay.

The network simulation was used to measure the end-to-end delay and the number of lost packets against an increasing value of the probability of forwarding p_f in an ad hoc network with devices running the Chameleon protocol. To set a basis for comparing the achieved results, a simulation scenario with no forwarding was also set up, i.e., where the sender device delivers the application data directly to the recipient device. Analytical modelling was used to obtain a general mathematical expression for describing the expected cumulative distribution function in terms of both the end-to-end delay and the resistance against malicious insiders, which is a function of the probability of forwarding p_f.

8.2 The Simulation Environment

This section is divided into two parts. In the first part, the distinct environments of the OPNET Modeler simulation tool are briefly introduced. The objective of this part is to introduce the different underlying building blocks of the simulation tool. In the second part, the implementation of Chameleon using the simulation tool is presented. The objective of this part is to present the simulation environment and the modifications implemented on standard simulation models during the development of the Chameleon protocol.

8.2.1 The Simulation Tool

The selected simulation tool was OPNET Modeler version 14.0 [OPNET]. Modeler is a discrete event simulation tool. Modeler is divided in distinct environments called editors. For the context of the implementation of the Chameleon protocol and the simulation of such an implementation, the most relevant editors in the Modeler simulation tool are the following:

- the *Project Editor* specifies the network topology and is used to configure some key characteristics of the device that are part of the topology, such as their location and their running services;

- the *Node Editor* specifies the internal structure of a device. This internal structure is composed of modules that represent particular functions in the device's behavior. The organization of such modules differs according to the modelled device. For some devices, such as the wireless network servers used in the simulation of Chameleon, these modules are organized in a similar matter as the TCP/IP stack. The Chameleon protocol is implemented as a module in a wireless network device;
- the *Process Editor* is used to define the behavior of a given module. A module is composed of process modules, which define the behavior and the aspects of the process module that are visible to its user. The behavior of process modules is determined using finite state machines. The deterministic finite state machines are represented using state transition diagrams. The behavior of the Chameleon protocol is thus implemented as a state transition diagram, and;
- the *Packet Format Editor* is used to define the fields of a formatted packet, i.e., a packet with the predefined fields, such as the packets used in the Chameleon protocol.

Modeler has other editors that were used in the implementation and simulation of the Chameleon protocol, such as the Simulation Sequence Editor and the Interface Control Information (ICI) Editor, but these editors are not fundamental for the understanding of the implementation and simulation of the Chameleon protocol and will therefore not be described here or mentioned in the remainder of this chapter.

8.2.2 Implementation of the Chameleon Protocol

The Chameleon protocol was implemented as an additional intermediate layer situated in between the application layer and the transport adaptation layer (tpal). Tpal is a basic and uniform interface between applications and transport layer models. The connectionless User Datagram Protocol (UDP) [Postel, 1980], was used as the standard transport protocol in the simulation of Chameleon. The literature shows that UDP has a better performance than TCP in single-hop and multi-hop static ad hoc networks, most probably because the connected-oriented characteristics of TCP, such as acknowledgments travelling in the backward direction [Rohner et al., 2005]. In addition, the objective of the simulation is to measure the end-to-end delay, and UDP provides the one-way traffic required for such a measurement.

The Chameleon protocol was integrated into an existing standard wireless server node model in OPNET Modeler version 14.0, the *wlan server adv* node model. This modified node model was later renamed to *wlan server overlay adv* node model and it is presented in Figure 8.1. The *wlan server overlay adv*

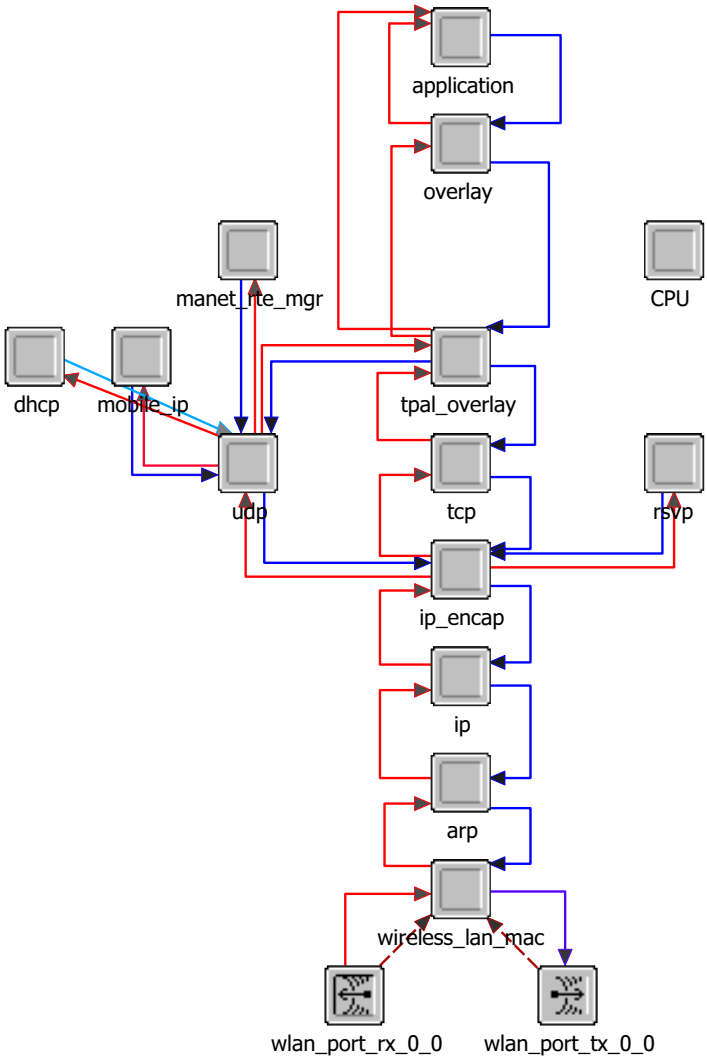

Figure 8.1: The wlan server overlay adv node model used in the simulation. The Chameleon protocol is implemented in the overlay processor. The overlay processor is positioned in between the application and tpal overlay processors. This node model is a variant of the wlan server adv node model, which is part of the standard node models in OPNET Modeler version 14.0. Although the implementation of Chameleon protocol is contained in the overlay processor, other modifications were required in other processors as well, such as tpal overlay, application, and udp processors.

node model is implemented with multiple processors, one radio receiver and one radio transmitter. The processors are organized according to the TCP/IP stack, with the application processor on the top and the radio transmitters on the bottom of the stack.

This wireless server node model was selected because it provides an advanced level of details on the amount of information that can be obtained from different layers. Furthermore, every device running the Chameleon protocol is required to behave as a client and a server device simultaneously, and such a functionality is available only on server devices in the wireless local area network devices in Modeler. The disadvantage of such a high level of detail is that it may degrade the performance of the simulator, since a large number of variables are needed to be taken into account during simulation time.

The remainder of this section is organized in three parts. First, the overlay processor, which contains the implementation of the Chameleon protocol, is introduced. Second, the modifications on other processors are briefly presented. Finally, the simplifications on the Chameleon framework are discussed and justified in the third part.

The Overlay Processor

The overlay processor is positioned between the application and the tpal overlay processors, as depicted in Figure 8.1. The behavior of the overlay processor is defined with the process model, i.e., as a state transition diagram, presented in Figure 8.2. This process model implements a simplified version of the Chameleon protocol, presented in Chapter 6. The simplifications in the Chameleon framework are listed in the end of this section.

The first three states in the state transition diagram presented in Figure 8.2 are *init*, *pre-start*, and *start*. These states are used for the initialization of the system parameters, such as the probability of forwarding p_f and the seed for the random number generator. Moreover, these three initial states are also used to discover the local address information, to obtain the addresses of the other Chameleon devices available in the simulation scenario, and to register the overlay processor in the list of available services in the transport adaptation layer. Following the execution of these three initial states, the *wait* state is reached. The *wait* state is an idle state that waits for incoming interrupts, i.e., packets, from either the application or the tpal processors. The expected interrupts in the *wait* state are incoming data from the tpal processor, incoming data from the application, or control messages, such as confirmations of open connections, indications of incoming connections, and indications of closing connections[1].

[1]The simulation tool requires the establishment of connections between tpal processors of distinct devices even if a connectionless protocol, such as UDP is used. Nevertheless, the establishment of such a connection between tpal processors is a logic connection used only by the simulation

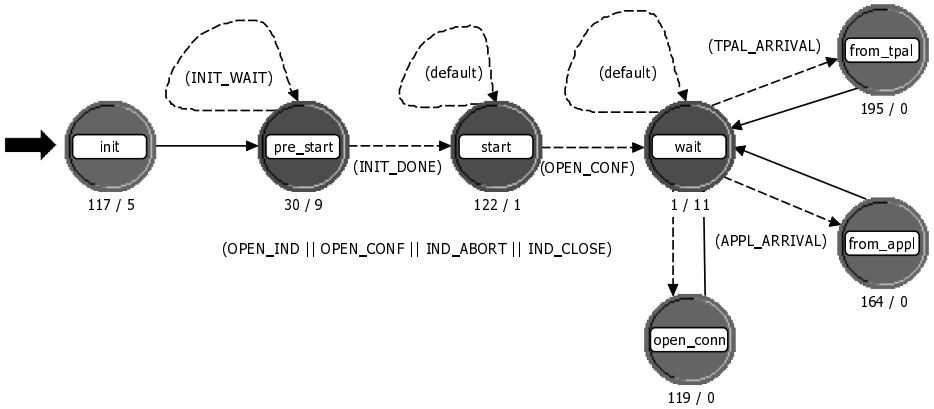

Figure 8.2: The process model that implements the Chameleon protocol. This process model resides within the overlay processor of the wlan server overlay adv node model. The bold arrow indicates the initial state. The empty transitions are illustrated with solid arrows and the conditional transitions with dashed arrows. The conditions associated with the conditional transitions are written in parentheses. The states in this diagram are drawn with two different shades of grey that indicate if a state is a forced state or an unforced state, i.e., if the outgoing transition of a state is an empty condition or not. The numbers located under each state indicate the number of lines of code existing in the entry and the exit executives in each of these states.

The states *from tpal* and *from appl* are used to forward incoming data either to the application processor or to tpal overlay processor. Data arriving from the application processor indicates that this device is the initiating sender γ_s. Data is thus sent to another randomly selected element in the set Γ, according to the state transition diagram presented in Figure 6.5. Data arriving from the tpal processor is either delivered to the application processor, if the current device γ_i is the final destination, that is $\gamma_i \in D_\Gamma$, or sent back to the tpal processor according to the state transition diagram in Figure 6.6.

In the cases where the data is sent to the tpal processor, a traffic engine process model is invoked to deliver the data to the next device $\gamma_{i+1} \in \Gamma$ on the anonymous path. The traffic engine process model is depicted in Figure 8.3. The purpose of this process model is to open a connection and deliver the data to the tpal processor in the *open* state. The tpal processor forwards the data according to the transport protocol defined in the *open* state. In this case, the transport protocol is the UDP connectionless transport protocol. The

tool to link tpal processors from different devices, and do not imply that UDP is modelled as a connection-oriented protocol.

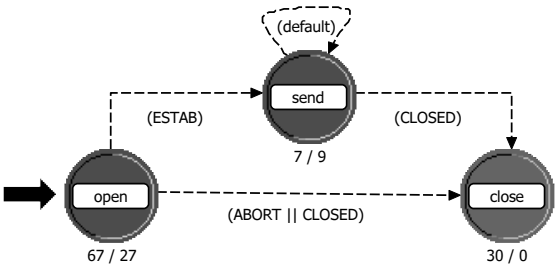

Figure 8.3: The process model that implements the traffic engine is invoked when data needs to be delivered from the overlay processor to the tpal processor. This process model is invoked in the from tpal and from appl states of the process model implemented at the overlay processor, presented in Figure 8.2.

process model is destroyed as soon as the data is sent in the *send* state. The *close* state is reached if the connection to the tpal processor is aborted or closed prematurely, i.e., before the data is successfully delivered to the tpal processor.

Modifications on Other Processors

The implementation of the Chameleon protocol in the Modeler simulation tool required not only the implementation of an overlay processor in between the application and the tpal processors, but also modifications in other processors. These modifications were restricted to the neighboring processors, i.e., application and tpal processors. The process model that implements the traffic engine for the UDP and TCP protocols in the tpal processor were modified to forward incoming data to the overlay processor instead of delivering the data directly to the application processor. The UDP traffic engine process model[2] is depicted in Figure 8.4.

The same modification implemented in the UDP traffic engine was also executed in the traffic engine for the TCP protocol. Although the simulation of the Chameleon protocol used UDP as transport protocol, the tpal processor is also prepared to handle incoming TCP connections and deliver the incoming data to the overlay processor. Thus, the Chameleon protocol is partially prepared to be simulated on top of a connection-oriented transport protocol, as long as the traffic engine of the overlay processor can initiate and manage TCP connections. Moreover, the application processor was modified to include Chameleon as an available service, i.e., application. This modification was required since the incoming and outgoing packets need to be associated with a registered service.

[2]The denomination of this standard process model is *tpal intf udp v3* in OPNET Modeler.

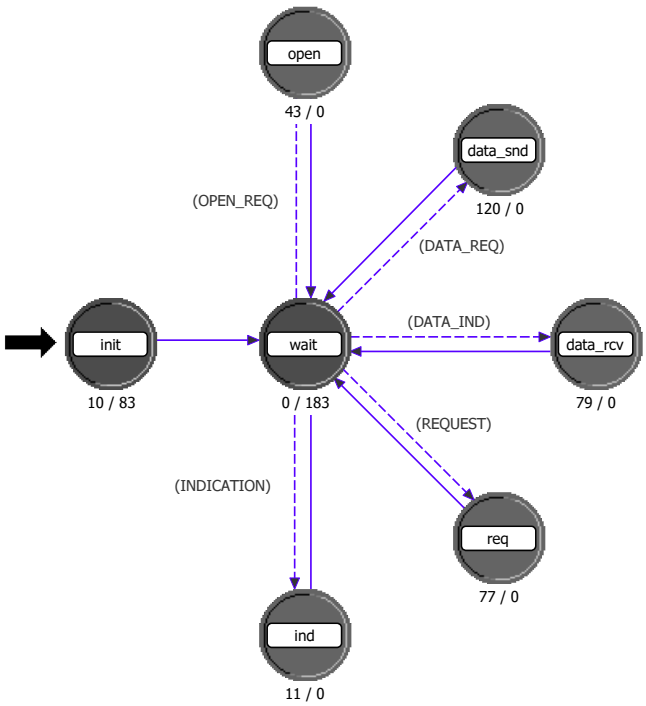

Figure 8.4: The process model that implements the traffic engine for the UDP protocol in the tpal processor. The data rcv state was modified to forward incoming data to the overlay processor instead of delivering the data directly to the application processor.

Simplifications on the Chameleon Framework

Determining the appropriate level of detail in a simulation is a fundamental tasks when simulating a system [Jain, 1991]. Some characteristics of the framework that describes the Chameleon protocol were excluded from the implementation mainly because they do not provide a substantial contribution or influence the accuracy of the results obtained from the simulation runs.

The aspects of the Chameleon framework that were not included in the simulation modelling include the directory servers and the backward communication channel, i.e., the reply data from the recipient back to the initial sender. The simulated network is assumed to be in a steady state, i.e., the participating devices already have knowledge about the other devices running the Chameleon protocol, and the network participants are thus ready to transmit and forward Chameleon packets. The backward communication channel was not

implemented because the objective of the simulation is to obtain the function of the end-to-end delay in terms of the probability of forwarding p_f, and not the round-trip time information.

In addition, the simulation does not implement any encryption mechanisms. The implementation of such items would make the implementation of the Chameleon protocol more complex and would eventually result in a reduced performance of the simulator[3]. Encryption mechanisms are usually not implemented in network simulation scenarios. Instead, encryption is modelled by adding a delay in the simulation time on the point that the data is encrypted or decrypted. There is no such delay included in our simulation model, since such a delay is dependent on the device hardware capabilities.

The list of available devices running the Chameleon protocol in the network is obtained during the initialization phase of the simulation and occur in the *init* state of the process model that implements the Chameleon protocol, presented in Figure 8.2.

8.3 The Simulation Parameters

In this section the simulation parameters used are outlined. The results aggregate the outcome of 3060 distinct simulation runs. All simulation were configured to run for 35 minutes, i.e., 2100 seconds, of simulation time. Five different values for the probability of forwarding p_f were evaluated, resulting in five simulation scenarios. Moreover, an additional basic scenario where the sender device delivers the application data directly to the destination was included in the simulation to set a basis for evaluating the added delay resulting on different values of p_f. The rest of this section is divided into two parts, where each defines different aspects of the parameters.

The first part presents the static input parameters that are common and remain static throughout all simulation runs. Static parameters include the network topology, the wireless technology and the characteristics of the network traffic, for instance. The static parameters are presented in Section 8.3.1.

The second part presents the non-static input simulation parameters, i.e., parameters that change according to the simulation run. There are two non-static parameters in the simulation: the values assigned to the probability of forwarding p_f and the seeds for the pseudo-random number generators used in the simulation. The non-static parameters are presented in Section 8.3.2.

[3]The performance of the simulator, i.e., the amount of time required to run a simulation scenario, is not related to the results obtained from it.

8.3.1 Static Parameters

The network topology used in the simulation is depicted in Figure 8.5. There are 30 static devices distributed in a square area of 210×210 meters with no obstacles. All devices are part of the set Γ of Chameleon users, and, thus, they can eventually be selected to be part of an anonymous path. There is only one sender and one recipient in this scenario. The sender device is labelled source, and the recipient device is labelled server in Figure 8.5. Those two devices are positioned approximately in the center of the topology[4].

All communication is performed using the wireless interfaces. The wireless technology used is the IEEE 802.11 standard [IEEE 802.11]. The maximum data rate is 11 Mbps, the modulation technique is direct sequence spread spectrum, and the transmit power is 5 mW. The maximum radio range for an IEEE 802.11 network interface card in OPNET Modeler using the aforementioned characteristics is approximately 300 meters. All devices are in the same IEEE 802.11 Basic Service Set (BSS).

There is no need for routing data in the network layer since all devices are in the same BSS and, thus, they are at one hop distance. Nevertheless, an on-demand ad hoc routing protocol, the Ad Hoc On-Demand Distance Vector (AODV) protocol [Perkins et al., 2003], is running and it generates a minimum amount of background network traffic[5]. This routing protocol is kept running in this simulation scenario to include some routing data traffic in the ad hoc network, since it is expected that an ad hoc routing mechanism would be available and running in such a scenario.

The sender device injects packets in the network at the constant ratio of one packet per two seconds, and each packet has a data payload of 1032 bytes. Thus, the average amount of traffic sent by the application running in the sender device is 516 bytes/second. This limited amount of data traffic was deliberately chosen to reduce the influence of undesired factors that may potentially interfere with the measurements of this simulation, which were defined in Section 8.1. These undesired factors include discarding packets due to queueing policies in the wireless interface, and interferences between multiple concurrent data flows. Therefore, the simulation conditions are optimal for achieving the simulation goals, i.e., to measure the delay and the number of lost packets in relation to an increasing value of the probability of forwarding p_f in an ad hoc network with devices running the Chameleon protocol. The static parameters used in the simulation are summarized in Table 8.1.

[4]The $\{x, y\}$ coordinates in meters of the sender device are $\{95, 95\}$, and the coordinates of the recipient device are $\{110, 110\}$, assuming that the $\{0, 0\}$ position is located on the top left and the position $\{210, 210\}$ is located on the bottom right of the Figure 8.5.

[5]The interval between consecutive transmissions of AODV hello messages is distributed uniformly in the time interval of $[1.0, 1.1]$ seconds. This result in an average of 370 bps of routing traffic sent per device. Hello messages are control messages specified in the AODV protocol [Perkins et al., 2003].

Figure 8.5: The network topology used in the simulation of the Chameleon protocol. There are 30 static devices distributed in a square area with 210×210 meters. All devices are part of the set Γ of Chameleon users, and there is only one sender and one recipient in this scenario. The sender device is labelled source, and the recipient device is labelled server, and they are located approximately in the center of the topology. The three boxes located on the right hand side of the figure are used in the configuration and specification of the applications running in the simulation scenario. The Application Definition box is used to configure application parameters such as the transport protocol used and port. The Profile Definition is used in the configuration of the profile parameters to be applied to given application, such as the start time, duration, and repeatability of the service. The Task Definition is used for the configuration of the steps performed during the application run.

Table 8.1: Selected static parameters used in the simulation scenario.

Parameter	Value
Network area	210×210 meters
Number of devices	30
Wireless technology	IEEE 802.11
Transmit data rate	516 bytes/second
Packet data payload	1032 bytes
Transmission power	5mW
Maximum data rate	11 Mbps

8.3.2 Non-Static Parameters

As already mentioned, there are two types of non-static parameters used in the simulation of the Chameleon protocol. The different values for the probability of forwarding p_f and the seeds of the pseudo-random number generators that used by Chameleon in the anonymous path establishment process and by Modeler to calculate the environmental variables.

The Probability of Forwarding

The five values for the probability of forwarding p_f used in the simulation are: 0.51, 0.60, 0.67, 0.75, and 0.83. The average expected path lengths L_{exp} associated with those values assigned to the probability of forwarding can be calculated using Equation 6.1. The expected path length L_{exp} for the p_f values are presented in Table 8.2.

Table 8.2: The probability of forwarding values p_f used in the simulation of Chameleon, the expected path length, and the maximum ($|\Gamma'|/|\Gamma|$) fraction of malicious insiders in relation to the total number of elements in the set Γ that are tolerated for these p_f values, according to Equation 7.2.

Probability of forwarding	Expected path length	Tolerated ($\|\Gamma'\|/\|\Gamma\|$) (%)
0.51	3.04	1.96
0.60	3.50	16.67
0.67	4.03	25.37
0.75	5.00	33.33
0.83	6.88	39.76
no forwarding	1.00	—

The chosen values for the probability of forwarding p_f used in the simulation of the Chameleon protocol were selected for the following reasons:

- The p_f value of 0.51 is close to the is minimum value, since p_f is bounded by the open interval $]0.5, 1[$, as presented in Section 6.2. Thus, this 0.51 is a natural minimum bound to be used as a simulation input.
- The p_f value of 0.67 provides an expected path length $L_{exp} > 4$, as presented in Table 8.2, which is a requirement for obtaining the degree of anonymity of probable innocence against a malicious outsider, for both sender and receiver anonymity, as shown in Table 7.1.
- The p_f value of 0.60 was chosen because it is close to the middle point between the p_f values 0.51 and 0.67. The p_f 0.60 provides anonymous paths with an expected path length L_{exp} of 3.5 devices.
- The p_f value of 0.75 was chosen for two different reasons. First it provides a similar step size regarding the previously selected values of probability of forwarding, i.e., the steps between p_f values 0.51, 0.60, and 0.67 has an average step size of 8%. Second, the p_f 0.75 provides anonymous paths with an L_{exp} of 5 devices, which is one device more than the p_f 0.67.
- The upper bound value of p_f, 0.83, was picked for several reasons. The first is that it also provides a regular step size in relation to the previously selected p_f values. Moreover, an evaluation of the relationship between the expected path length and the fraction of malicious insiders tolerated in relation to the total number of participants in the set Γ presented in Section 7.3, was also taken into account for the selection of this upper bound value of p_f.

  The p_f value 0.83 means that degree of sender anonymity against malicious insiders is probable innocence if the fraction $(|\Gamma'|/|\Gamma|)$ of malicious insiders in relation to the cardinality of the set Γ is not more than approximately 0.40. The first derivative of the Equation 7.3, with respect to the fraction $(|\Gamma'|/|\Gamma|)$ provides the rate of change of the expected path length in relation to the $(|\Gamma'|/|\Gamma|)$. The derivative $dL_{exp}/d(|\Gamma'|/|\Gamma|)$ is outlined in Equation 8.1 and the resulting curve plotted in Figure 8.6.

$$\frac{dL_{exp}}{d\left(\frac{|\Gamma'|}{|\Gamma|}\right)} = \frac{2}{\left(1 - 2 \cdot \left(\frac{|\Gamma'|}{|\Gamma|}\right)\right)^2} \quad | \quad |\Gamma| \gg 1,\ 0 \leq \left(\frac{|\Gamma'|}{|\Gamma|}\right) < 0.5,\ \ \forall \left(\frac{|\Gamma'|}{|\Gamma|}\right) \in \mathbb{R} \qquad (8.1)$$

  The analysis of the equation and the resulting plot indicates that the function $dL_{exp}/d(|\Gamma'|/|\Gamma|)$ has a particular characteristic. The rate of change quadruplicates when increasing the fraction $(|\Gamma'|/|\Gamma|)$ in discrete steps that

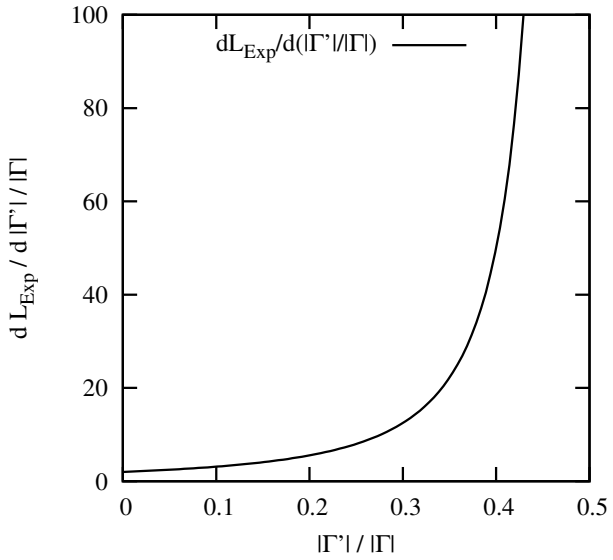

Figure 8.6: This curve plots the derivative of the Equation 8.1, $dL_{exp}/d(|\Gamma'|/|\Gamma|)$. It indicates the rate of change of the expected path length L_{exp} with respect to the fraction $(|\Gamma'|/|\Gamma|)$. This curve shows that the rate of change $dL_{exp}/d(|\Gamma'|/|\Gamma|)$ increases four times if $(|\Gamma'|/|\Gamma|)$ increases from 30% to 40%.

are halved after every step, according to the Equation 8.2, where x_n is the value for the fraction $(|\Gamma'|/|\Gamma|)$ in the n^{th} discrete step.

$$x_n = 0.1 + \sum_{j=1}^{n} \frac{0.2}{2^{j-1}} \mid n \geq 0, \quad \forall\, n \in \mathbb{N} \tag{8.2}$$

The rate of change corresponding to Equation 8.2 is given in Equation 8.3, where y_n is the rate of change for the n^{th} discrete step.

$$y_n = f(n) = 3.125 \cdot 2^{2n} \mid n \geq 0, \quad \forall\, n \in \mathbb{N} \tag{8.3}$$

Equations 8.2 and 8.3 show a series of discrete points in the $(|\Gamma'|/|\Gamma|)$ domain where the rate of change quadruples after every discrete step. The fraction $(|\Gamma'|/|\Gamma|) = 0.40$ is the first of those steps ($n = 2$) that are positioned after the p_f value 0.75. The p_f value that can tolerate up to 40% of malicious insiders in relation to the cardinality of the set Γ and still provides a degree of sender anonymity that equals probable innocence is approximately 0.83, according to Equation 7.2.

The Seeds

Seeds are used by pseudo-random number generators as an initial input to generate a sequence of apparently random values. Using different seed values in a same simulation scenario is a common practice to gather results that can be evaluated statistically, since different seeds generate different environmental parameters. When gathering results from simulation runs, it is possible that some simulation runs generate spurious data. Thus, an adequate amount of simulation runs is necessary to provide statistical soundness to the achieved results.

In the simulation of the Chameleon protocol, every simulation run has associated with it a unique pair of seeds, $\{seed_{modeler}, seed_{chameleon}\}$, which are given by:

- the $seed_{modeler}$ is used by the simulator to compute the random values required by the simulated environment and network modules for the computation of parameters such as background noise and the backoff timers used in IEEE 802.11;
- the $seed_{chameleon}$ is used exclusively by the Chameleon protocol for two tasks during the establishment of an anonymous path. These tasks are to define if an anonymous path should be terminated or not, and, if not, to select the next device in the anonymous path from the set Γ.

For each simulation scenario, i.e., for each different simulated value of p_f, 20 values were used for $seed_{modeler}$, in the interval $[128, 147]$, combined with 30 values for $seed_{chameleon}$, in the interval $[1, 30]$. Therefore, for each different simulated value of p_f, 600 simulation runs were executed. In addition, to the scenario where messages are delivered from the sender to the recipient directly, 60 simulation runs were performed using 60 different values for $seed_{modeler}$, in the interval $[128, 187]$. The summary of the evaluated values for the probability of forwarding p_f and the number of simulation runs performed for each scenario are presented in Table 8.3.

8.4 Simulation Results and Analysis

This section presents the results collected from the simulation of the Chameleon protocol and the statistical analyses of the results. The statistical analyses included in this section are: the average delay for delivering application data from the sender to the recipient, the cumulative distribution functions (CDF) that describe the probability distribution of such a delay, and the amount of undelivered packets in relation to the total amount of packets sent. The expected end-to-end delay and amount of packet loss indicate what type of applications can be deployed on top of the Chameleon protocol.

Table 8.3: Simulation scenarios and number of simulation runs.

Probability of forwarding	Number of simulation runs
0.51	600
0.60	600
0.67	600
0.75	600
0.83	600
No forwarding	60
Total number of runs	3060

The analysis of the influence of an increasing probability of forwarding p_f in the expected average end-to-end delay provides an insight of the performance cost in terms of such a delay for a device that wants to join the anonymous communication network. The same rationale is valid for the amount of undelivered packets. Moreover, the analysis of the scenario with no forwarding is important when defining the minimum delay for the end-to-end delay between two devices in the ad hoc network.

The analysis of the CDF provides a more detailed insight of the distribution of the packet arrivals in relation to different values of the probability of forwarding p_f. Moreover, the evaluation of the CDF obtained from the simulation runs is used to estimate the average delay introduced in the anonymous path on each hop. The average delay obtained in the simulation is used to fine-tune the analytical model of the CDF. Such an analytical model is used for modelling the CDF in terms of the probability of forwarding p_f. The analytical model can be subsequently rewritten in terms of the fraction ($|\Gamma'|/|\Gamma|$) of malicious insiders in relation to the cardinality of the set Γ. The analytical model can be illustrated as a 3-D plot that describes the trade-off between the anonymity properties and performance in terms of the CDF of the expected end-to-end delay.

In all simulation runs, the first 100 seconds of simulation time were discarded to remove transient data. Thus, only the steady state is considered in the analysis presented in this chapter.

Table 8.4 presents the average length of the anonymous path obtained from the simulation runs regarding the different probability of forwarding values associated with them and compares the simulated results to the values for expected path length, which were calculated following Equation 6.1. The table shows that the average path length obtained from the simulation runs is similar to the theoretical average path length. Such a comparison is made to validate the results obtained from the simulation against the analytical modelling

Table 8.4: This table shows that the average path length obtained from the simulation results is similar to the expected average path length obtained from the analytical modelling presented in Chapter 7. Such a comparison is used to validate the results obtained from the simulation. The probability of forwarding associated with these results is also included in this table.

Probability of forwarding	Theoretical path length	Simulated path length
0.51	3.04	3.13
0.60	3.50	3.47
0.67	4.03	4.37
0.75	5.00	5.10
0.83	6.88	6.77
No forwarding	1.00	1.00

in [Reiter and Rubin, 1997]. This procedure follows one of the three rules of validation presented in [Jain, 1991]: "do not trust the results of a simulation model until they have been validated by analytical modelling or measurements".

Table 8.5 presents the average end-to-end delay and the packet loss ratio obtained in the simulation runs regarding the different probability of forwarding values. This tables shows that the increasing the probability of forwarding from 0.51 to 0.83 more than double the average delay and increases the packet loss rate from 1.01 to 43.5 messages for every thousand messages, i.e, an increase of over 430% in the packet loss ratio.

Figure 8.7 depicts the average end-to-end delay for all the simulated probabilities of forwarding p_f in relation to the simulation time in the interval $[100, 2100]$ seconds. Thus, each data point corresponds to the average end-to-

Table 8.5: This table outlines the average end-to-end delay, standard deviation, and the packet loss ratio associated with the probability of forwarding p_f obtained from the simulation runs.

Probability of forwarding	Average delay (ms)	Standard deviation (ms)	Packet Loss (‰)
0.51	4.52	2.54	1.01
0.60	5.35	2.91	1.46
0.67	6.58	4.85	8.30
0.75	7.79	5.55	13.6
0.83	9.90	7.52	43.5
no forwarding	0.98	5.71×10^{-3}	0.07

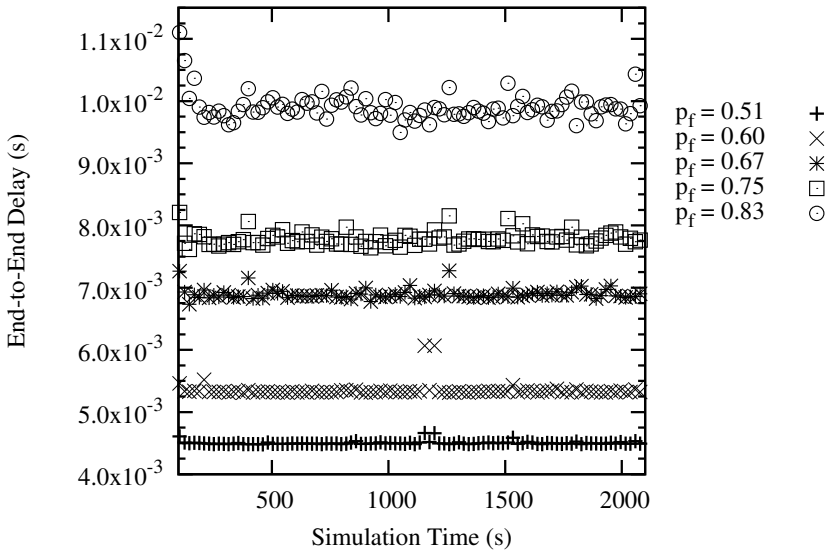

Figure 8.7: The end-to-end delay for different probabilities of forwarding in relation to the simulation time. Each plotted point corresponds to the average value of 600 simulation runs.

end delay value observed in a given instant of the simulation time regarding 600 simulation runs for a given value of p_f. In the simulation, the end-to-end delay is calculated using the time interval between the instant of time when the data packet is created in the application processor of the sender device and the instant that this data packet is received at the *from tpal* state of the overlay processor in the recipient device.

The average end-to-end delay is also presented in Figure 8.8. This figure is comprised of six individual graphs, where each graph represents one of the six simulated scenarios. The scenario is indicated at the top left corner of each graph. These graphs include the confidence intervals for a significance level $\alpha = 0.05$.

Figures 8.9 to 8.13 present the histograms of packet arrivals and the CDF in relation to the time of the packet arrival for the different simulated scenarios running the Chameleon protocol. Each figure corresponds to a different probability of forwarding p_f and is comprised of two graphs:

- the graph at the top of each figure consists of a histogram of the number of packet arrivals in relation to the time of arrival and the CDF associated with this histogram. Each plot contains data obtained from 600 simulation runs;

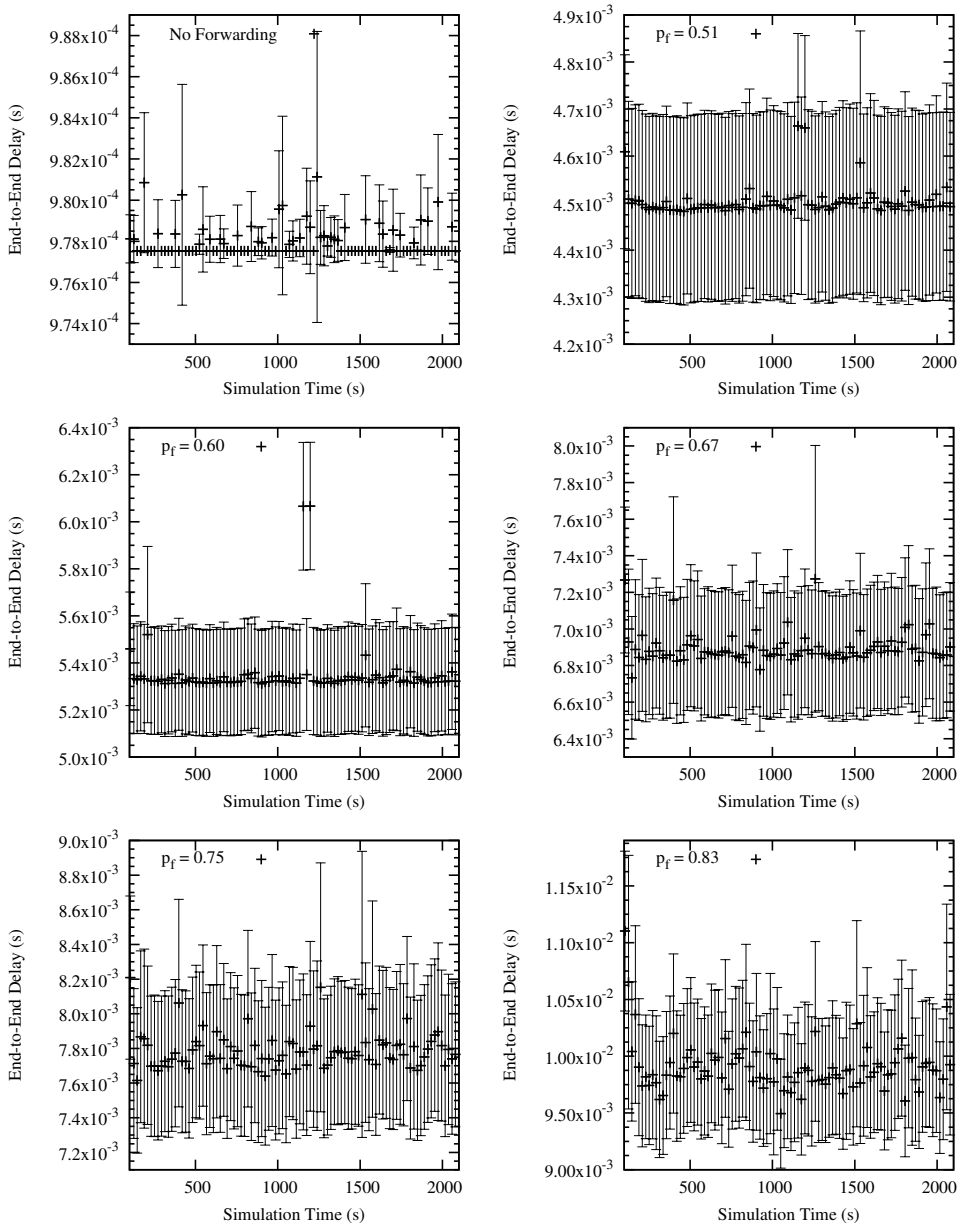

Figure 8.8: The end-to-end delay for the different simulation scenarios, which are indicated in the top right corner of each graph by the p_f value. The vertical bars display the confidence intervals for a significance level $\alpha = 0.05$.

- the graph at the bottom of each figure is comprised of four curves: the CDF curve obtained from the simulation (also included in the top graph) and, in addition, a theoretical CDF that is plotted using two different curve styles: steps and lines. The theoretical CDF curve with steps is used in the comparison with the CDF obtained from the simulation runs, while the theoretical CDF curve with lines is the same curve, but plotted using a continuous function. The theoretical CDF was calculated using Equation 8.5 with the assumption that the average transmission delay in the wireless network is a constant interval. This graph also contains a residual curve, which depicts the difference between the simulated and theoretical CDF curves.

The theoretical CDF $F(t)$ of the Chameleon protocol in relation to the probability of forwarding p_f can be determined by evaluating the cumulative probability associated with the occurrence of a given path length L_n, where n corresponds to the sum of device appearances in the anonymous path. It is assumed that the transmission time between the sender device γ_s and the first device γ_1 in the anonymous path consumes approximately Δt_1 to be delivered and each subsequent transmission in the wireless network takes an average amount of time Δt. Thus, the CDF $F(t)$ can be defined as $F(\Delta t_1 + n \cdot \Delta t)$, where $n \in \mathbb{N}^*$ and Δt_1 is the minimum interval of time required by a device γ_1 to receive a message from a device γ_s. The function $F(\Delta t_1 + n \cdot \Delta t)$ can be written in relation to the probability of forwarding p_f as presented in Equation 8.4:

$$\begin{aligned} F(\Delta t_1 + n \cdot \Delta t) &= (1 - p_f) + p_f \cdot (1 - p_f) + (p_f)^2 \cdot (1 - p_f) + \ldots + (p_f)^{n-1} \cdot (1 - p_f) \\ &= (1 - p_f) \cdot (1 + p_f + (p_f)^2 + \ldots + (p_f)^{n-1}) \\ &= (1 - p_f) \cdot \sum_{i=0}^{n-1} (p_f)^i = (1 - p_f) \cdot \frac{(1 - (p_f)^n)}{(1 - p_f)} = 1 - (p_f)^n \\ &= F(\Delta t_1 + t) = 1 - (p_f)^{\frac{t}{\Delta t}} \end{aligned} \tag{8.4}$$

In Figures 8.9 to 8.13 the values for the parameters Δt_1 and Δt in Equation 8.4 can be extrapolated from the simulation results presented in Tables 8.4 and 8.5. In Table 8.5, the average delay for transmitting application data from the sender directly to the recipient is approximately one millisecond. Thus, we can reckon that $\Delta t_1 = 1 \times 10^{-3}$ second.

Assuming that Δt_1 is known, it is possible to estimate a common value for Δt that can be applied in all simulation scenarios. This value can be estimated using the information from Tables 8.4 and 8.5 as follows. First, by subtracting Δt_1 from the average delay time values presented in Table 8.5, it is possible to estimate the average time interval required to send a packet from the overlay processor of the first device in the anonymous path length γ_1 to the overlay

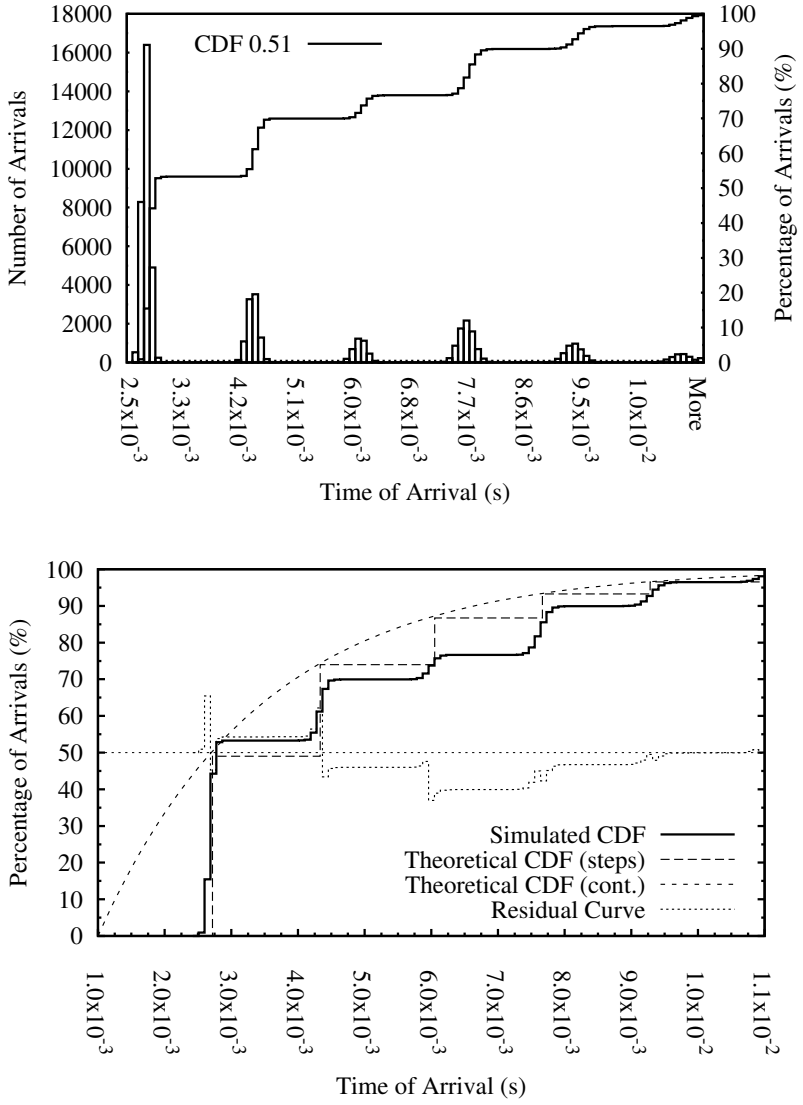

Figure 8.9: Histogram of packet arrivals in relation to the time of arrival and the CDF curves associated with this histogram. The p_f value is 0.51.

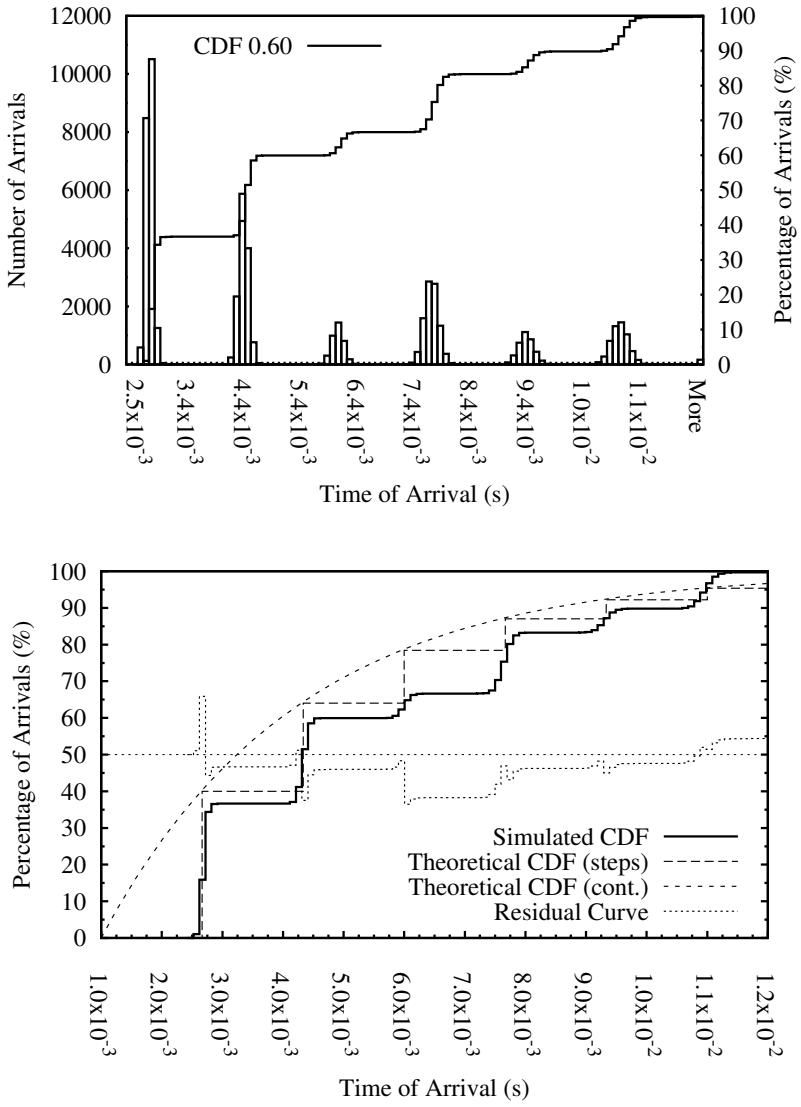

Figure 8.10: Histogram of packet arrivals in relation to the time of arrival and the CDF curves associated with this histogram. The p_f value is 0.60.

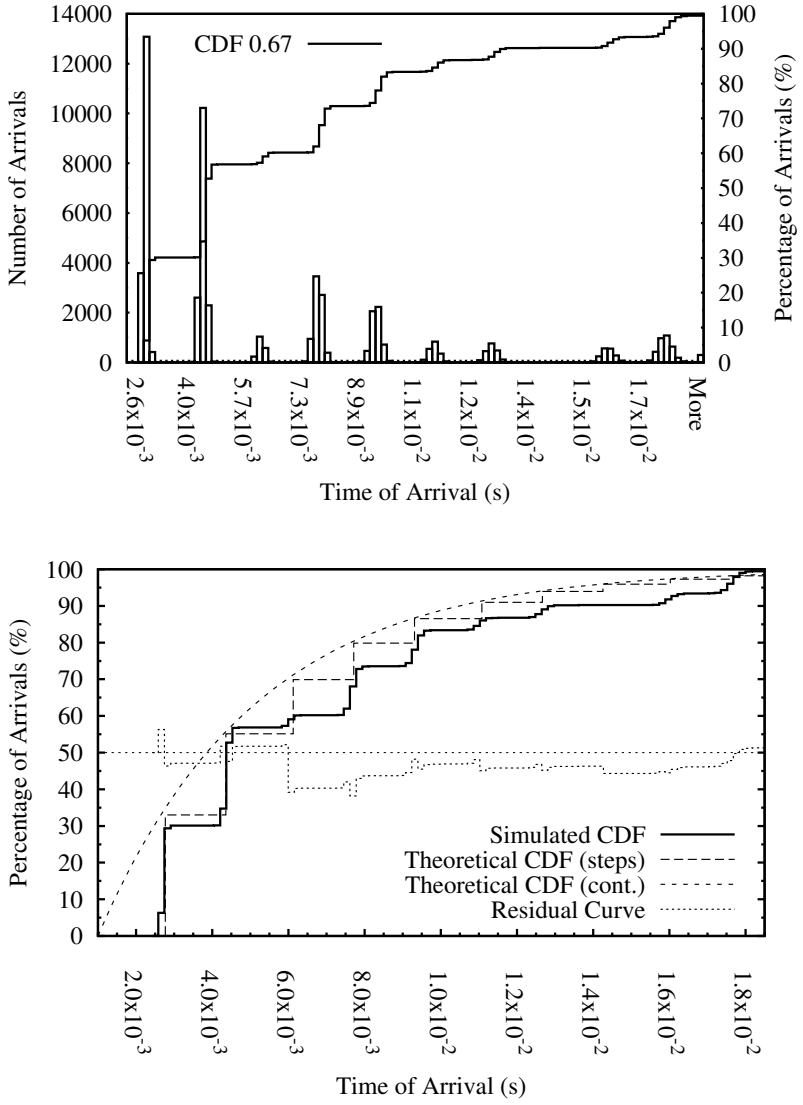

Figure 8.11: Histogram of packet arrivals in relation to the time of arrival and the CDF curves associated with this histogram. The p_f value is 0.67.

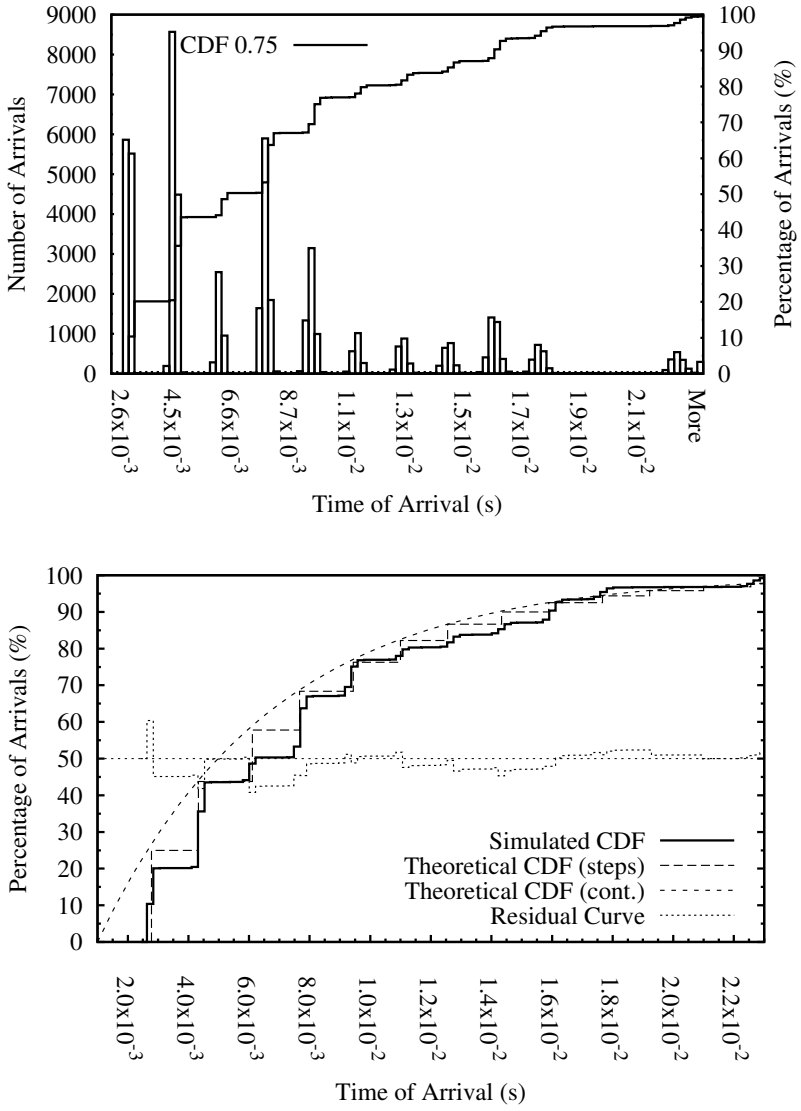

Figure 8.12: Histogram of packet arrivals in relation to the time of arrival and the CDF curves associated with this histogram. The p_f value is 0.75.

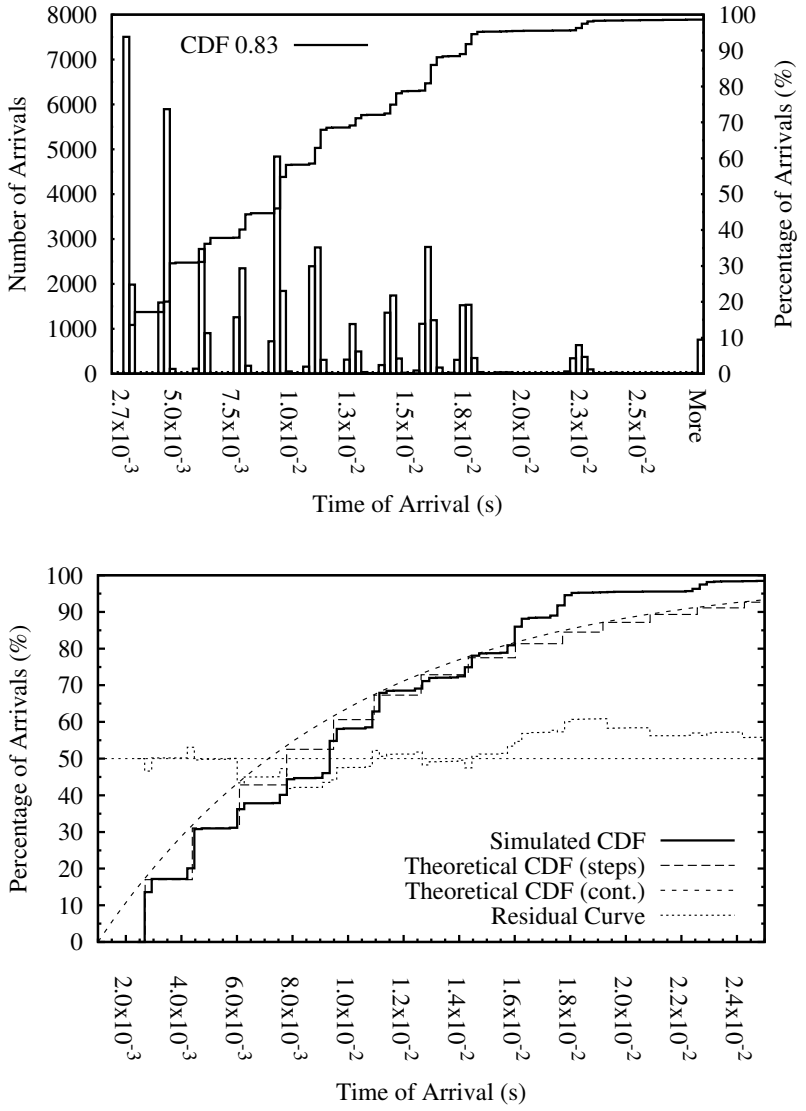

Figure 8.13: Histogram of packet arrivals in relation to the time of arrival and the CDF curves associated with this histogram. The p_f value is 0.83.

processor of the recipient device. The obtained values can then be divided by the simulated path length, from Table 8.4, subtracted by one[6]. Thus, we can estimate that $\Delta t \approx 1.65 \times 10^{-3}$ seconds.

Therefore, the theoretical CDF depicted in the bottom graphs in Figures 8.9 to 8.13 are plotted using $\Delta t = 1.65 \times 10^{-3}$ seconds and $\Delta t_1 = 1 \times 10^{-3}$ second, as shown in Equation 8.5.

$$F(t + 1.0 \times 10^{-3}) = 1 - (p_f)^{\frac{t}{1.65 \times 10^{-3}}} \tag{8.5}$$

The total amount of packets depicted in the histograms presented in Figures 8.9 to 8.13 are in between 55200 and 57000 that were obtained from the 600 simulation runs for each value of the probability of forwarding p_f. Each packet in the histogram corresponds to the sample mean of the end-to-end delay of a bucket of packets. Each bucket has one or more data packets that are collected in a given time interval during the simulation run. The use of packet buckets is useful to increase the performance of the simulation tool. This parameter is specified in the statistics collection mechanism of Modeler.

The residual curves plotted in Figures 8.9 to 8.13, i.e., the differences between the CDF curves obtained through the simulation runs and the theoretical CDF, indicate that there is a good fit between these two CDF, since the residual curve fluctuates around the reference line horizontally positioned in the center of the graph. Thus, the assumption that the average delay Δt between any two devices in the wireless network is constant is valid for the simulated scenarios. Moreover, the values for $\Delta t = 1.65 \times 10^{-3}$ seconds and $\Delta t_1 = 1 \times 10^{-3}$ second, which were estimated from the analysis of the Tables 8.4 and 8.5, proved to be well selected for the constant parameters in the theoretical CDF presented in Equation 8.4, as shown in Figures 8.9 to 8.13, since the first step and the step width matches in both simulated and theoretical CDF curves. This fact is also noticeable by analyzing the residual curve and verifying that there are few spikes in the residual curve. Such spikes are likely to occur when plotting the difference between two step curves that are shifted in time, i.e., not closely aligned.

Equation 8.4 can be rewritten in terms of the fraction ($|\Gamma'|/|\Gamma|$) of malicious insiders in relation to the cardinality of the set Γ instead of in terms of the probability of forwarding p_f. Equation 8.6 is a CDF that describes the trade-off between anonymity and performance in a wireless network scenario running the Chameleon protocol. Such an equation is obtained by substituting p_f for the inequality presented in Equation 7.2. The fraction ($|\Gamma'|/|\Gamma|$) is replaced by the symbol Ω in Equation 8.6.

$$F(t, \Omega) = 1 - \left(\frac{1}{2 \cdot (1 - \Omega)} \right)^{\frac{t - \Delta t_1}{\Delta t}} \quad | \; 0 \leq \Omega < 0.5, \; \forall \, \Omega \in \mathbb{R} \tag{8.6}$$

[6]This subtraction by one refers to the first hop from the sender device to the first device in the anonymous γ_1, which takes an expected time interval Δt_1.

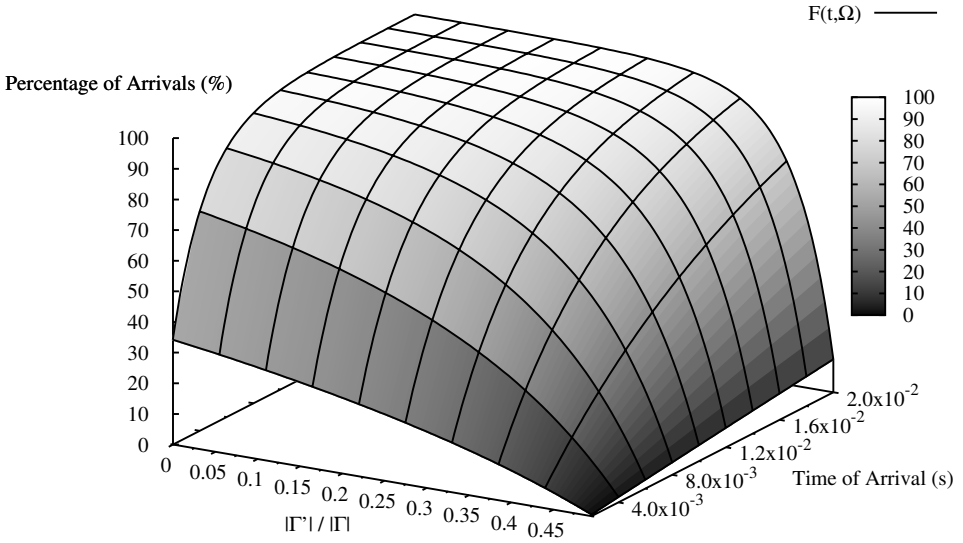

Figure 8.14: The CDF curve $F(t,\Omega)$ in terms of the fraction Ω, i.e., $(|\Gamma'|/|\Gamma|)$, of malicious insiders in relation to the cardinality of the set Γ. This curve outlines the trade-off between network performance, in terms of the expected distribution of the end-to-end delay, and anonymity, in terms of the relation of the fraction of malicious insiders Γ' in the set Γ, respectively. The gray-scale on the right of the figure indicates the percentage of arrivals and it is equivalent to the z-axis.

Equation 8.6 describes the lower bound of the CDF representing packet arrivals in terms of the fraction $(|\Gamma'|/|\Gamma|)$ that provides a degree of sender anonymity of *probable innocence* against malicious insiders. Substituting the equation parameters Δt_1 and Δt for the estimated values $\Delta t = 1.65 \times 10^{-3}$ seconds and $\Delta t_1 = 1 \times 10^{-3}$ second results in Equation 8.7. This equation describes the trade-off between network performance and anonymity, in terms of the CDF of the end-to-end delay and the fraction of malicious insiders in the anonymity set.

$$F(t,\Omega) = 1 - \left(\frac{1}{2\cdot(1-\Omega)}\right)^{\frac{t-1\times10^{-3}}{1.65\times10^{-3}}} \quad | \, 0 \leq \Omega < 0.5, \; \forall\, \Omega \in \mathbb{R} \tag{8.7}$$

Equation 8.7 is depicted in a 3-dimensional plot presented in Figure 8.14, where the x-axis is the fraction Ω, i.e., $(|\Gamma'|/|\Gamma|)$, of malicious insiders in relation to the cardinality of the set Γ. The y-axis is the expected time of arrival in seconds, i.e., the end-to-end delay, and the z-axis is the cumulative percentage of message arrivals.

Figure 8.14 can be analyzed from the perspective of the expected CDF for a given resistance against malicious insiders. It is thus possible to select a given fraction of malicious insiders in relation to the set Γ and obtain the CDF of the end-to-end delay in the anonymous communication network running the Chameleon protocol. Likewise, the figure can be read from the perspective of the time of arrival, i.e., the end-to-end delay, to obtain the percentile of packets that is expected to be delivered until the selected time. Such information is valuable to determine the performance cost of having applications running on top of the Chameleon protocol for different levels of resistance against malicious insiders, given by the fraction ($|\Gamma'|/|\Gamma|$), in an ad hoc network environment.

Applications with minimum requirements for the quality of service parameters, such as audio or video streaming applications deployed in ad hoc networks, can benefit from the information obtained from anonymity and performance trade-offs presented in Equation 8.7 and from the packet loss ratio presented in Table 8.5, especially if anonymity is perceived as a quality of service parameter. Such applications can thus predict the performance impact of the Chameleon protocol.

8.5 Summary

This chapter presented the trade-off between anonymity, in terms of the resistance against malicious insiders, and performance, in terms of end-to-end delay, for the Chameleon protocol. Such an evaluation was performed through the simulation of the Chameleon protocol using the OPNET Modeler network simulation tool. Six simulation scenarios were evaluated in 3060 simulation runs. For each of the simulated scenarios, the average end-to-end delays and the cumulative distribution functions were plotted using the results acquired in the simulation runs. The analysis of the simulation results were used to define a cumulative distribution function of message arrivals in relation to the probability of forwarding, and, moreover, in relation to the fraction ($|\Gamma'|/|\Gamma|$) of malicious insiders in relation to the cardinality of the set Γ.

In the following chapter we summarize the results and the achieved contributions of this dissertation. Moreover, the next chapter presents the concluding remarks of this dissertation.

Chapter 9

Final Remarks

> "And so, if I understand you correctly, you act, and you know why you act, but you don't know why you know that you know what you do?"
>
> *Adso of Melk*
> *— The Name of the Rose (1980), Umberto Eco*

In this chapter we review the objectives and evaluate the contributions of this dissertation. Moreover, we point future directions for the presented research. This chapter, and the dissertation, closes with the concluding remarks.

9.1 Reviewing the Achievements

The overall goal of this dissertation is to offer better anonymous communication in ad hoc network environments. As presented in Chapter 1, such an objective was divided into three goals. The first goal is to establish the connection between the need of trusted identifiers and the provisioning of anonymity. The second goal is to design and evaluate privacy-friendly identifiers that are suitable for ad hoc network environments. The third goal is to design and evaluate an anonymous communication mechanism for such environments. These three goals were then formulated into two research questions. The first research question is:

I. *How to design proper and trusted privacy-friendly digital identifiers to be used in ad hoc network environments?*

Addressing such a research question required to identify the relation between proper and trusted identifiers with the provisioning of anonymity in ad hoc networks. We pointed out that proper identification is a keystone factor to

construct the anonymity sets free of Sybil identifiers. The relationship between the need of trusted identifiers and the provisioning of anonymity in ad hoc networks was called the "identity-anonymity paradox". The identity-anonymity paradox is the first contribution of the dissertation introduced in Chapter 3.

The identification of the identity-anonymity paradox was followed by the definition of the requirements for privacy-friendly identifiers in Chapter 4, and, finally, by the proposal and analysis of the self-certified Sybil-free pseudonyms in Chapter 5, which address the aforementioned requirements. The self-certified Sybil-free pseudonyms are produced within a framework for the provisioning of identifiers that are bound to a group and are Sybil-free and self-certified, i.e., they are issued by the device that holds it and supports the detection of devices that issue more than one identifier in a given group. Such pseudonyms provide unlinkability between different identifiers issued to different groups by the same device and can be produced without the assistance of any trusted third party, but with a restriction of producing one pseudonym per application context at most.

The proposal of the self-certified Sybil-free pseudonyms are the second major contribution presented in the dissertation, and they provide an answer to the first research question, which led us to the second research question:

II. *How to provide anonymous communication in ad hoc networks and what is the performance cost in relation to the obtained degree of anonymity?*

The second research problem refers to the design of an anonymous communication protocol suitable for ad hoc network environments. The research question was addressed in different steps. First, the requirements for anonymous communication mechanisms in ad hoc networks were identified and listed. Such requirements are outlined in Chapter 4. The definition of requirements led us to the design of an overlay anonymous communication mechanism for ad hoc network environments, which we called Chameleon. Chameleon is a low-latency overlay mechanism that operates in between the application and the transport layer. The objective of Chameleon is to provide anonymous communication in ad hoc networks. Chameleon was described in Chapter 6. Chapter 7 presented the analysis of the anonymity properties of Chameleon, which were evaluated using analytical methods. The network performance properties were analyzed using simulation and validated through analytical methods. Moreover, we identified the trade-off between anonymity and performance of Chameleon in an ad hoc network environment. Such a trade-off is described by the cumulative distribution function of the expected end-to-end delay in relation to the projected resistance against malicious users. The details of the simulation environment and the performance analysis of Chameleon were presented in Chapter 8. The proposal of Chameleon and the identification of the trade-off between anonymity and performance are the third major contribution of the dissertation, and they provide an answer to the second research question.

Other contributions of this dissertation are the summary of sources of device identification presented in Chapter 2 and the definition of the requirements for anonymous communication mechanisms and privacy-friendly identifiers in Chapter 4.

9.2 Future Directions

A proof-of-concept implementation of the self-certified Sybil-free pseudonyms is an interesting future step. The implementation of such pseudonyms would allow us to evaluate the performance of such identifiers in different platforms and also to analyze the usability of the privacy-friendly identifiers.

An important contribution of this dissertation was the implementation of the Chameleon protocol in a network simulator environment. Such an implementation allows us to evaluate the network performance parameters under different situations and network topologies. Thus, the effect of other variables in the network performance could be analyzed by changing the simulation parameters used, without the need of changing the implementation code. Moreover, Chameleon is an overlay protocol situated in between the application and the transport layers, i.e., the application and the tpal overlay processors, as shown in Figure 8.1. The overlay processor that contains the implementation of the Chameleon protocol is implemented in the standard wireless server node model from Modeler. Such a processor can be seen as a black box that is interconnected and interact with the other processors surrounding it. Thus, it is possible to replace the implementation code of the Chameleon protocol with code from another anonymous communication mechanism without having to modify the connections of the overlay processor. Such a procedure would permit other protocols to be evaluated using the same simulation parameters used in the evaluation of Chameleon.

The implementation, deployment, and evaluation of a prototype of the Chameleon protocol in an ad hoc network platform for experimentation, i.e., a testbed, is another possible future step in the validation of the results obtained through simulation and analytical modelling and presented in Chapter 8.

9.3 Concluding Remarks

To conclude, we proposed identifiers that are used to build anonymity sets that allow the detection of Sybil identifiers and provide unlinkability between multiple pseudonyms that belong to a given user. Moreover, we designed the Chameleon protocol, which is an anonymous communication mechanism for ad hoc networks. The combination of the self-certified Sybil-free pseudonyms and the Chameleon protocol gives users in an ad hoc network the option to be anony-

mous and, thus, to determine if the information regarding their communicating partners, or even their presence in the ad hoc network, is communicated to others or not. Furthermore, we evaluated the Chameleon protocol using a network simulator and identified a trade-off between anonymity and performance in an ad hoc network. Applications with minimum requirements for the quality of service parameters, such as audio or video streaming applications deployed in ad hoc networks, can benefit from the information obtained from such a trade-off.

It is important to remark there are leaks of personal data from sources not covered by the solutions presented above, such as identifiable information leaks the physical, data link, and application layers, as seen in Chapter 2. Nevertheless, the solutions proposed in this dissertation are important steps towards the achievement of better anonymous communications in ad hoc network environments that can complement other mechanisms that prevent leaks of personal data from sources not covered by the research presented here.

Appendix A

The Cryptographic Foundation

> "... Often books speak of other books. Often a harmless book is like a seed that will blossom into a dangerous book, or it is the other way around: it is the sweet fruit of a bitter stem. In reading Albert, couldn't I learn what Thomas might have said? Or in reading Thomas, know what Averroës said?"
>
> *Brother William of Baskerville*
> *— The Name of the Rose (1980), Umberto Eco*

Different cryptographic systems can be used to create unlinkable and unique pseudonyms. As long as the identification of "double-spent" pseudonyms is not an issue, such pseudonyms can be realized based on the so-called epoch number of direct anonymous attestation [Brickell et al., 2004]. Schemes that support identification were presented in [Camenisch et al., 2006] and [Damgård et al., 2006]. By binding a different tag to every identity domain, k-times anonymous authentication [Teranishi et al., 2004] can be used to create unique pseudonyms.

The scheme presented in this dissertation uses the cryptographic techniques proposed by Camenisch et al. [2006] (i.e., e-tokens), but can be seen as a more general systems framework that could also be instantiated using other cryptographic techniques. This appendix was originally published in [Andersson et al., 2008a] and some parts of it also appeared in [Martucci et al., 2008a].

This appendix is divided into five sections. Section A.1 presents a more detailed explanation regarding the cryptographic algorithms introduced in Chapter 5. The properties of unlinkability and identification of Sybil identifiers in a

given identity domain are surveyed in Section A.2. The cryptographic building blocks used in the realization of the cryptographic algorithms are presented in Section A.3 and some of the cryptographic primitives used are outlined in Section A.4. Finally, Section A.5 presents a brief discussion regarding the efficiency of the proposed solution.

A.1 The Cryptographic Algorithms

Briefly, k-spendable e-tokens can be realized as follows. Both the issuer and the device that is requesting a membership certificate, generate key pairs. Let the device's key pair be (pk_a, sk_a), where $pk_a = g^{sk_a}$ and g generates a group $\mathbb{G}$ of known order. The issuer's key pair is used for creating and verifying Camenisch and Lysyanskaya (CL) signatures [Camenisch and Lysyanskaya, 2002]. We use a pseudo-random function f_s whose range is the group $\mathbb{G}$.

By using the *Obtain* algorithm, the device interacts with the issuer running the *Issue* algorithm and obtains an e-token dispenser $\mathbb{D}$ that allows it to show one e-tokens per identity domain identifier. The dispenser $\mathbb{D}$ is comprised of a seed s for the pseudo-random function f_s, the device's secret key sk_U, and the issuer's CL signature on (s, sk_a). CL signatures are used to prevent the issuer from learning anything about s or sk_a. Moreover, the dispenser $\mathbb{D}$ is revoked by revoking the corresponding CL signature.

In the *Sign* algorithm, a device shows its token for an identity domain z. It releases a serial number $S = f_s(0||z)$, a double-show tag $E = pk_a \cdot f_s(1||z)^{h(m)}$, and using the Fiat-Shamir heuristic [Fiat and Shamir, 1987] it creates a non-interactive zero-knowledge proof σ that (S, E) corresponds to a valid dispenser for the identity domain z, i.e., the device proves in zero-knowledge that S and E were properly formed from values (s, sk_a) signed by the issuer. To sign message m, m is hashed into the challenge together with the first message and the public parameters of the proof. The transcript τ contains both E and σ. An e-token is verified by checking the non-interactive proof.

A.2 Unlinkability and Identification

As f_s is a pseudo-random function, and all proof protocols are zero-knowledge, it is computationally infeasible to link the resulting e-token to the device, the dispenser $\mathbb{D}$, or any other e-tokens corresponding to $\mathbb{D}$. If a device shows two e-tokens in the same identity domain to authenticate two messages m and m', then both e-tokens *must* use the same serial number.

The issuer, or any other participating device, can easily detect the violation and compute pk_a from the two double-show tags:

$$E = pk_a \cdot f_s(1||z)^{h(m)} \text{ and } E' = pk_a \cdot f_s(1||z)^{h(m')} \tag{A.1}$$

Thus, from the aforementioned Equations A.1, we have:

$$f_s(1||z) = \left(\frac{E}{E'}\right)^{(h(m)-h(m'))^{-1}} \text{ and } pk_a = \frac{E}{f_s(1||z)^{h(m)}} = \frac{E'}{f_s(1||z)^{h(m')}} \tag{A.2}$$

For a more detailed security analysis see Camenisch et al. [2006].

A.3 The Cryptographic Building Blocks

In this section, we summarize the necessary information about the underlying cryptographic building blocks of the self-certified Sybil-free pseudonyms.

A.3.1 Zero-Knowledge Proofs of Knowledge

A zero-knowledge proof is an interactive proof in which the verifier learns nothing besides the fact that the statement that is proven is true. This notion is defined by means of a simulator, which can reproduce the communication knowing only what the verifier knows. A proof of knowledge is an interactive proof in which the prover succeeds in convincing a verifier that it knows something. What it means for a machine to know something is defined in terms of computation. A machine knows something, if this something can be computed, given the machine as an input. The machine extracting the knowledge is called the knowledge extractor. Protocols with a simulator and a knowledge extractor are called zero-knowledge proofs of knowledge.

A.3.2 Sigma protocols and the Fiat-Shamir heuristic

For some protocols only simulators that work for honest verifiers are known. These are verifiers that choose the challenge according to a predetermined distribution. Honest verifier zero-knowledge proofs-of-knowledge protocols that have a three move structure—commitment, challenge and response—are called sigma protocols.

Sigma protocols exist for proving knowledge of discrete logarithm (DL), equality of DLs, and linear relations between DLs in groups of known [Brands, 1997; Camenisch and Stadler, 1997; Schnorr, 1991], and hidden order [Bangerter et al., 2005; Kunz-Jacques et al., 2006]. This allows us to prove statements about certain algorithms that operate in these groups, for instance that two commitments contain the same value or that a committed value lies in a certain interval [Boudot, 2000], that we know a signature for a value or a committed value, that a value was verifiable encrypted, or that a value was correctly created using a pseudo-random function and a secret seed.

Such protocols can be made non-interactive by applying a cryptographic trick called Fiat-Shamir heuristic [Fiat and Shamir, 1987]. This heuristic uses

a cryptographic hash function to allow the prover to compute the challenge herself without involving the verifier. Non-interactive proofs of knowledge have the advantage that they do not require interaction between the prover and the verifier. In addition, they allow to sign any message by hashing it together with the first message when creating the challenge.

A.4 The Cryptographic Primitives

In this section, we briefly introduce some of the cryptographic primitives used in the construction of the self-certified Sybil-free pseudonyms. The cryptographic primitives presented in this section are the same as the periodic n-times e-tokens presented in [Camenisch et al., 2006], since the Sybil-free self-certified pseudonyms are a cryptographic variant of such e-tokens.

A.4.1 Dodis and Yampolskiy Pseudo-random Function

Let $\mathbb{G} = \langle g \rangle$ be a group of prime order $q \in \Theta(2^k)$. Let a be a random element of $\mathbb{Z}_q^*$. Dodis and Yampolskiy [Dodis and Yampolskiy, 2005] showed that $f_{g,a}^{DY}(x) = g^{\frac{1}{(a+x)}}$ is a pseudo-random function, under the decisional Diffie-Hellman inversion assumption (y-DDHI), when either:

- the inputs are drawn from the restricted domain $\{0,1\}^{O(\log k)}$ only, or;
- the adversary specifies a polynomial-sized set of inputs from $\mathbb{Z}_q^*$ *before* a function is selected from the pseudo-random function family (i.e., before the value a is selected).

We require that the DY construction work for inputs drawn arbitrarily and adaptively from $\mathbb{Z}_q^*$. The Dodis-Yampolskiy pseudo-random function is adaptively secure for inputs in $\mathbb{Z}_q^*$ under the secure decisional Diffie-Hellman inversion assumption (SDDHI) [Camenisch et al., 2006].

A.4.2 Pedersen and Fujisaki-Okamoto Commitments

In the Pedersen commitment scheme [Pedersen, 1992] the public parameters are a group $\mathbb{G}$ of prime order q and generators $(g_0, \ldots, g_m)$. To commit to the values $(v_1, \ldots, v_m) \in {\mathbb{Z}_q}^m$, pick a random $r \in \mathbb{Z}_q$ and set $C = PedCom(v_1, \ldots, v_m; r) = g_0^r \prod_{i=1}^m g_i^{v_i}$. Fujisaki and Okamoto [Fujisaki and Okamoto, 1997] showed how to expand this scheme to composite order groups.

A.4.3 Camenisch and Lysyanskaya Signatures

The Camenisch and Lysyanskaya (CL) signature scheme [Camenisch and Lysyanskaya, 2002] includes two protocols:

- an efficient protocol for a user to obtain a signature on the value in a Pedersen (or Fujisaki-Okamoto) commitment [Fujisaki and Okamoto, 1997; Pedersen, 1992] without the signer learning anything about the message and;
- an efficient proof of knowledge of a signature protocol.

The security is based on the strong RSA assumption. By using bilinear maps, we can use other signature schemes [Camenisch and Lysyanskaya, 2004] for shorter signatures.

A.5 Efficiency

The overall costs of our system are linear in the size of the identity domain with respect to users joining the domain, and quadratic with respect to the verification of the Sybil-free property:

- users needs to execute the *Sign* algorithm for themselves, and;
- users need to execute the *Verify* algorithm for all other users.

The construction in [Camenisch et al., 2006] requires 10 multi-base exponentiations for pseudonym certificate creation and a similar number of multi-exponentiations for verification. Using multi-base exponentiation tricks, multi-base exponentiations can be made almost as efficient as normal exponentiations. This compares to schemes that do not support identification with about half the number of multi-exponentiations, and ordinary pseudonym certificates issued by a trusted third party with one or two exponentiations. Verification may not be needed in all cases, e.g., if users trust the domain controller to verify users on their behalf, or if the application bases its security properties on the assumption that only a set of key users are not Sybil identifiers, rather than every single user.

References

Imad Aad, Jean-Pierre Hubaux, and Edward W. Knightly. Denial of service resilience in ad hoc networks. In Zygmunt J. Haas, Samir R. Das, and Ravi Jain, editors, *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM 2004)*, pages 202–215, New York, NY, USA, 26 Sep – 1 Oct 2004. ACM Press. ISBN 1-58113-868-7. doi: http://doi.acm.org/10.1145/1023720.1023741.

Alfarez Abdul-Rahman and Stephen Hailes. A distributed trust model. In *Proceedings of the 1997 Workshop on New Security Paradigms (NSPW 1997)*, pages 48–60, New York, NY, USA, 23–26 Sep 1997. ACM. ISBN 0-89791-986-6. doi: http://doi.acm.org/10.1145/283699.283739.

Patrick Albers, Olivier Camp, Jean-Marc Percher, Bernard Jouga, Ludovic Mé, and Ricardo Staciarini Puttini. Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. In Qusay H. Mahmoud, editor, *Proceedings of the 1st International Workshop on Wireless Information Systems (WIS 2002), in conjunction with ICEIS 2002*, pages 1–12. ICEIS Press, 2002. ISBN 972-98816-0-X.

Christer Andersson. *Design and Evaluation of Anonymity Solutions for Mobile Networks*. PhD thesis, Karlstad University, Jan 2008.

Christer Andersson, Jan Camenisch, Stephen Crane, Simone Fischer-Hübner, Ronald Leenes, Siani Pearson, John Sören Pettersson, and Dieter Sommer. Trust in PRIME. In *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2005)*, pages 552–559, 18–21 Dec 2005a.

Christer Andersson, Markulf Kohlweiss, Leonardo A. Martucci, and Andryi Panchenko. A Self-Certified and Sybil-Free Framework for Secure Digital Identity Domain Buildup. In Jose A. Onieva, Damien Sauveron, Serge Chaumette, Dieter Gollmann, and Konstantinos Markantonakis, editors, *Information Security Theory and Practices: Smart Devices, Convergence and Next Generation Networks, Proceedings of the 2nd IFIP WG 11.2 International*

Workshop (WISTP 2008), Lecture Notes in Computer Science, LNCS 5019, pages 64–77. Springer, 13–16 May 2008a. ISBN 978-3-540-79965-8.

Christer Andersson, Reine Lundin, and Simone Fischer-Hübner. Privacy Enhanced WAP Browsing with mCrowds: Anonymity Properties and Performance Evaluation of the mCrowds System. In Hein Venter, Jan Eloff, Les Labuschagne, and Mariki Elof, editors, *Proceedings of the ISSA 2004 Enabling Tomorrow Conference*, 30 Jun – 2 Jul 2004.

Christer Andersson, Leonardo A. Martucci, and Simone Fischer-Hübner. Requirements for Privacy-Enhancements for Mobile Ad Hoc Networks. In Armin B. Cremers, Rainer Manthey, Peter Martini, and Volker Steinhage, editors, 3^{rd} *German Workshop on Ad Hoc Networks (WMAN 2005), Proceedings of INFORMATIK 2005 – Informatik LIVE! Band 2*, volume 68 of *LNI*, pages 344–348. GI, 19–22 Sep 2005b. ISBN 3-88579-397-0.

Christer Andersson, Leonardo A. Martucci, and Simone Fischer-Hübner. Privacy & anonymity in mobile ad hoc networks. In Zhang et al. [2008], chapter 27, pages 431–448. ISBN 978-1599048994.

William A. Arbaugh, Narendar Shankar, Yung-Chun Justin Wan, and Kan Zhang. Your 802.11 wireless network has no clothes. *IEEE Communications Magazine*, 6(9):44–51, Dec 2002.

Patroklos G. Argyroudis and Donal O'Mahony. Secure routing for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 7(3):2–21, Third Quarter 2005. ISSN 1553-877X.

Nadarajah Asokan, Valtteri Niemi, and Kaisa Nyberg. Man-in-the-Middle in Tunneled Authentication Protocols. Technical Report 2002/163, IACR ePrint Archive, Oct 2002. See http://eprint.iacr.org/2002/163/.

Tuomas Aura. Cryptographically Generated Addresses (cga). RFC 3972, Mar 2005. See http://www.ietf.org/rfc/rfc3972.txt.

Tuomas Aura, Janne Lindqvist, Michael Roe, and Anish Mohammed. Chattering laptops. In Nikita Borisov and Ian Goldberg, editors, *Proceedings of the 8^{th} International Symposium on Privacy Enhancing Technologies (PETS 2008)*, volume 5134 of *Lecture Notes in Computer Science*, pages 167–186. Springer, 23–25 Jul 2008. ISBN 978-3-540-70629-8.

Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. An on-demand secure routing protocol resilient to byzantine failures. In WiSE 2002 WiSE 2002, pages 21–30. ISBN 1-58113-585-8. doi: http://doi.acm.org/10.1145/570681.570684.

Baruch Awerbuch and Christian Scheideler. Group spreading: A protocol for provably secure distributed name service. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, volume 3142 of *Lecture Notes in Computer Science*, pages 183–195. Springer, 12–16 Jul 2004. ISBN 3-540-22849-7.

Paramvir Bahl and Venkata N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Communication Society (INFOCOM 2000)*, volume 2, pages 775–784, Tel Aviv, Israel, 26–30 Mar 2000. ISBN 0-7803-5880-5.

Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana K. Smetters, Jessica Staddon, and Hao-Chi Wong. Secret Handshakes from Pairing-Based Key Agreements. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy (S&P 2003)*, pages 180–196. IEEE Computer Society, 11–14 May 2003. ISBN 0-7695-1940-7.

Dirk Balfanz, Diana K. Smetters, Paul Stewart, and Hao Chi Wong. Talking to Strangers: Authentication in Ad Hoc Wireless Networks. In NDSS 2002 NDSS 2002. ISBN 1-891562-14-2, 1-891562-13-4.

Endre Bangerter, Jan Camenisch, and Ueli M. Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In Vaudenay [2005], pages 154–171. ISBN 3-540-24454-9.

Michel Barbeau, Jeyanthi Hall, and Evangelos Kranakis. Detecting Impersonation Attacks in Future Wireless and Mobile Networks. In Mike Burmester and Alec Yasinsac, editors, *Revised Selected Papers of the 1st International Workshop Secure Mobile Ad-hoc Networks and Sensors (MADNES 2005)*, volume 4074 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 20–22 Sep 2006. ISBN 3-540-36646-6.

Rida A. Bazzi and Goran Konjevod. On the Establishment of Distinct Identities in Overlay Networks. In *Proceedings of the 24th annual ACM Symposium on Principles of Distributed Computing (PODC05)*, pages 312–320, New York, NY, USA, 17–20 Jul 2005. ACM. ISBN 1-59593-994-2. doi: http://doi.acm.org/10.1145/1073814.1073873.

Vicente Benjumea, Seung Geol Choi, Javier Lopez, and Moti Yung. Anonymity 2.0: X.509 Extensions Supporting Privacy-Friendly Authentication. In Feng Bao and Tatsuaki Okamoto, editors, *CANS 2007, 6th International Conference on Cryptography and Network Security*, volume 4856 of *Lecture Notes in Computer Science*, pages 265–281. Springer-Verlag, 8–10 Dec 2007. ISBN 3-540-57186-8.

Vicente Benjumea, Javier Lopez, Jose Antonio Montenegro, and Jose Maria Troya. A First Approach to Provide Anonymity in Attribute Certificates. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Proceedings of the 7th International Workshop on Practice and Theory in Public Key Cryptography (PKC 2004)*, volume 2947 of *Lecture Notes in Computer Science*, pages 402–415. Springer-Verlag, LNCS 2974, 1–4 Mar 2004. ISBN 3-540-57186-8.

Vicente Benjumea, Javier Lopez, and Jose Maria Troya. Anonymous Attribute Certificates based on Traceable Signatures. *Internet Research: Electronic Networking Applications and Policy. Special Issue on Privacy and Anonymity in the Digital Era: Theory, Technologies and Practice*, 16(2):120–139, 2006. ISSN 1066-2243.

Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform Resource Identifiers (URI): Generic Syntax. RFC 3986, Jan 2005. See http://www.ietf.org/rfc/rfc3986.txt.

Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The Disadvantages of Free MIX Routes and How to Overcome Them. In Hannes Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 30–45, Berlin/Heidelberg, Germany, 25–26 Jul 2000. Springer-Verlag, LNCS 2009.

Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: a taxonomy and open issues. In Ari Juels, Marianne Winslett, and Atsuhiro Goto, editors, *Proceedings of the 2006 Workshop on Digital Identity Management*, pages 1–10, New York, NY, USA, 3 Nov 2006. ACM Press. ISBN 1-59593-547-9. doi: http://doi.acm.org/10.1145/1179529.1179531.

Matt Bishop. *Introduction to Computer Security*. Addison-Wesley, Reading, MA, USA, Nov 2004. ISBN 978-0321247445.

Bluetooth. Specification of the Bluetooth System: Wireless Connections Made Easy. Core Package version: 2.1 + edr. Bluetooth Specification Version 2.1 + EDR, 27 Jul 2007. See http://www.bluetooth.com/.

Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *Proceedings of the 6th International Workshop on Practice and Theory in Public Key Cryptography (PKC 2003)*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 6–8 Jan 2003. ISBN 3-540-00324-X.

Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Franklin [2004], pages 41–55. ISBN 3-540-22668-0.

Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology (ASIACRYPT 2001)*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 9–13 Dec 2001. ISBN 3-540-42987-5.

Katrin Borcea-Pfitzmann, Elke Franz, and Andreas Pfitzmann. Usable presentation of secure pseudonyms. In Vijay Atluri, Pierangela Samarati, and Atsuhiro Goto, editors, *Proceedings of the 2005 Workshop on Digital Identity Management*, pages 70–76, New York, NY, USA, 11 Nov 2005. ACM Press. ISBN 1-59593-232-1. doi: http://doi.acm.org/10.1145/1102486.1102498.

Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM-01)*, pages 180–189, New York, NY, USA, 16–21 Jul 2001. ACM Press.

Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2000)*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer, 14–18 May 2000. ISBN 3-540-67517-5.

Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 618–624, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2260-2.

Stefan Brands. Rapid demonstration of linear relations connected by boolean operators. In Walter Fumy, editor, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 1997)*, volume 1233 of *Lecture Notes in Computer Science*, pages 318–333. Springer, 11–15 May 1997. ISBN 3-540-62975-0.

Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS 2004)*, pages 132–145. ACM, 25–29 Oct 2004. ISBN 1-58113-961-6.

Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking (MOBICOM 2008)*, pages 116–127, New York, NY, USA, 14–19 Sep 2008. ACM Press. ISBN 978-1-60558-096-8. doi: http://doi.acm.org/10.1145/1409944.1409959.

Levente Buttyán and Jean-Pierre Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, 8(5):579–592, 2003. ISSN 1383-469X. doi: http://dx.doi.org/10.1023/A:1025146013151.

Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to Win the Clone Wars: Efficient Periodic n-Times Anonymous Authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, 30 Oct–3 Nov 2006.

Jan Camenisch and Anna Lysyanskaya. A Signature Scheme with Efficient Protocols. In *Security in Communication Networks: Third International Conference (SCN 2002)*, volume 2576/2003 of *Lecture Notes in Computer Science*, pages 268–289, Amalfi, Italy, 12–13 Sep 2002. Springer.

Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Franklin [2004], pages 56–72. ISBN 3-540-22668-0.

Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical Report TR 260, Institute for Theoretical Computer Science, ETH Zürich, Mar 1997.

Alan F. Chalmers. *What is this thing called Science?* Open University Press, Buckingham, England, third edition, 1999. ISBN 0-335-20109-1.

David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 1991)*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 8–11 Apr 1991. ISBN 3-540-54620-0.

David L. Chaum. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. *Communication of the ACM*, 24(2):84–88, Feb 1981.

Bruce Christianson and William S. Harbison. Why isn't trust transitive? In *Proceedings of the International Workshop on Security Protocols*, volume 1189/1997 of *Lecture Notes in Computer Science*, pages 171–176. Springer

Berlin / Heidelberg, 10–12 Apr 1996. ISBN 978-3-540-62494-3. doi: 10.1007/3-540-62494-5.

Thomas Heide Clausen and Philippe Jacquet. Optimized link state routing protocol (olsr). RFC 3626, Oct 2003. See http://www.ietf.org/rfc/rfc3626.txt.

Mathew Scott Corson and Joseph Macker. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, Jan 1999. See http://www.ietf.org/rfc/rfc2501.txt.

Ivan Damgård, Kasper Dupont, and Michael Østergaard Pedersen. Unclonable group identification. In Serge Vaudenay, editor, *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 555–572. Springer, May 28–Jun 1 2006. ISBN 3-540-34546-9.

George Danezis. *Better Anonymous Communications*. PhD thesis, University of Cambridge, Jan 2004.

Bernard Dauenhauer. Paul Ricœur. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2008. URL http://plato.stanford.edu/archives/fall2008/entries/ricoeur/.

Karim El Defrawy and Gene Tsudik. ALARM: Anonymous Location-Aided Routing in Suspicious MANETs. In *Proceedings of the 15th IEEE International Conference on Network Protocols (ICNP 2007)*, pages 304–313. IEEE, 16–19 Oct 2007. ISBN 1-4244-1588-8.

Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, Aug 2008. See http://www.ietf.org/rfc/rfc5246.txt.

Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th Conference on USENIX Security Symposium (USENIX-SS 2004)*, pages 303–320, Berkeley, CA, USA, 9–13 Aug 2004. USENIX Association.

Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Vaudenay [2005], pages 416–431. ISBN 3-540-24454-9.

John R. Douceur. The Sybil Attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems: Proceedings of the 1st International Peer-to-Peer Systems Workshop (IPTPS)*, volume 2429, pages 251–260. Springer-Verlag, 7–8 Mar 2002.

Ralph Droms. Dynamic Host Configuration Protocol. RFC 2131, Mar 1997. See http://www.ietf.org/rfc/rfc2131.txt.

Laura Marie Feeney, Bengt Ahlgren, and Assar Westerlund. Spontaneous Networking: an Application Oriented Approach to Ad Hoc Networking. *IEEE Communications Magazine*, 39:176–181, Jun 2001.

Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Proceedings of Advances in Cryptology (CRYPTO 1986)*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1987.

Roy T. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Jun 1999. See http://www.ietf.org/rfc/rfc2616.txt.

Simone Fischer-Hübner. *IT-Security and Privacy – Design and Use of Privacy-Enhancing Security Mechanisms*, volume 1958 of *Lecture Notes in Computer Science*. Springer-Verlag Berlin/Heidelberg, 2001.

Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, Jamie Van Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the 15th Conference on USENIX Security Symposium (USENIX-SS 2006)*, pages 12–12, Berkeley, CA, USA, 31 Jul–4 Aug 2006. USENIX Association.

Matthew K. Franklin, editor. *Proceedings of the 24th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO 2004)*, volume 3152 of *Lecture Notes in Computer Science*, 15–19 Aug 2004. Springer. ISBN 3-540-22668-0.

Elke Franz and Katrin Borcea-Pfitzmann. Intra-Application Partitioning in an eLearning Environment - a Discussion of Critical Aspects. In *Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES 2006)*, pages 872–878, Washington, DC, USA, 20–22 Apr 2006. IEEE Computer Society. ISBN 0-7695-2567-9. doi: http://dx.doi.org/10.1109/ARES.2006.77.

James A. Freebersyser and Barry Leiner. A DoD Perspective on Mobile Ad Hoc Networks. In Charles E. Perkins, editor, *Ad Hoc Networking*, chapter 2, pages 29–51. Addison-Wesley, Reading, MA, USA, first edition, Dec 2000. ISBN 0201309769.

Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *Proceedings of the 17th Annual International Cryptology Conference on Advances*

in Cryptology (CRYPTO'97), volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 17–21 Aug 1997. ISBN 3-540-63384-7.

Ryan M. Gerdes, Thomas E. Daniels, Mani Mina, and Steve F. Russell. Device identification via analog signal fingerprinting: A matched filter approach. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2006)*. The Internet Society, 2–3 Feb 2006. ISBN 1-891562-22-3, 1-891562-21-5.

Steve Glass, Marius Portmann, and Vallipuram Muthukkumarasamy. Securing wireless mesh networks. *IEEE Internet Computing*, 12(4):30–36, Jul–Aug 2008.

David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding Routing Information. In Ross J. Anderson, editor, *Proceedings of the 1st International Workshop on Information Hiding (IH 1996)*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 30 May – Jun 1 1996. ISBN 3-540-61996-8.

Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. ISSN 0097-5397. doi: http://dx.doi.org/10.1137/0217017.

Marco Gruteser and Dirk Grunwald. Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis. In Parviz Kermani, editor, *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH 2003)*, 19 Sep 2003. ISBN 1-58113-768-0.

Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. *Mobile Networks and Applications*, 10(3):315–325, Jun 2005. ISSN 1383-469X. doi: http://doi.acm.org/10.1145/1145911.1145917.

Fanglu Guo and Tzi-Cker Chiueh. Sequence number-based mac address spoof detection. In Alfonso Valdes and Diego Zamboni, editors, *Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection - Revised Papers (RAID 2005)*, volume 3858 of *Lecture Notes in Computer Science*, pages 309–329. Springer, 7–9 Sep 2005. ISBN 3-540-31778-3.

Sumi Helal, Nitin Desai, Varum Verma, and Choonhwa Lee. Konark – a Service Discovery and Delivery Protocol for Ad Hoc Networks. In *Proceedings of the IEEE Wireless Communications and Networking (WCNC 2003)*, volume 3, pages 2107–2113. IEEE, 16–20 Mar 2003.

Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, pages 3–13, Washington, DC, USA, 20–21 Jun 2002. IEEE Computer Society. ISBN 0-7695-1647-5.

Yih-Chun Hu and Adrian Perrig. A survey of secure wireless ad hoc routing. *IEEE Security and Privacy*, 2(3):28–39, 2004. ISSN 1540-7993. doi: http://dx.doi.org/10.1109/MSP.2004.1.

Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2nd ACM Workshop on Wireless Security (WiSE 2003)*, pages 30–40, New York, NY, USA, 19 Sep 2003. ACM. ISBN 1-58113-769-9. doi: http://doi.acm.org/10.1145/941311.941317.

Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11 (1-2):21–38, Jan 2005. ISSN 1022-0038. doi: http://dx.doi.org/10.1007/s11276-004-4744-y.

Dijiang Huang. Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *International Journal of Security and Networks (IJSN)*, 2(3/4):272–283, 2007.

Jean-Pierre Hubaux, Levente Buttyán, and Srdjan Čapkun. The Quest for Security in Mobile Ad Hoc Networks. In *Proceedings of the 2th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'01)*, pages 146–155, New York, NY, USA, 4–5 Oct 2001. ACM Press. ISBN 1-58113-428-2.

IEEE 802.15.3. *IEEE Std 802.15.3, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 15.1: Wireless medium access control (MAC)and physical layer (PHY) specificatons for high rate wireless personal area networks (WPANs)*. IEEE Computer Society, New York, NY, USA, 29 Sep 2003.

IEEE 802.1X. *IEEE Std 802.1X, IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control*. IEEE Computer Society, New York, NY, USA, 13 Dec 2004.

IEEE 802.15.1. *IEEE Std 802.15.1, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 15.1: Wireless*

medium access control (MAC)and physical layer (PHY) specificatons for wireless personal area networks (WPANs). IEEE Computer Society, New York, NY, USA, 14 Jun 2005.

IEEE 802.15.4. *IEEE Std 802.15.4, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 15.1: Wireless medium access control (MAC)and physical layer (PHY) specificatons for low-rate wireless personal area networks (WPANs)*. IEEE Computer Society, New York, NY, USA, 8 Sep 2006.

IEEE 802.11. *IEEE Std 802.11, 2007, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE Computer Society, New York, NY, USA, 12 Jun 2007.

IEEE 802.16. *IEEE Std 802.16, IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed Broadband Wireless Access System*. IEEE Computer Society and the IEEE Microwave Theory and Techniques Society, New York, NY, USA, 1 Oct 2004.

IETF autoconf. Ad Hoc Network Autoconfiguration (autoconf), 2006. See http://www3.ietf.org/html.charters/autoconf-charter.html.

IETF zeroconf. Zero Configuration Networking (zeroconf), 2003. See http://www.zeroconf.org/zeroconf-charter.html.

HMAC. *The Keyed-Hash Message Authentication Code (HMAC)*. Information Technology Laboratory – National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Jul 2008.

ITUT X.509. ITU-T Recommendation X.509, The Directory: public-key and attribute certificate frameworks. Recommendation X.509 - International Telecommunications Union, The International Telegraph and Telephone Consultative Committee, Data Communication Networks: Open Systems Interconnection (OSI); Security, Structure and Applications, Aug 2005.

ITUT X.800. Security architecture for open systems interconnection for ccit applications. Recommendation X.800 - International Telecommunications Union, The International Telegraph and Telephone Consultative Committee, Data Communication Networks: Open Systems Interconnection (OSI); Security, Structure and Applications, Mar 1991.

Van Jacobson, Bob Braden, and Dave Borman. TCP Extensions for High Performance. RFC 1323, May 1992. See http://www.ietf.org/rfc/rfc1323.txt.

Raj Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, New York, NY, USA, Apr 1991. ISBN 0471503361.

Shu Jiang, Nitin H. Vaidya, and Wei Zhao. A Mix Route Algorithm for Mix-net in Wireless Mobile Ad Hoc Networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS2004)*, 24–27 Oct 2004.

Jini. The Jini Architecture Specification – Version 1.2, 2001. See http://www.sun.com/software/jini/specs/.

David B. Johnson, David A. Maltz, and Yih-Chun Hu. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728, Feb 2007. See http://www.ietf.org/rfc/rfc4728.txt.

Audun Jøsang. The right type of trust for distributed systems. In *Proceedings of the 1996 Workshop on New Security Paradigms (NSPW 1996)*, pages 119–131, New York, NY, USA, 23–26 Sep 1996. ACM. ISBN 0-89791-944-0. doi: http://doi.acm.org/10.1145/304851.304877.

Audun Jøsang. An algebra for assessing trust in certification chains. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 1999)*. The Internet Society, 3–5 Feb 1999. ISBN 1-891562-04-5, 1-891562-05-3.

Joseph M. Kahn, Randy Howard Katz, and Kristofer S. J. Pister. Emerging challenges: Mobile networking for smart dust. *Journal of Communications and Networks*, 2(3):188–196, Sep 2000.

Frank Kargl, Stefan Schlott, and Michael Weber. Identification in Ad Hoc Networks. In *Proceedings of the 39th Hawaiian International Conference on System Sciences (HICSS-39)*, Washington, DC, USA, 4–7 Jan 2006. IEEE Computer Society. ISBN 0-7695-2507-5.

Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, Sep 2003.

Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In Christian Cachin and Jan Camenisch, editors, *Proceedings of the 23th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2004)*, volume 3027 of *Lecture Notes in Computer Science*, pages 571–589. Springer, 2–6 May 2004. ISBN 3-540-21935-8.

Yongdae Kim, Daniele Mazzocchi, and Gene Tsudik. Admission control in peer groups. In *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA 2003)*, pages 131–139. IEEE Computer Society, 16–18 Apr 2003. ISBN 0-7695-1938-5.

Tadayoshi Kohno, Andre Broido, and Kimberly C. Claffy. Remote physical device fingerprinting. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P 2005)*, pages 211–225. IEEE Computer Society, 8–11 May 2005. ISBN 0-7695-2339-0.

Jeijun Kong and Xiaoyan Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad Hoc Networks. In MOBIHOC 2003 MOBIHOC 2003, pages 291–302. ISBN 1-58113-684-6. doi: http://doi.acm.org/10.1145/778415.778449.

Jiejun Kong, Xiaoyan Hong, Medy Yahya Sanadidi, and Mario Gerla. Mobility Changes Anonymity: Mobile Ad Hoc Networks Need Efficient anonymous Routing. In *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, pages 57–62, Washington, DC, USA, 27–30 Jun 2005. IEEE Computer Society. ISBN 0-7695-2373-0. doi: http://dx.doi.org/10.1109/ISCC.2005.105.

Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *Proceedings of the 9th International Conference on Network Protocols (ICNP 2001)*, pages 251–260, Washington, DC, USA, 11–14 Nov 2001. IEEE Computer Society.

Sébastien Kunz-Jacques, Gwenaëlle Martinet, Guillaume Poupard, and Jacques Stern. Cryptanalysis of an efficient proof of knowledge of discrete logarithm. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Proceedings of the 9th International Workshop on Theory and Practice in Public Key Cryptography (PKC 2006)*, volume 3958 of *Lecture Notes in Computer Science*, pages 27–43. Springer, 24–26 Apr 2006. ISBN 3-540-33851-9.

Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Lydia E. Kavraki, Dan S. Wallach, and Guillaume Marceau. Robotics-based location sensing using wireless ethernet. In Ian F. Akyildiz, Jason Yi-Bing Lin, Ravi Jain, Vaduvur Bharghavan, and Andrew T. Campbell, editors, *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MOBICOM 2002)*, pages 227–238, New York, NY, USA, 23–28 Sep 2002. ACM Press. ISBN 1-58113-486-X.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transanctions on Programming Languages and Systems*, 4

(3):382–401, 1982. ISSN 0164-0925. doi: http://doi.acm.org/10.1145/357172.357176.

Brian Neil Levine, Clay Shields, and N. Boris Margolin. A Survey of Solutions to the Sybil Attack. Technical Report 2006-052, University of Massachusetts Amherst, Amherst, MA, USA, Oct 2006.

Yingbin Liang, H. Vincent Poor, and Shlomo Shamai. Secrecy capacity region of fading broadcast channels. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2007)*, pages 1291–1295, 24–29 Jun 2007. ISBN 978-1-4244-1397-3.

Xiaodong Lin, Rongxing Lu, Chenxi Zhang, Haojin Zhu, Pin-Han Ho, and Xuemin Shen. Security in Vehicular Ad Hoc Networks. *IEEE Communications Magazine*, 46(4):88–95, Apr 2008.

Xiaodong Lin, Xiaoting Sun, Pin-Han Ho, and Xuemin Shen. GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications. *IEEE Transactions on Vehicular Technology*, 56(6):3442–3456, Nov 2007.

Jun Liu, Jiejun Kong, Xiaoyan Hong, and Mario Gerla. Performance evaluation of anonymous routing protocols in MANETs. In *Proceedings of the IEEE Wireless Communications and Networking (WCNC 2006)*, volume 2, pages 646–651. IEEE, 3–6 Apr 2006. ISBN 1-4244-0269-7.

Rolf Lunheim and Guttorm Sindre. Privacy and Computing: a Cultural Perspective. In Richard Sizer, Louise Yngström, Henrik Kaspersen, and Simone Fischer-Hübner, editors, *Proceedings of the IFIP TC9/WG9.6 Working Conference on Security and Control of Information Technology in Society*, pages 25–40. North-Holland, 12–17 Aug 1993. ISBN 0-444-81831-6.

Haiyun Luo, Petros Zefros, Jiejun Kong, Songwu Lu, and Lixia Zhang. Self-securing Ad Hoc Wireless Networks. In *Proceedings of the 7th IEEE Symposium on Computers and Communications (ISCC 2002)*, pages 567–574, 1–4 Jul 2002. ISBN 0-7695-1671-8.

Ningrinla Marchang and Raja Datta. Collaborative techniques for intrusion detection in mobile ad-hoc networks. *Ad Hoc Networks*, 6(4):508–523, 2008. ISSN 1570-8705. doi: http://dx.doi.org/10.1016/j.adhoc.2007.04.003.

N. Boris Margolin and Brian Neil Levine. Quantifying Resistance to the Sybil Attack. In Gene Tsudik, editor, *Proceedings of the 12th Financial Cryptography and Data Security International Conference (FC08)*, volume 5143 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 28–31 Jan 2008. ISBN 978-3-540-85229-2. doi: http://dx.doi.org/10.1007/978-3-540-85230-8_1.

Leonardo A. Martucci. The Identity Anonymity Paradox: on the Relationship between Identification, Anonymity and Security in Mobile Ad Hoc Networks, Licentiate Thesis, Karlstad University Studies 2006:36, Sep 2006.

Leonardo A. Martucci, Christer Andersson, and Simone Fischer-Hübner. Towards Anonymity in Mobile Ad Hoc Networks: the Chameleon Protocol and its Anonymity Analysis. Technical Report 2006:35, Karlstad University, Karlstad, Sweden, Aug 2006a.

Leonardo A. Martucci, Christer Andersson, Wim Schreurs, and Simone Fischer-Hübner. Trusted Server Model for Privacy-Enhanced Location Based Services. In Viiveke Fåk, editor, *Proceedings of the 11th Nordic Workshop on Secure IT Systems (NordSec 2006)*, pages 13–25, 19–20 Oct 2006b.

Leonardo A. Martucci, Tereza Cristina M. B. Carvalho, and Wilson V. Ruggiero. A Lightweight Distributed Group Authentication Mechanism. In Steven M. Furnell and Paul S. Downland, editors, *Proceedings of the 4th International Network Conference (INC 2004)*, pages 393–400, Plymouth, Devon, UK, 6–9 Jul 2004a. ISBN 1-84102-125-3.

Leonardo A. Martucci, Markulf Kohlweiss, Christer Andersson, and Andryi Panchenko. Self-Certified Sybil-Free Pseudonyms. In *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec'08)*, pages 154–159. ACM Press, Mar 31 – Apr 2 2008a. ISBN 978-1-59593-814-5.

Leonardo A. Martucci, Christiane M. Schweitzer, Yeda R. Venturini, Tereza C. M. B. Carvalho, and Wilson V. Ruggiero. A Trust-Based Security Architecture for Small and Medium-Sized Mobile Ad Hoc Networks. In Ian F. Akyildiz, Erdal Cayirci, Eylem Ekici, and Giacomo Morabito, editors, *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2004)*, pages 278–290, 27–30 Jun 2004b. ISBN 975-98840-0-3.

Leonardo A. Martucci, Albin Zuccato, and Simone Fischer-Hübner. Identity Deployment and Management in Wireless Mesh Networks. In Simone Fischer-Hübner, Penny Duquenoy, Albin Zuccato, and Leonardo Martucci, editors, *The Future of Identity in the Information Society, Proceedings of the 3rd IFIP WG 9.2, 9.6/11.6, 11.7/FIDIS International Summer School*, pages 223–233, New York, NY, USA, 4–10 Aug 2008b. Springer. ISBN 978-0-387-79025-1.

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996. ISBN 0849385237.

Johann Van Der Merwe, Dawoud Dawoud, and Stephen McDonald. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing*

Surveys, 39(1):1–45, Apr 2007. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/1216370.1216371.

David L. Mills. Network Time Protocol (Version 3) – Specification, Implementation and Analysis. RFC 1305, Mar 1992. See http://www.ietf.org/rfc/rfc1305.txt.

MOBIHOC 2003. *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03)*, New York, NY, USA, 1–3 Jun 2003. ACM Press. ISBN 1-58113-684-6.

Gabriel Montenegro and Claude Castelluccia. Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses. In NDSS 2002 NDSS 2002. ISBN 1-891562-14-2, 1-891562-13-4.

Oliver Morton. A survey of defence technology: To dissolve, to disappear. *The Economist*, 335(7918):5–20, 10 Jun 1995.

Farid Naït-Abdesselam, Brahim Bensaou, and Tarik Taleb. Detecting and Avoiding Wormhole Attacks in Wireless Ad Hoc Networks. *IEEE Communications Magazine*, 46(4):127–133, Apr 2008.

NDSS 2002. *Proceedings of the Network and Distributed System Security Symposium (NDSS 2002)*, 6–8 Feb 2002. The Internet Society. ISBN 1-891562-14-2, 1-891562-13-4.

James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN04)*, pages 259–268, New York, NY, USA, 26–27 Apr 2004. ACM Press.

Richard G. Ogier, Fred L. Templin, and Mark G. Lewis. Topology dissemination based on reverse-path forwarding (tbrpf). RFC 3684, Feb 2004. See http://www.ietf.org/rfc/rfc3684.txt.

OPNET. *OPNET Modeler v.14.0*. OPNET Technologies, Inc., Bethesda, MD, USA, 2007. See http://www.opnet.com.

Panos Papadimitratos and Zygmunt J. Haas. Secure Routing for Mobile Ad hoc Networks. In *Proceedings of the SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, pages 193–204, San Antonio, TX, USA, 27–31 Jan 2002.

Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'91)*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 11–15 Aug 1992. ISBN 3-540-55188-3.

Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, Jul 2003. See http://www.ietf.org/rfc/rfc3561.txt.

Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination sequenced distance vector routing (dsdv) for mobile computers. *SIGCOMM Computer Communication Review*, 24(4):234–244, Oct 1994. ISSN 0146-4833. doi: http://doi.acm.org/10.1145/190809.190336.

Adrian Perrig, Ran Canetti, Dawn Xiaodong Song, and J. D. Tygar. Efficient and secure source authentication for multicast. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2001)*, pages 35–46. The Internet Society, 6–8 Feb 2001. ISBN 1-891562-10-X, 1-891562-11-8.

Adrian Perrig, John Stankovic, and David Wagner. Security in Wireless Sensor Networks. *Communications of the ACM*, 47(6):53–57, 2004. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/990680.990707.

Andreas Pfitzmann and Marit Hansen. Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology v0.31, 15 Feb 2008. See http://dud.inf.tu-dresden.de/literatur/.

Tom Phelan. Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP). RFC 5238, May 2008. See http://www.ietf.org/rfc/rfc5238.txt.

Chris Piro, Clay Shields, and Brian Neil Levine. Detecting the Sybil Attack in Ad Hoc Networks. In SecureComm 2006 SecureComm 2006, pages 1–11. ISBN 1-4244-0423-1.

Jon Postel. User datagram protocol. RFC 768, 28 Aug 1980. See http://www.ietf.org/rfc/rfc0768.txt.

Jon Postel. Internet control message protocol. RFC 792, Sep 1981. See http://www.ietf.org/rfc/rfc0792.txt.

Michael Reiter and Avi Rubin. Crowds: Anonymity for Web Transactions. In *DIMACS Technical report*, pages 97–115, 1997.

Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/290163.290168.

Eric Rescorla and Nagendra Modadugu. Datagram Transport Layer Security. RFC 4347, Apr 2006. See http://www.ietf.org/rfc/rfc4347.txt.

Paul Ricœur. *Oneself as Another*. University of Chicago Press, Chicago, IL, USA, Jan 1992. ISBN 9780226713298.

Howard Robinson. Dualism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2008. URL http://plato.stanford.edu/archives/fall2008/entries/dualism/.

Christian Rohner, Erik Nordström, Per Gunningberg, and Christian Tschudin. Interactions between TCP, UDP and routing protocols in wireless multi-hop ad hoc networks. In *Proceedings of the 1st IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality (REALMAN 2005)*, pages 69–76, 14 Jul 2005.

David Sanchez Sanchez and Heribert Baldus. Hybrid key management for mobile ad hoc networks. In Khaldoun Al Agha, Isabelle Guérin Lassous, and Guy Pujolle, editors, *Proceedings of the 4th Annual Mediterranean Ad Hoc Networking Workshop, Challenges in Ad Hoc Networks (Med-Hoc-Net 2005)*, pages 337–346. Springer, 21–24 Jun 2005. ISBN 9780-387-31171-8.

Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of the 10th International Conference on Network Protocols (ICNP 2002)*, pages 78–89, Washington, DC, USA, 12–15 Nov 2002. IEEE Computer Society. ISBN 0-7695-1856-7.

Nitesh Saxena, Gene Tsudik, and Jeong Hyun Yi. Admission control in peer-to-peer: design and performance evaluation. In Sanjeev Setia and Vipin Swarup, editors, *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2003)*, pages 104–113, New York, NY, USA, 31 Oct 2003. ACM. ISBN 1-58113-783-4.

Nitesh Saxena, Gene Tsudik, and Jeong Hyun Yi. Efficient node admission for short-lived mobile ad hoc networks. In *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP 2005)*, pages 269–278, Washington, DC, USA, 6–9 Nov 2005. IEEE Computer Society. ISBN 0-7695-2437-0.

Claus P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.

SecureComm 2006. *Proceedings of the 2nd IEEE/CreateNet International Conference on Security and Privacy in Communication Networks (SECURECOMM 2006)*, 28 Aug–1 Sep 2006. ISBN 1-4244-0423-1.

Stefaan Seys and Bart Preneel. ARM: Anonymous Routing Protocol for Mobile Ad Hoc Networks. In *International Workshop on Pervasive Computing and Ad Hoc Communications (PCAC06), Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, volume 2, pages 133–137, Washington, DC, USA, 18–19 Apr 2006. IEEE Computer Society. ISBN 0-7695-2466-4-02. doi: http://dx.doi.org/10.1109/AINA.2006.104.

Christopher Shields. Aristotle's Psychology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2008. URL http://plato.stanford.edu/archives/win2008/entries/aristotle-psychology/.

Robert W. Shirey. Internet Security Glossary, Version 2. RFC 4949, Aug 2007. See http://www.ietf.org/rfc/rfc4949.txt.

Ronggong Song, Larry Korba, and George Yee. Anondsr: efficient anonymous dynamic source routing for mobile ad-hoc networks. In *Proceedings of the 3rd ACM workshop on Security of Ad Hoc and Sensor Networks (SASN 2005)*, pages 33–42, New York, NY, USA, 7 Nov 2005. ACM. ISBN 1-59593-227-5. doi: http://doi.acm.org/10.1145/1102219.1102226.

Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. Fair blind signatures. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 1995)*, volume 921 of *Lecture Notes in Computer Science*, pages 209–219. Springer, 21–25 May 1995. ISBN 3-540-59409-4.

Frank Stajano. The Resurrecting Duckling: What Next? In *Revised Papers from the 8th International Workshop on Security Protocols*, Lecture Notes in Computer Science, LNCS 2133, pages 204–214, London, UK, 3–5 Apr 2001. Springer. ISBN 3-540-42566-7.

Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks. In *Proceedings of the 3rd AT&T Software Symposium*, Oct 1999.

Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy (S&P 1997)*, pages 44–54. IEEE Computer Society, 4–7 May 1997. ISBN 0-8186-7828-3.

Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-times anonymous authentication (extended abstract). In *Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology (ASIACRYPT 2004)*, volume 3329 of *Lecture*

Notes in Computer Science, pages 308–322. Springer, 5–9 Dec 2004. ISBN 3-540-23975-8.

Gene Tsudik and Shouhuai Xu. A flexible framework for secret handshakes. In George Danezis and Philippe Golle, editors, *Revised Selected Papers of the 6th International Workshop Privacy Enhancing Technologies (PET 2006)*, volume 4258 of *Lecture Notes in Computer Science*, pages 295–315. Springer, 28–30 Jun 2006. ISBN 3-540-68790-4.

United Nations. *Universal Declaration of Human Rights, Resolution 217 A (III)*. United States Government Print. Off., Washington, DC, USA, 1949.

UPnP. UPnP Device Architecture 1.0, Jul 2006. See http://www.upnp.org.

Serge Vaudenay, editor. *Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC 2005)*, volume 3386 of *Lecture Notes in Computer Science*, 23–26 Jan 2005. Springer. ISBN 3-540-24454-9.

Srdjan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. Small worlds in security systems: an analysis of the pgp certificate graph. In *Proceedings of the 2002 Workshop on New Security Paradigms (NSPW 2002)*, pages 28–35, New York, NY, USA, 23–26 Sep 2002. ACM. ISBN 1-58113-598-X. doi: http://doi.acm.org/10.1145/844102.844108.

Srdjan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, Jan–Mar 2003a.

Srdjan Čapkun, Levente Buttyán, and Jean-Pierre Hubaux. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, Jan 2006. ISSN 1536-1233. doi: http://dx.doi.org/10.1109/TMC.2006.12.

Srdjan Čapkun, Jean-Pierre Hubaux, and Levente Buttyán. Mobility Helps Security in Ad Hoc Networks. In MOBIHOC 2003 MOBIHOC 2003, pages 46–56. ISBN 1-58113-684-6.

Srdjan Čapkun, Jean-Pierre Hubaux, and Markus Jakobsson. Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks. Technical Report IC/2004/10, EPFL-IC, CH-1015 Lausanne, Switzerland, 30 Jan 2004.

Yeda R. Venturini, Christiane M. Schweitzer, Leonardo A. Martucci, Fernando F. Redigolo, Armin W. Mittelsdorf, Wilson V. Ruggiero, and Tereza Cristina M. B. Carvalho. Security Model for Ad Hoc Networks. In *Proceedings of the International Conference on Wireless Networks (ICWN 2002)*, pages 185–191, Las Vegas, NV, USA, 24–27 Jun 2002.

Samuel Warren and Louis Brandeis. The Right to Privacy. *Harvard Law Review*, 4(5), 15 Dec 1890.

Duncan J. Watts. *Small Words: the Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ, USA, 23 Aug 1999. ISBN 0-691-00541-9.

Mark Weiser. Creating the Invisible Interface (invited talk). In *Proceedings of the 7th annual ACM Symposium on User Interface Software and Technology (UIST 1994)*, page 1. ACM Press, 2–4 Nov 1994.

Alan F. Westin. *Privacy and Freedom*. Atheneum, New York, NY, USA, 1967.

WiSE 2002. *Proceedings of the 1st ACM Workshop on Wireless Security (WiSE 2002)*, New York, NY, USA, 28 Sep 2002. ACM. ISBN 1-58113-585-8.

Bing Wu, Jie Wu, and Mihaela Cardei. A survey of key management in mobile ad hoc networks. In Zhang et al. [2008], chapter 30, pages 479–499. ISBN 978-1599048994.

Bin Xiao, Bo Yu, and Chuanshan Gao. Detection and localization of sybil nodes in vanets. In *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS 2006)*, pages 1–8, New York, NY, USA, 26 Sep 2006. ACM. ISBN 1-59593-471-5. doi: http://doi.acm.org/10.1145/1160972.1160974.

Liu Yang, Markus Jakobsson, and Susanne Wetzel. Discount anonymous on demand routing for mobile ad hoc networks. In SecureComm 2006 SecureComm 2006. ISBN 1-4244-0423-1.

Seung Yi and Robin Kravets. Composite key management for ad hoc networks. In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services*, pages 52–61. IEEE Computer Society, 22–25 Aug 2004. ISBN 0-7695-2208-4.

Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. SybilGuard: defending against sybil attacks via social networks. In Luigi Rizzo, Thomas E. Anderson, and Nick McKeown, editors, *Proceedings of the ACM SIGCOMM 2006 Conference on Applications Technologies, Architectures, and Protocols for Computer Communications*, pages 267–278. ACM, 11–15 Sep 2006. ISBN 1-59593-308-5.

Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham D. Flaxman. Sybilguard: defending against Sybil attacks via social networks. *IEEE/ACM Transactions on Networking*, 16(3):576–589, 2008. ISSN 1063-6692. doi: http://dx.doi.org/10.1109/TNET.2008.923723.

Manel Guerrero Zapata and Nadarajah Asokan. Securing ad hoc routing protocols. In WiSE 2002 WiSE 2002, pages 1–10. ISBN 1-58113-585-8. doi: http://doi.acm.org/10.1145/570681.570682.

Yanchao Zhang, Wei Liu, and Wenjing Lou. Anonymous Communication in Mobile Ad Hoc Networks. In *Proceedings of the* 24^{th} *Annual Joint Conference of the IEEE Communication Society (INFOCOM 2005)*, volume 3, pages 1940–1951, Miami, FL, USA, 13–17 Mar 2005.

Yang Zhang, Jun Zheng, and Miao Ma, editors. *Handbook of Research on Wireless Security*. Information Science Reference, Hershey, PA, USA, first edition, 14 Mar 2008. ISBN 978-1599048994.

Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.

Philip R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA, May 1995. ISBN 978-0-262-74017-3. Out Of Print.

Index of References

Merwe et al. [2007], 18, 19, 31, 34, 35, 181
Mills [1992], 29, 182
Montenegro and Castelluccia [2002], 18, 31, 38, 182
Morton [1995], 17, 182
NDSS 2002 [], 169, 182
Naït-Abdesselam et al. [2008], 24, 182
Newsome et al. [2004], 52–54, 182
OPNET [], 130, 182
Ogier et al. [2004], 20, 182
Papadimitratos and Haas [2002], 35, 36, 182
Pedersen [1992], 164, 165, 182
Perkins and Bhagwat [1994], 36, 183
Perkins et al. [2003], 20, 29, 36, 138, 182
Perrig et al. [2001], 37, 183
Perrig et al. [2004], 17, 183
Pfitzmann and Hansen [2008], 5, 6, 26, 39, 56, 68, 183
Phelan [2008], 103, 183
Piro et al. [2006], 54, 183
Postel [1980], 131, 183
Postel [1981], 29, 183
Reiter and Rubin [1997], xv, 47, 95, 97–99, 105, 114, 115, 118, 121, 145, 183
Reiter and Rubin [1998], 44, 183
Rescorla and Modadugu [2006], 101, 103, 183
Ricœur [1992], 5, 183
Robinson [2008], 5, 184
Rohner et al. [2005], 131, 184
Sanchez and Baldus [2005], 31, 184
Sanzgiri et al. [2002], 36, 37, 184
Saxena et al. [2003], 89, 184
Saxena et al. [2005], 89, 184
Schnorr [1991], 163, 184
SecureComm 2006 [], 183, 184, 187
Seys and Preneel [2006], 42, 184
Shields [2008], 5, 185
Shirey [2007], 7, 185
Song et al. [2005], 41, 43–45, 185
Stadler et al. [1995], 92, 185
Stajano and Anderson [1999], 18, 22, 32, 185
Stajano [2001], 18, 25, 32, 185
Syverson et al. [1997], 41, 43–45, 96, 185
Teranishi et al. [2004], 161, 185
Tsudik and Xu [2006], 10, 90, 186
UPnP [], 32, 101, 186
United Nations [1949], 4, 186
Vaudenay [2005], 169, 173, 186
Venturini et al. [2002], 32, 186
Warren and Brandeis [1890], 4, 186
Watts [1999], 34, 187
Weiser [1994], 1, 187
Westin [1967], 4, 187
WiSE 2002 [], 168, 187, 188
Wu et al. [2008], 31, 187
Xiao et al. [2006], 54, 187
Yang et al. [2006], 42, 187
Yi and Kravets [2004], 35, 187
Yu et al. [2006], 55, 187
Yu et al. [2008], 55, 187
Zapata and Asokan [2002], 35, 36, 187
Zhang et al. [2005], 40, 41, 43, 44, 188
Zhang et al. [2008], 168, 187, 188
Zhou and Haas [1999], 19, 34, 188
Zimmermann [1995], 34, 188
Čapkun et al. [2002], 34, 186
Čapkun et al. [2003a], 19, 34, 35, 186
Čapkun et al. [2003b], 19, 186
Čapkun et al. [2004], 41–43, 186
Čapkun et al. [2006], 18, 19, 22, 34, 186